



ALTAIR

ONLY FORWARD

Altair Accelerator 2021.2.0

Tutorials

Intellectual Property Rights Notice

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

Copyright © 1986-2021 Altair Engineering Inc. 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.

Altair Simulation Products

Altair AcuConsole™ ©2006-2021

Altair AcuSolve™ ©1997-2021

Altair Activate® ©1989-2021 (formerly solidThinking Activate)

Altair Compose® © 2007-2021 (formerly solidThinking Compose)

Altair ConnectMe™ © 2014-2021

Altair EDEM © 2005-2021 DEM Solutions Ltd, © 2019-2021 Altair Engineering Inc.

Altair ElectroFlo™ ©1992-2021

Altair Embed® ©1989-2021 (formerly solidThinking Embed)

- **Altair Embed SE** ©1989-2021 (formerly solidThinking Embed SE)
- **Altair Embed / Digital Power Designer** © 2012-2021
- **Altair Embed Viewer** © 1996-2021

Altair ESAComp™ ©1992-2021

Altair Feko™ © 1999-2014 Altair Development S.A. (Pty) Ltd., © 2014-2021 Altair Engineering Inc.

Altair Flux™ ©1983-2021

Altair FluxMotor™ ©2017-2021

Altair HyperCrash™ ©2001-2021

Altair HyperGraph™ ©1995-2021

Altair HyperLife™ ©1990-2021

Altair HyperMesh™ ©1990-2021

Altair HyperStudy™ ©1999-2021

Altair HyperView™ ©1999-2021

Altair HyperWorks™ ©1990-2021

Altair HyperXtrude™ ©1999-2021

Altair Inspire™ ©2009-2021 including Altair Inspire Motion, Altair Inspire Structures, and Altair Inspire Print3D

Altair Inspire Cast ©2011-2021 (formerly Click2Cast)

Altair Inspire ElectroFlo ©1992-2021

Altair Inspire Extrude Metal ©1996-2021 (formerly Click2Extrude - Metal)
Altair Inspire Extrude Polymer ©1996-2021 (formerly Click2Extrude - Polymer)
Altair Inspire Form ©1998-2021 (formerly Click2Form)
Altair Inspire Friction Stir Welding ©1996-2021
Altair Inspire Mold ©2009-2021
Altair Inspire PolyFoam ©2009-2021
Altair Inspire Play ©2009-2021
Altair Inspire Render ©1993-2016 Solid Iris Technologies Software Development One PLLC, ©
2016-2021 Altair Engineering Inc. (formerly Thea Studio)
Altair Inspire Resin Transfer Molding ©1990-2021
Altair Inspire Studio ©1993-2021 (formerly 'Evolve')
Altair Manufacturing Solver™ ©2011-2021
Altair Material Data Center ©2019-2021
Altair MotionSolve™ ©2002-2021
Altair MotionView™ ©1993-2021
Altair Multiscale Designer™ ©2011-2021
Altair nanoFluidX™ ©2013-2018 Fluidyna GmbH, © 2018-2021 Altair Engineering Inc.
Altair newFASANT ©2010-2021 Altair Software and Services, S.L., ©2010-2021 Altair Engineering Inc.
Altair OptiStruct™ ©1996-2021
Altair PolIEx ©2003-2021
Altair Radioss™ ©1986-2021
Altair Seam™ ©1985-2019 Cambridge Collaborative, Inc., © 2019-2021 Altair Engineering Inc.
Altair SimLab™ ©2004-2021
Altair SimSolid™ ©2015-2021
Altair ultraFluidX™ © 2010-2018 Fluidyna GmbH, © 2018-2021 Altair Engineering Inc.
Altair Virtual Wind Tunnel™ ©2012-2021
Altair WinProp™ ©2000-2021
Altair WRAP ©1998-2021 Altair Engineering AB

Altair Packaged Solution Offerings (PSOs)
Altair Automated Reporting Director™ ©2008-2021
Altair GeoMechanics Director™ ©2011-2021
Altair Impact Simulation Director™ ©2010-2021
Altair Model Mesher Director™ ©2010-2021

Altair NVH Director™ ©2010-2021

Altair Squeak and Rattle Director™ ©2012-2021

Altair Virtual Gauge Director™ ©2012-2021

Altair Weight Analytics™ ©2013-2021

Altair Weld Certification Director™ ©2014-2021

Altair Multi-Disciplinary Optimization Director™ ©2012-2021.

Altair HPC & Cloud Products

Altair® PBS Professional® ©1994-2021

Altair Control™ ©2008-2021; (formerly **PBS Control**)

Altair Access™ ©2008- 2021; (formerly **PBS Access**)

Altair Accelerator™ ©1995- 2021; (formerly **NetworkComputer**)

Altair Accelerator™ Plus ©1995- 2021; (formerly **WorkloadXelerator**)

Altair FlowTracer™ ©1995- 2021; (formerly **FlowTracer**)

Altair Allocator™ ©1995- 2021; (formerly **LicenseAllocator**)

Altair Monitor™ ©1995- 2021; (formerly **LicenseMonitor**)

Altair Hero™ ©1995- 2021; (formerly **HERO**)

Altair Software Asset Optimization™ (SAO) ©2007- 2021



Note:

Compute Manager™ ©2012-2020 is now part of **Altair Access**

Display Manager™ ©2013-2020 is now part of **Altair Access**

PBS Application Services™ ©2008-2020 is now part of **Altair Access**

PBS Analytics™ ©2008-2020 is now part of **Altair Control**

PBS Desktop™ ©2008-2020 is now part of **Altair Access**, specifically **Altair Access desktop**, which also has **Altair Access web** and **Altair Access mobile**

e-Compute™ ©2000-2020 was replaced by "**Compute Manager**" which is now **Altair Access**

Altair Data Analytics Products

Altair Knowledge Studio® © 1994-2020 Angoss Software Corporation, © 2021 Altair Engineering Inc.

Altair Knowledge Studio for Apache Spark © 1994-2020 Angoss Software Corporation, © 2020-2021 Altair Engineering Inc.

Altair Knowledge Seeker™ © 1994-2020 Angoss Software Corporation, © 2020-2021 Altair Engineering Inc.

Altair Knowledge Hub™ © 2017-2020 Datawatch Corporation, © 2020-2021 Altair Engineering Inc.

Altair Monarch™ © 1996-2020 Datawatch Corporation, © 2020-2021 Altair Engineering Inc.

Altair Monarch Server © 1996-2020 Datawatch Corporation, © 2020-2021 Altair Engineering Inc.

Altair Panopticon™ © 2004-2020 Datawatch Corporation, © 2020-2021 Altair Engineering Inc.

Altair SmartWorks™

Altair SmartCore™ © 2011-2021

Altair SmartEdge™ © 2011-2021

Altair SmartSight™ © 2011-2021

Altair One™ ©1994-2021

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.

For a complete list of Altair Accelerator Third Party Software Licenses, please click [here](#).

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 web FAQs, tutorials, training classes, telephone and e-mail.

Altair One Customer Portal

Altair One (<https://altairone.com/>) is Altair's customer portal giving you access to product downloads, Knowledge Base and customer support. We strongly recommend that all users create an Altair One account and use it as their primary means of requesting technical support.

Once your customer portal account is set up, you can directly get to your support page via this link: www.altair.com/customer-support/.

Altair Training Classes

Altair training courses provide a hands-on introduction to our products, focusing on overall functionality. Courses are conducted at our main and regional offices or at your facility. If you are interested in training at your facility, please contact your account manager for more details. If you do not know who your account manager is, e-mail your local support office and your account manager will contact you

Telephone and E-mail

If you are unable to contact Altair support via the customer portal, you may reach out to the technical support desk via phone or e-mail. You can use the following table as a reference to locate the support office for your region.

When contacting Altair support, please specify the product and version number you are using along with a detailed description of the problem. It is beneficial for the support engineer to know what type of workstation, operating system, RAM, and graphics board you have, so please include that in your communication.

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

Location	Telephone	E-mail
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
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

See www.altair.com for complete information on Altair, our team and our products.

3rd Party Licenses

ArgumentParser

A slimline C++ class for parsing command-line arguments

Copyright (c) 2017, Hilton Bristow All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Boost

Portions of the Boost collection of C++ libraries are used in certain Altair Engineering software to aid in software portability across platforms.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Bootstrap

Version: 4.3.1

The MIT License (MIT)

Copyright (c) 2011-2021 Twitter, Inc.

Copyright (c) 2011-2021 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

client-oauth2

Version: 4.2.5

Copyright 2014 (c) MuleSoft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

formik

Version: 1.5.2

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union

of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory

patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- a.** You must give any other recipients of the Work or Derivative Works a copy of this License; and
- b.** You must cause any modified files to carry prominent notices stating that You changed the files; and
- c.** You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- d.** If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR

PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2020 Formik, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Fossil SCM

The Fossil software configuration management tool is used in certain Altair Engineering software to provide versioning capabilities for various configuration files. Fossil is released under a 2-clause BSD license:

Copyright 2007 D. Richard Hipp. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and contributors and should not be interpreted as representing official policies, either expressed or implied, of anybody else.

GD Graphics Library

The built-in graphics capabilities of the software take advantage of the GD Graphics Library, which is available from <http://www.boutell.com/gd>. More detail on the GD Graphics Library is available at <http://www.boutell.com/gd/index.html> on the <http://www.boutell.com> web site.

The GD Graphics Library distribution used is *gd-2.0.33*.

GNU Utilities for Windows

The LicenseMonitor™ LicenseManager functionality employs CVS (Concurrent Versions System) to maintain a history of changes to licensing files. It also makes use of the file program in order to determine file types. This utility requires the magic, regex, and zlib DLLs on Windows. On Unix-based systems, these utilities and their required libraries are normally already present in the operating system loadset, but they are not normally present on Windows systems. The cvs, file, and gzip programs are included in the Windows distribution of Altair Engineering software, as are the aforementioned DLLs that are required for the file program.

The distribution of CVS included in this version of Altair Engineering software is 1.11.22. The distribution of file (and magic) included in this version of Altair Engineering software is 5.03.3414.

The distribution of regex2 included in this version of Altair Engineering software is 2.7.2853. The distribution of zlib1 included in this version of Altair Engineering software is 1.2.8.

The distribution of gzip included in this version of Altair Engineering software is 1.2.4.

All of these software components are released under the [GNU Public License \(GPL\)](#).

Graphviz -- Graphical Visualization Software

Altair Engineering software makes use of Graphviz libraries as part of its console Graphical User Interface (GUI). The version of Graphviz distributed with this version of Altair Engineering software is 2.38.0. The Graphviz license can be viewed at <http://www.graphviz.org/License.php>.

Altair Engineering has modified the Graphviz libraries for its use with Altair Engineering products. To obtain a copy of the modified Graphviz libraries, please contact <https://www.pbsworks.com/ContactSupport.aspx>.

history

Version: 4.9.0

MIT License

Copyright (c) React Training 2016-2018

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

hwloc

Altair Engineering uses version 2.0.4 of the [hwloc](#) library. *hwloc* is distributed under the [New BSD License](#), listed below.

Copyright © 2004-2006 The Trustees of Indiana University and Indiana University Research and Technology Corporation. All rights reserved.

Copyright © 2004-2005 The University of Tennessee and The University of Tennessee Research Foundation. All rights reserved.

Copyright © 2004-2005 High Performance Computing Center Stuttgart, University of Stuttgart. All rights reserved.

Copyright © 2004-2005 The Regents of the University of California. All rights reserved.

Copyright © 2009 CNRS

Copyright © 2009-2016 Inria. All rights reserved.

Copyright © 2009-2015 Université Bordeaux

Copyright © 2009-2015 Cisco Systems, Inc. All rights reserved.

Copyright © 2009-2012 Oracle and/or its affiliates. All rights reserved.

Copyright © 2010 IBM

Copyright © 2010 Jirka Hladky

Copyright © 2012 Aleksey Saushev, The NetBSD Foundation

Copyright © 2012 Blue Brain Project, EPFL. All rights reserved.

Copyright © 2013-2014 University of Wisconsin-La Crosse. All rights reserved.

Copyright © 2015 Research Organization for Information Science and Technology (RIST). All rights reserved.

Copyright © 2015-2016 Intel, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

[incr Tcl]

The [incr Tcl] software is used in constructing single-file distributables of Altair Engineering software. The distribution of [incr Tcl] included in this version of Altair Engineering software is that which is included in the TclKit 1.8.5 distribution. [incr Tcl] is licensed under a BSD-style license:

This software is copyrighted by Lucent Technologies, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

jwt-decode

Version: 2.2.0

The MIT License (MIT)

Copyright (c) 2015 Auth0, Inc. <support@auth0.com> (<http://auth0.com>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Metakit

The Metakit software is used in constructing single-file distributables of Altair Engineering software. The distribution of Metakit included in this version of Altair Engineering software is that which is included in the TclKit 1.8.5 distribution. Metakit is licensed under an MIT-style license:

Copyright (c) 1996-2020 Jean-Claude Wippler

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

mysqлтcl

Altair Engineering makes use of the [mysqлтcl](#) MySQL Tcl Interface library. Distribution version: 3.05.

Copyright (c) 1994, 1995 Hakan Soderstrom, Enskede, Sweden and Tom Poindexter, Denver, Colorado

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice and this permission notice appear in all copies of the software and related documentation.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL HAKAN SODERSTROM OR SODERSTROM PROGRAMVARUVERKSTAD AB BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

nginx

The nginx software is used by Altair Engineering software to provide the entry point for HTTP/HTTPS communications between the Altair Engineering web server and remote web clients. The distribution of nginx included in this version of Altair Engineering software is 1.9.2. nginx is licensed under a [BSD-style license](#).

Oat++

Copyright 2021 Altair Engineering, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Octtools

This software also uses packages from the Octtools-5.1 distribution from UC Berkeley, namely: *errtrap options st timer utility*

Octtools are covered by the following copyright notice:

Oct Tools Distribution 5.1

Copyright © 1988, 1989, 1990, 1991 Regents of the University of California.

All rights reserved.

Use and copying of this software and preparation of derivative works based upon this software are permitted. However, any distribution of this software or derivative works must include the above copyright notice.

This software is made available AS IS, and neither the Electronics Research Laboratory or the University of California make any warranty about the software, its performance or its conformity to any specification.

Suggestions, comments, or improvements are welcome and should be addressed to:

octtools@eros.berkeley.edu

These packages have been developed at UC Berkeley from 1985 to 1991 by the Berkeley CAD group. Special thanks to David Harrison, Tom Laidig, Peter Moore, Richard Rudell, Rick Spickelmeir.

OpenLDAP

The [OpenLDAP](#) client library is utilized by the vovserver binary to perform LDAP-based authentication. OpenLDAP software is released under the [OpenLDAP Public License](#):

The OpenLDAP Public License Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,
2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and
3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2020 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

OpenSSL

Altair Engineering software makes use of OpenSSL libraries to provide secure communications to various services. The version of OpenSSL distributed with this version of Altair Engineering software on Unix-based platforms is 1.0.2a, and on Windows, is 1.0.2e. The OpenSSL license can be viewed at <https://www.openssl.org/source/license.html>.

pgtcl-ng

Altair Engineering software makes use of the [pgtcl-ng](#) PostgreSQL Tcl Interface library. Distribution version: 2.0.0.

This is the license for pgtcl-ng:

Portions Copyright © 2004-2020, L Bayuk

Portions Copyright © 1996-2020, PostgreSQL Global Development Group

Portions Copyright © 1994, The Regents of the University of California

PostgreSQL is Copyright © 1996-2020 by the PostgreSQL Global Development Group and is distributed under the terms of the license of the University of California below.

Postgres95 is Copyright © 1994-5 by the Regents of the University of California.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

PhantomJS

Altair Engineering software makes use of the PhantomJS utility for parsing HTML files and working with JavaScript objects from the command line. Distribution version: 2.1.1.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PostgreSQL

Altair Engineering software makes use of [PostgreSQL](#) database for its back-end data storage reporting needs. PostgreSQL database is released under the [PostgreSQL License](#). Distribution version: 9.6.1.

PostgreSQL is released under the PostgreSQL License, a liberal Open Source license, similar to the BSD or MIT licenses.

PostgreSQL Database Management System (formerly known as Postgres, then as Postgres95)

Portions Copyright (c) 1996-2020, The PostgreSQL Global Development Group

Portions Copyright (c) 1994, The Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

react

Version: 16.9.0

MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-bootstrap

Version: 1.0.0-beta.12

The MIT License (MIT)

Copyright (c) 2014-present Stephen J. Collings, Matthew Honnibal, Pieter Vanderwerff

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-bootstrap-table-next

Version: 4.0.3

MIT License

Copyright (c) 2018 react-bootstrap-table2

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-cookies

Version: 0.1.1

MIT License

Copyright (c) 2019 Bu Kinoshita <bukinoshita@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-data-grid

Version: 7.0.0-canary.15

The MIT License (MIT)

Original work Copyright (c) 2014 Prometheus Research

Modified work Copyright 2015 Adazzle

For the original source code please see <https://github.com/prometheusresearch/react-grid>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-dom

Version: 16.11.0

MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-grid-layout

Version: 0.16.6

The MIT License (MIT)

Copyright (c) 2016 Samuel Reed

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-items-carousel

Version: 2.8.0

Copyright (c) 2017, Gajus Kuizinas (<http://gajus.com/>)

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Gajus Kuizinas (<http://gajus.com/>) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANY PARTY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, REDISTRIBUTIONS OF SOURCE CODE) ARISING IN ANY MANNER FROM THE USE OF THIS SOFTWARE. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-page-visibility

Version: 6.4.0

The MIT License (MIT)

Copyright (c) Gilad Peleg <gilad@giladpeleg.com> (<https://www.giladpeleg.com>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-resize-detector

Version: 5.2.0

The MIT License (MIT)

Copyright (c) 2016 Denis Rul

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF

CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-responsive-modal

Version: 4.0.1

The MIT License (MIT)

Copyright (c) 2020 Léo Pradel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-router-dom

Version: 5.1.2

MIT License

Copyright (c) React Training 2016-2018

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-scripts

Version: 3.2.0

MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-select

Version: 3.1.0

MIT Licensed. Copyright (c) Jed Watson 2019.

react-sparklines

Version: 1.7.0

The MIT License (MIT)

Copyright (c) 2015 Boris Yankov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-time-ago

Version: 6.2.1

(The MIT License)

Copyright (c) 2016 @catamphetamine <purecatamphetamine@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-tooltip

Version: 3.11.1

The MIT License (MIT)

Copyright (c) 2015 Wang Zixiao

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

reactstrap

Version: 8.0.1

The MIT License (MIT)

Copyright (c) 2016-Present Eddy Hernandez, Chris Burrell, Evan Sharp

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Reprise

This software includes and ships with Reprise Software Licensing Manager (RLM) binaries (v9.3rel) which are used to validate Altair Engineering Licenses.

Copyright © 2006-2020, Reprise Software, Inc. All Rights Reserved.

Reprise License Manager, OpenUsage, and Transparent License Policy are all trademarks of Reprise Software, Inc. RLM contains software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org>) Copyright © 1998-2020 The OpenSSL Project. All rights reserved.

<http://www.reprisesoftware.com>

rxjs

Version: 6.3.3

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- a. You must give any other recipients of the Work or Derivative Works a copy of this License; and
- b. You must cause any modified files to carry prominent notices stating that You changed the files; and
- c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2015-2018 Google, Inc., Netflix, Inc., Microsoft Corp. and contributors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SQLite

The SQLite database is public domain as described in <http://www.sqlite.org/copyright.html>. Distribution version: 3.8.3.

Swagger UI

The VOV REST API interactive documentation is enabled through the use of the Swagger UI library. Swagger is licensed under the Apache 2.0 license, as described in <https://swagger.io/license/>. Distribution version: 3.18.3.

Tcl/Tk

The graphical user interface is implemented with Tcl 8.6.5/Tk 8.6.5.

Tcl/Tk includes the following copyright notice:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 © (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 © (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

tclrmq

The tclrmq library is used to interface with an externally-hosted RabbitMQ message broker server. The distribution of tclrmq included in this version of Altair Engineering software is 1.3.8.

Copyright (c) 2017, FlightAware LLC All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the FlightAware LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TclVfs

The TclVfs software is used in constructing single-file distributables of Altair Engineering software. The distribution of TclVfs included in this version of Altair Engineering software is that which is included in the TclKit 1.8.5 distribution. TclVfs is licensed under a BSD-style license:

This software is copyrighted by the Vince Darley, and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

tcpkill

The utility 'tcpkill' has been included in the release, under the name 'vovtcpkill'. The utility comes with this copyright notice.

Copyright (c) 1999-2020 Dug Song <dugsong@monkey.org>, et al. All rights reserved, all wrongs reversed.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A

PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TkConsole

Copyright 1995-2020 Jeffrey Hobbs, jeff(a)hobbs(.)org

Release Info: v2.4, CVS v1.82 2004/11/11 17:22:13

Documentation available at: <http://tkcon.sourceforge.net>

treemap-squarify

Version: 1.0.1

MIT License

Copyright (c) 2020 Clément Bataille

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

TWAPI

On Windows, the TWAPI Tcl library is used to interface with the Windows API for various functions, such as service management. Distribution version: 3.1.17.

TWAPI includes the following copyright notice:

Copyright (c) 2003-2020, Ashok P. Nadkarni

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name of the copyright holder and any other contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

XYNTService

The XYNTService utility is a general-purpose Windows service wrapper that allows for the creation of custom services for programs that do not have native service support. This utility is only included in the Windows distribution of Altair Engineering software. The distribution of XYNTService included in this version of Altair Engineering software is dated 02.22.2008 and is released under the [Code Project Open License \(CPOL\)](#).

yup

Version: 0.27.0"

The MIT License (MIT)

Copyright (c) 2014 Jason Quense

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Zlib

The 'zlib' compression library provides in-memory compression and decompression functions, including integrity checks of the uncompressed data. Distribution version: 1.2.8.

Zlib includes the following copyright notice:

Copyright © 1995-2020 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org

Mark Adler madler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files <http://www.ietf.org/rfc/rfc1950.txt> (zlib format), <http://www.ietf.org/rfc/rfc1951.txt> (deflate format) and <http://www.ietf.org/rfc/rfc1952.txt> (gzip format).

Contents

Intellectual Property Rights Notice	ii
Technical Support	vi
3rd Party Licenses	viii
Altair Accelerator User Tutorials	39
Use Accelerator Help.....	40
Enable CLI Access on UNIX.....	42
Troubleshooting the UNIX Setup.....	45
Enable CLI Access on Windows.....	46
Verify Context Is Working.....	47
Run Basic Jobs.....	49
Get Summary Information.....	51
Run Jobs with Various Options.....	54
Submit Multiple Jobs at Once: -f <file>.....	54
Use Environments: -e <env>.....	54
Use of Resources: -r <res1 res2 ...>.....	55
Wait for Jobs.....	55
Specify Name of logfile: -l <logfile>.....	56
Job Control.....	57
Rerun Jobs.....	59
Get Detailed Information about a Job.....	60
Monitor Jobs, Taskers and Resources.....	61
Invoke the GUI.....	63
Use the Web Browser.....	65
Troubleshooting.....	66
Altair Accelerator Administrator Tutorials	67
Verify the License File.....	68
Start/Stop the RLM License Server.....	70
Start a Test Queue.....	71
Start/Stop Accelerator.....	74
Browser-based Setup.....	76
Configure Policy - FairShare and Other Parameters.....	77
Server Configuration Parameters.....	78
Advanced Policy Configuration.....	79
Configure Resources.....	81
Configure Security.....	84
Configure Taskers.....	86
Configure an Environment.....	91
Logical Names (Equivalences).....	96

Resource Management.....	99
Altair Monitor-Basic Setup.....	99
Configure and Manage Monitor-basic.....	100
vtk_flexIm_monitor Procedures.....	101
Resource Throttling.....	101
Upgrade Accelerator.....	103
Index	106

Altair Accelerator User Tutorials

In the following User Tutorials, you will experiment with Accelerator on most topics that a user would be most interested in, including submitting jobs, tracking job information, analyzing and solving common problems, etc.

This chapter covers the following:

- [Use Accelerator Help](#) (p. 40)
- [Enable CLI Access on UNIX](#) (p. 42)
- [Enable CLI Access on Windows](#) (p. 46)
- [Run Basic Jobs](#) (p. 49)
- [Get Summary Information](#) (p. 51)
- [Run Jobs with Various Options](#) (p. 54)
- [Job Control](#) (p. 57)
- [Rerun Jobs](#) (p. 59)
- [Get Detailed Information about a Job](#) (p. 60)
- [Monitor Jobs, Taskers and Resources](#) (p. 61)
- [Invoke the GUI](#) (p. 63)
- [Use the Web Browser](#) (p. 65)
- [Troubleshooting](#) (p. 66)

Use Accelerator Help

Accelerator documentation is available in HTML and PDF format.

Access the Help when Accelerator is Running

When Accelerator is running, it displays the documentation through its browser interface. To access it from browser, you need to know which host and port Accelerator is running on. Ask your administrator, or find the URL for Accelerator with the following command:

```
% Accelerator cmd vovbrowser  
http://comet:6271/project
```

In the example below, assume Accelerator is running on host comet, port 6271. The URL for Accelerator is:


```
http://comet:6271
```

To get the entire suite of Altair Accelerator documents, including FlowTracer™, Accelerator™, Monitor™ and the VOV subsystem, use the following URL:

```
http://comet:6271/doc/html/bookshelf/index.htm
```

Access the Help when Accelerator is not Running

All the documentation files are in the Altair Accelerator install directory, so you can access them even if vovserver is not running. To do this, open `/installation_directory/common/doc/html/bookshelf/index.htm` in your browser.

 **Tip:** Bookmark the above URL for future reference.

Access the Help PDF Files

Altair Accelerator also provides PDF files for each of the guides. All the PDF files are in the directory `/installation_directory/common/doc/pdf`

Access the Help via the Command Line

The main commands of Accelerator are `nc` and `ncmgr`, with some subcommands and options. You can get usage help, descriptions and examples of the commands by running the command without any options, or with the `-h` option. For example,

```
% nc info -h  
nc:  
nc:  NC INFO:  
nc:  Get information about a specific job or list of jobs.  
nc:  USAGE:  
nc:  % nc info <jobId> [options]...  
nc:  -h          -- Show this message  
nc:  -l          -- Show the log file  
nc:
```


Access the Help via the vovshow Command

Another source of live information is using the command `vovshow`. The following options are often useful:

<code>vovshow -env RX</code>	Displays the environment variables that match the regular expression <code>RX</code> provided.
<code>vovshow -fields</code>	Shows the fields known to the version of VOV in use.
<code>vovshow -failcodes</code>	Shows the table of known failure codes.

For example, to find a variable that controls the name of the stdout/stderr files, without knowing the exact name of that variable, the following command can be used:

```
% vovshow -env STD
VOV_STDOUT_SPEC          Control the names of file used to save stdout and
                          stderr. The value is computed by substituting
                          the substrings @OUT@ and @UNIQUE@ and @ID@.
                          Examples: % setenv VOV_STDOUT_SPEC
                          .std@OUT@.@UNIQUE@ % setenv VOV_STDOUT_SPEC
                          .std@OUT@.@ID@
```

The output provides a description of all the variables used by the FlowTracer system that include the substring "STD". In this example, the output result `VOV_STDOUT_SPEC`.

Enable CLI Access on UNIX

This section explains how to set up a UNIX user's shell environment to have a proper context for the user to run installed Altair Accelerator programs from the command line. The programs that are run from the command line are called the CLI commands.

When a UNIX user logs into the system, a login script is executed which sets up their working environment. The default shell for the user determines what login script will run. A csh shell uses a `.cshrc` login script. A bash shell uses a `.profile` login script. The best way to set up a user's shell environment to run Altair Accelerator programs is to change the login script to properly set the user's working environment.

Altair Accelerator products require that the PATH environment variable be set to include the directories in the Altair Accelerator release which hold programs that will be run from the command line. Also, there are two environment variables that need to be set so that when an Altair Accelerator program is run, it will know where the release is installed and know what type machine it is running on. These two environment variables are VOVDIR and VOVARCH.

The Altair Accelerator product installation provides a helper file that can be sourced in order to set the needed environment variables to values that are appropriate for the local situation.

The intended use of the helper file is to have it sourced from a user's shell login script so that the user gets a proper environment without doing anything extra.

There is one helper file for each of the shell types that exist on UNIX. One helper script file is in the csh syntax and the other is in the shell syntax.

- `vovrc.csh`
- `vovrc.sh`

You can change each user's shell login script to source the file that is appropriate for the particular shell that they use.

Shell	Instructions
C-shell, tcsh	Add the following line to your <code>.cshrc</code> csh login script file: <pre>source /<install_path>/<version>/<platform>/ common/etc/vovrc.csh</pre>
sh, ksh, bash, zsh	Add the following line to your <code>.profile</code> shell login script file: <pre>./<install_path>/<version>/<platform>/common/etc/ vovrc.csh</pre>

Verify Access to Altair Accelerator Products

After making the changes to source the helper files, and then logging in, you can check that needed environment variables are set properly by looking at environment variables to note that the PATH value contains a reference to the folders in the release which hold programs, and to note that VOVDIR and VOVARCH are set. VOVDIR should contain the path to the installation. VOVARCH should contain the name that matches the machine type it is running on.

1. At the command prompt, enter the following:

```
% echo $PATH
% echo $VOVDIR
% echo $VOVARCH
```

Beyond just looking, you can try running a Altair Accelerator product program as a positive test that the context is set up properly. One program that is simple to run and is available with all Altair Accelerator products is `vovarch`. This program reports on the machine type on which it is running. A side effect is that it tests that needed environment variables are set properly. If the environment is valid, the program will run and report the correct machine type. If the environment is not valid, the program will not work.

2. From the shell prompt, run the command `vovarch`.

```
% vovarch
```

You should get a response that is similar to one of the following, "linux64", or "macosx". This indicates the program was found, it ran, and it produced an expected output that matches your UNIX environment.

You have successfully set up the environment and verified it is correct.

3. If the response is `Command not found`, then the working environment does not have a VOVARCH setting for the programs in the Altair Accelerator products install area. If this is the case, review the steps to make sure the helper file from Altair Accelerator is being sourced correctly.

```
% vovarch
linux64
... or ...
macosx
```

or

```
% vovarch
vovarch: Command not found.
```

Enable Altair Accelerator for Non-Interactive Shells

It is possible to have the shell login script build a different working environment for interactive and non-interactive shells. The non-interactive environment is used when you run a batch script.

A common way this is done is to do an early exit from the login script file for non-interactive shells. For CSH, this can be done by testing for the existence of the shell variable "prompt".

Early exit from non-interactive shell login CSH script:

```
# This is a fragment of .cshrc.  
:  
# Batch scripts can skip doing actions needed by interactive scripts.  
if ( ! $?prompt ) then exit  
:  
# Below are actions needed by interactive scripts.  
:
```

If this exit happens early in the script, before sourcing the Altair Accelerator helper file, then the environment variables will not be set for the non-interactive shell.

A batch script needs access to Altair Accelerator products. This means that the non-interactive shell needs to have the needed environment variables set.

You should place the code that sources the helper file from Altair Accelerator early in the shell login script, before any logic causes the script to differ between interactive and batch shells.

Source helper file before exit in login CSH script:

```
# This is a fragment of .cshrc.  
:  
# Altair Accelerator env vars are needed by batch scripts.  
source /<install_path>/<version>/<platform>/common/vovrc.csh  
:  
# Batch scripts can skip doing actions needed by interactive scripts.  
if ( ! $?prompt ) then exit  
:  
# Below are actions needed by interactive scripts.  
:
```

Enable the Shell to Communicate With a Running Product Server

If you need to issue commands that will communicate to a running product instance, the instance will need to be "enabled", which involves setting certain environment variables that point the commands to the location of the running vovserver.

```
% vovproject enable instance-name
```

Some common instance names are "licmon" (for Monitor), "vnc" (for Accelerator), and "wx" (for Accelerator Plus). Instance names can be anything though since they are user-definable upon first start of each product.

Troubleshooting the UNIX Setup

An earlier section of this manual explained the importance of editing the `.cshrc` file so that batch shells would have the proper environment for running Altair Accelerator products.

The following is a command line to verify that the `.cshrc` file is set properly for batch shells. This runs the `vovarch` command within a batch context.

```
% csh -c vovarch
```

If this fails, the `.cshrc` file is not edited properly to enable access for batch shells. Review the details of the earlier topic on editing the `.cshrc` to enable access to batch shells.

Enable CLI Access on Windows

This section explains how to set up a Windows user's command prompt environment to have the proper context for the user to run installed Altair Accelerator programs from the command line. The programs that are run from the command line are called the CLI commands.

Altair Accelerator products require that the PATH environment variable be set to include the directories in the Altair Accelerator release which hold programs that will be run from the command line. Also, there are two environment variables that need to be set so that when an Altair Accelerator program is run, it will know where the release is installed and know what type of machine it is running on. These two environment variables are VOVDIR and VOVARCH.

There are two methods for setting the correct environment. Both involve running the same context-setting bat script. This context-setting bat script establishes the correct environment variables for the local situation, reflecting where Altair Accelerator products are installed.

 **Note:** This operation to set the environment is not required to use every Altair Accelerator feature on Windows. This operation is only needed to enable using the CLI commands from the command prompt.

Method 1: Use Windows Explorer to Set Command Line Environment

1. Using Windows Explorer, navigate to the Altair Accelerator installation directory.
2. Enter the `win64/startup` folder and double-click the `vovcmd.bat` script to run it. This will open a command prompt with the proper environment settings for the Altair Accelerator and scripts to work.

When `vovcmd.bat` runs, it will execute the `win64/bat/vovinit.bat` script as part of what it does. The following section covering Method 2 explains what `win64/bat/vovinit.bat` does when it runs. It does the same thing when run by either method.

Method 2: Using Windows Command Prompt to Set Command Line Environment

1. In a command prompt window, navigate to the Altair Accelerator installation directory using the `cd` command.
2. Change directory to the `win64/bat` folder with `cd` and run the `vovinit.bat` script. This will establish the needed environment for the open command prompt.

When `win64/bat/vovinit.bat` runs, it figures out needed environment variables and sets them, based upon where it is located. In particular, it sets `VOVDIR` to be the path to where Altair Accelerator is installed. It then executes another initialization script, `$VOVDIR/win64/local/vovinit.bat`, if it exists.

This is an initialization script that you can create and modify to perform site specific activities customized for your local configuration and usage. You can add commands to the `local/vovinit.bat` file that you want to run whenever a user starts up a command prompt.

After running both `vovinit.bat` scripts, the context of the command prompt has the correct environment needed so that CLI commands will work correctly with the installed Altair Accelerator.

Customize Actions Needed to Enable Access to Altair Accelerator Products on Windows

You can add commands to the `win64/local/vovinit.bat` file that perform specific operations that enable your Windows users to access Altair Accelerator programs properly, or to set up things on the machine to follow a standard convention.

You could add operations that perform actions such as these, and others:

- Mounting network drives
- Setting environment variables for local needs
- Establishing time synchronization

Example of a custom `$VOVDIR/win64/local/vovinit.bat`

```
rem -- Mount network drives:
rem -- In this example we mount the Altair Accelerator installation on drive v:
if not exist v:\nul net use v: \\somehost\rttda

rem -- Set locally useful environment variables.
set VNCSWD=v:\vnc
set DLOG=d:\dailylog

rem -- Set Windows time from server on local network
rem -- Put this last; it may fail if lacking time set privilege
net time \\timehost /set /y
```

Verify Context Is Working


When you have a command prompt open and expect that it has a context for accessing Altair Accelerator programs, check that the environment is set by looking at environment variables to note that the `PATH` value contains a reference to the folders in the release which hold programs, and to note that `VOVDIR` and `VOVARCH` are set. `VOVDIR` should contain the path to the installation. `VOVARCH` should contain the name that matches the machine it is running on.

Beyond just looking, you can try running a program as a positive test that the context is set up properly. One program that is simple to run and is available with all Altair Accelerator products is

vovarch. This program reports on the machine type on which it is running. A side effect is that it tests that needed environment variables are set properly. If the environment is valid, the program will run and report the correct machine type. If the environment is not valid, the program will not work.

Run `vovarch` to verify the environment is set ok:

```
c:\ > vovarch  
win64
```

 **Note:** The output will always show **win64** when running on any of the Microsoft Windows operating systems. This result is expected. It reports that we are on a generic "windows" architecture and indicates that the context is working.

If you are not able to verify that the context is valid, check the details within the `/folder/where/Altair Accelerator/software/is/installed/win64/bat/vovinit.bat` file.

Enable the Command Prompt to Communicate With a Running Product Server

If you need to issue commands that will communicate to a running product instance, the instance will need to be "enabled", which involves setting certain environment variables that point the commands to the location of the running vovserver.

```
c:\ > vovproject enable instance-name
```

Some common instance names are "licmon" (for Monitor), "vnc" (for Accelerator), and "wx" (for Accelerator Plus). Instance names can be anything though since they are user-definable upon first start of each product.

Run Basic Jobs

Submitting jobs to Accelerator is quite simple: just use `nc run` followed by the command you would use without Accelerator.

Run a Sleep Job


```
% cd # Go to your home directory
% nc run sleep 10
Resources= linux
Env = D(VOV_ENV_SOURCE=vnc_logs/envdixin36362.env)
Command = vw vwrap sleep 10
Logfile = vnc_logs/20021230/103409.13784
JobId = 04194422
nc: message: Scheduled jobs: 1 Total estimated time: 0s
```

Run Sleep Job with -r Option to Override Default Resources

By default, `nc run` takes the architecture of the machine from which the job is submitted as the job's resource. This can be overridden with `-r` option when needed.

This job requires no resource.

```
% nc run -r "" -- sleep 10
Resources=
Env = D(VOV_ENV_SOURCE=vnc_logs/envdixin36362.env)
Command = vw vwrap sleep 10
Logfile = vnc_logs/20021230/103653.13794
JobId = 04194459
nc: message: Scheduled jobs: 1 Total estimated time: 0s
```

 **Note:** Since the `-r` option accepts multiple arguments, you need to terminate the resource list explicitly with another option, or `--` if no other options are needed.

This job requires `spice_license`.

```
% nc run -r spice_license -- sleep 10
Resources= spice_license
Env = D(VOV_ENV_SOURCE=vnc_logs/envdixin36362.env)
Command = vw vwrap sleep 10
Logfile = vnc_logs/20021230/103948.13803
JobId = 04194497
nc: message: Scheduled jobs: 1 Total estimated time: 0s
```

Run Job with -e Option to Override Default Environment

By default, `nc run` takes a snapshot of the environment as the job's environment and uses this to run your job. You can override this with `-e` option to use a named environment.

```
% nc run -e BASE sleep 10
Resources= linux
Env = BASE
Command = vw vwrap sleep 10
Logfile = vnc_logs/20021230/122737.14946
```

```
JobId      = 04195924  
nc: message: Scheduled jobs: 1          Total estimated time: 0s
```

Use the command `vel` to list all the available environments:

```
% vel  
vel: message: Environment directories:  
1 /remote/release/VOV/7.0u3/linux64/local/environments  
1 . tcl BASE          UNIX utilities, X windows, and VOV.  
1 . tcl D             Define variables: Usage: ves "+D(VAR1=value1,...)"  
1 * tcl DEFAULT      Just a name for whatever you already have.  
1 . tcl HSIM         Fake Nassda hsim env for Virage testing  
1 . csh SYNOPSIS     Synopsys tools
```

Scripts that implement named environments may be written in `csh`, `sh`, or `Tcl` syntax. Cross-platform environments between Windows and UNIX/Linux must be written in `Tcl`. You can use named environments from the command line using the command `ves`.

Get Summary Information

nc summary, nc list and nc info *jobId* will be the three commands that you use frequently to get information about the jobs managed by Accelerator.

Get Summary of All My Jobs: nc summary

```
% nc summary
NC Summary For User dextrin
TOTAL JOBS      11      Duration: 23s
  Done          1
  Idle          0
  Queued        5
  Running       2
  Failed        3

  JOBS  GROUP  TOOL      WAITING FOR...
    1   alpha  sleep     'aa'
    3   alpha  sleep     'linux'
```

In the above example, the command displays summary information for user dextrin. There are a total of 11 jobs, 1 of which is Done, 2 Running, 3 Failed, etc. In the bottom part, it shows the summary for jobs that are queued, that is, there is 1 sleep job queued because it is waiting for resource 'aa', and there are 3 sleep jobs waiting for resource 'linux'.

To get an idea of what is going on in the whole Accelerator system, the following shows all the jobs and what the jobqueue buckets are waiting for. Note that Accelerator subcommands may be abbreviated.

```
% nc sum -a -b
```

Get a List of My Jobs: nc list

This command, without any option, displays the last 20 of your jobs in the format of "jobId status command".

```
% nc list
04193437 Done      sleep 1
04193911 Done      sleep 5
04193913 Failed    sleep 10
04193917 Failed    sleep 20
04193919 Idle      sleep 60
04193937 Running   sleep 10
04193943 Queued    sleep 56
```

Status Meaning

In Accelerator, each job is assigned an "Accelerator Computing Status" which is defined as follows:

Status	Color *	Explanation
Queued	Cyan	The job is scheduled to be executed.

Status	Color *	Explanation
Running	Orange	The job is currently executing
Done	Green	The job ran successfully
Failed	Red	The job ran and failed.
Idle	BlueViolet	The job needs to be run, but it is not scheduled.

*(Colors may look different on some systems)

Some Options about nc list

The `nc list` command has several useful options. They include:

Get detailed usage of this command:

```
% nc list -h
```

List all jobs, including others' jobs:

```
% nc list -a
04193437 Done    alpha    dextrin  rhino    sleep 1
04193898 Done    users    integ    rhino    sleep 2
04193939 Done    alpha    dextrin  rhino    sleep 15
04193941 Done    alpha    dextrin  rhino    sleep 20
04193943 Done    alpha    dextrin  rhino    sleep 60
```

Control output format, show job Id the (first) tool of job only:

```
% nc list -O "@ID@ @TOOL@"
04193437 sleep
04193898 sleep
04193939 sleep
04193941 sleep
04193943 sleep
```

In the above example, we use "fields", i.e., ID and TOOL, surrounded by two "@" signs, to format strings.

Get Detailed Information About a Job: nc info jobId

Without options, this command displays the basic information about the job, like user, group, directory, command, environment, queue time, etc.

```
% nc info 04193937
Id,User,Group    04193937,dextrin,alpha
Environment      D(VOV_ENV_SOURCE=vnc_logs/envdextrin36362.env)
Directory        ${HOMES}/dextrin
Command          sleep 10
Status           Done
  Host           rhino
  QueueTime      1s
```

```
Duration      10s
Age           20m37s
AutoForget    1
```

With option `-l`, this command shows the contents of the log file.

```
% nc info -l 04193937
Log file is: '${HOMES}/dexin/vnc_logs/20021230/095337.13512.3'
vwrap: message: Start date: Mon Dec 30 09:53:38 PST 2002
vwrap: message: On host: rhino
vwrap: message: Sourcing environment vnc_logs/envdexin36362.env
vwrap: message: Running: 'sleep 10'
vwrap: message: Exit status: 0
vwrap: message: End   date : Mon Dec 30 09:53:48 PST 2002
```

Run Jobs with Various Options

In this section, we will exercise some options in `nc run` command.

To get the detailed usage, use the following command:

```
% nc run -h
```

Submit Multiple Jobs at Once: `-f <file>`

1. Prepare a file with one command on each line. Empty lines are ignored and lines that begin with `#` are considered comments.

```
# Example of file used to submit multiple jobs at once.  
sleep 10  
sleep 11  
sleep 12  
sleep 13
```

2. Use the option `-f` to specify the command file, as in the following example:

```
% nc run -f commandFile
```

All jobs submitted with this method share the same environment, the same resources, and are scheduled at the same priority level. Each job has its own ID.

Use Environments: `-e <env>`


1. Get a list of all available environments:

```
% vel  
vel: message: Environment directories:  
1 /<install_path>/local/environments  
1 . tcl BASE UNIX utilities, X windows, and Flowtracer.  
1 . tcl D Define vars: Usage: ves "+D(VAR1=value1,...)"  
1 . tcl DEFAULT Just a name for whatever you already have.  
1 . csh SPICE The Analog simulator SPICE3.  
1 . csh JAVA JAVA Development Environment (1.2.2)
```

2. Use proper environment(s) to submit jobs, for example:

```
% nc run -e BASE sleep 10  
% nc run -e BASE+SPICE spice chip.spi
```

Use of Resources: -r <res1 res2 ...>

 **Note:** Since this option accepts multiple arguments, you need to terminate the resource list explicitly with another option, or "--" if no other option is needed.

1. Submit a job that only runs on Linux machine:

```
% nc run -r os=linux64 -- sleep 10
```

2. Submit a job that requires resources "hpux" and "hspice":

```
% nc run -r hpux hspice -- sleep 10
```

Wait for Jobs

Waiting for jobs is especially useful for scripts. By default, `nc run` returns immediately after you submit a job. This allows you to submit multiple jobs at once. There could be times when you want to run jobs in sequence, in which case option `-w` is very useful. With this option, `nc run` waits for the job(s) to finish (Done or Failed).

```
% nc run -w sleep 10
Resources= linux
Env       = D(VOV_ENV_SOURCE=vnc_logs/envdixin36362.env)
Command  = vw vwrap sleep 10
Logfile   = vnc_logs/20021231/111314.21781
JobId     = 04211346
vnc: message: Scheduled jobs: 1          Total estimated time: 0s
```

Wait for Jobs and Show Log File of the last job: -wl

With this option, `nc run` waits for the job(s) and shows the log file of the last job.

```
% nc run -wl date
Resources= linux
Env       = D(VOV_ENV_SOURCE=vnc_logs/envdixin36362.env)
Command  = vw vwrap date
Logfile   = vnc_logs/20021231/111530.21819
JobId     = 04211392
nc: message: Scheduled jobs: 1          Total estimated time: 0s
<<<STARTING ON rhino>>>
Tue Dec 31 11:15:31 PST 2002
<<<END OF LOG>>>
<<<EXIT STATUS 0>>>
```

Please only use `-wl` for jobs where you are actively monitoring the output. Such jobs listen to the vovservers event stream and are called 'notify clients'. They are more resource-intensive than plain batch jobs.

Wait for Jobs Using `nc wait`

You can also wait for jobs using the command `nc wait`. We will cover that in next tutorial "Job Control".

Specify Name of logfile: -l <logfile>

The default logfile name has the form of `./vnc_logs/date/time`. You can explicitly specify the logfile of the jobs you submitted by this option, for example:

```
% nc run -l /home/john/logs/log1.log sleep 10
```

Warning: Conflicts may occur if same log file is used for multiple jobs. New users are not recommended to use this option.

Job Control

Accelerator provides several commands of job controls, including wait, stop and forget.

Wait for Jobs

Besides option `-w` and `-wl` with `nc run` command, you can also wait for job(s) to finish (Done or Failed) after they are submitted using the command `nc wait`. Here are some examples:

Get usage help

```
% nc wait -h
```

Wait for a job

```
% nc run sleep 10
Resources= linux
Env       = D(VOV_ENV_SOURCE=vnc_logs/envdixin36362.env)
Command  = vw vwrap sleep 10
Logfile   = vnc_logs/20021231/140024.23307
JobId     = 04213283
nc: message: Scheduled jobs: 1          Total estimated time: 0s

% nc wait 04213283
```

Wait for all jobs in current directory

```
% nc wait -dir .
nc: message: Job 04193913 is already FAILED
nc: message: Job 04193915 is already FAILED
nc: message: Job 04193917 is already FAILED
nc: message: Job 04193919 is not scheduled
nc: message: Job 04194308 is not scheduled
nc: message: Job 04211259 is already FAILED
nc: message: Job 04211268 is not scheduled
nc: message: Job 04213283 is already VALID
nc: message: Job 04213297 is already VALID
nc: message: Exiting with status 2 (Failed jobs)
```

Wait for all jobs using tool spice

```
% nc wait -select "tool==spice"
```

In the above example we use "Selection Rules" to perform a "wait" on those jobs that satisfy that rule. This could be useful for other commands as well, including `nc list`, `nc forget`, etc. Refer to the *Altair Accelerator User Guide* for more information.

Forget Jobs

The Accelerator server remembers the jobs you submitted for some configurable time. You can explicitly forget them by `nc forget` command, which will delete all job information from the server database.

Get a list of jobs

```
% nc list
04146420 Done      sleep 1      #in the form of "jobId status command"
```

```
04146425 Done    sleep 5
04146427 Running sleep 10
04146429 Running sleep 15
04146431 Queued  sleep 20
04146433 Queued  sleep 60
```

Forget some of them

```
% nc forget 04146420 04146425
nc: message: Forgetting 2 jobs
```

List jobs again (notice those two are gone)

```
% nc list
04146427 Done    sleep 10
04146429 Done    sleep 15
04146431 Done    sleep 20
04146433 Done    sleep 60
```

Forget all my jobs

```
% nc forget -mine
nc: message: Forgetting 4 jobs
```

Stop Jobs

A job can be stopped when it is either Running or Queued. Stopping a job does not forget it from the server database. "Running" jobs will exit, and "Queued" jobs will be dequeued when you stop them.

Stop some jobs

```
% nc run sleep 60
Resources= linux
Env      = D(VOV_ENV_SOURCE=vnc_logs/envdixin36362.env)
Command  = vw vwrap sleep 60
Logfile  = vnc_logs/20021231/140803.23386
JobId    = 04213393
nc: message: Scheduled jobs: 1          Total estimated time: 0s

nc stop 04213393
nc: message: Stopping RETRACING job 04213393
```

Stop all my jobs

```
% nc stop -mine
nc: message: Stopping RETRACING job 04146627
nc: message: Stopping RETRACING job 04146629
nc: message: De-queuing  job 04146631
nc: message: De-queuing  job 04146633
```

Rerun Jobs

The `nc rerun` command initiates the scheduling and execution of jobs that are already in the server database. By default, only the jobs that are Idle or Queued are affected by this command. If you want to force the rerunning of jobs that are either Done or Failed, use the option `-F`.

Rerun a "Done" job won't do anything

```
% nc rerun 04146622
nc: message: Not rerun: 04146622
```

Force rerunning a "Done" job

```
% nc rerun -F 04146622
nc: message: Job 04146622 is already VALID.
nc: message: Scheduled jobs: 1          Total estimated time: 1s
```

Rerun "Idle" jobs

```
% nc rerun 04146631 04146633
nc: message: Scheduled jobs: 1          Total estimated time: 0s
nc: message: Scheduled jobs: 1          Total estimated time: 0s
```

Get Detailed Information about a Job

The command `nc` displays information about a job.

Get Detailed Information About a Job

The command `nc getfield` also gives information about a job, but in an undecorated form that is in scripts.

```
vnc: Usage Message

NC GETFIELD:
  Get one or all fields of one or more NC jobs.  Specify the jobID
  or use '!' for the most recent job in the current working directory.

  If the -J jobName option is given, only the first match
  is reported.  If there is no match, an error is reported.

OPTIONS:
  -h                Show this help.  You can also get the
                   usage message by specifying no option at all.
  -v                Increase verbosity.
  -J JOBNAME        Find first job with given JOBNAME
                   The search is restricted to the jobs that
                   belong to the current user.
                   This is significantly more expensive than
                   using jobIDs.  Use sparingly.
  -f field          Specify field when giving multiple jobIDs
  -showid           Show jobId
  -s                Same as -showid
  -sep STRING       Use STRING as separator (default is a single space)
  -tab              Use a TAB character as separator.

EXAMPLES:
  % nc getfield -h
  % nc getfield 01234455
  % nc getfield 00123445 jobclass
  % nc getfield ! status
  % nc getfield -J JOBNAME
  % nc getfield 01234455 0123458 -f jobclass
  % nc getfield -s 01234455 0123458 -f jobclass
```

Examples:

```
% nc getfield 00012345 jobclass
normal
% nc getfield 00012345 cputime
7.125
% nc getfield 00012345
... get list of all known fields (more than 100 of them)...
```

Monitor Jobs, Taskers and Resources

The activity of Accelerator can be monitored with a dialog.

The dialog is invoked with:

```
% nc monitor
```

The following is a list of the tabs available in the dialog:

TaskersGroups	The activity of tasker groups.
Taskers	The activity of taskers.
Taskers HW	The hardware offered by taskers.
Taskers Resources	The resources offered by taskers.
Who	Who is running jobs.
Running Jobs	The progress of running jobs.
Running Commands	The details of running commands.
Running Details	The details of running jobs.
Resources	The usage and availability of resources.
Queued Jobs	The jobs in the job queue.
Queue Buckets	The jobs in the job queue organized by groups of similar jobs (called 'buckets').
FairShare	The FairShare statistics.

The screenshot shows a window titled "nc VOV Monitors 2021.1.1-dev-91405 vnc". The window contains a menu bar with "File", "Options", "TaskersGroups", "Taskers", and "Resources". Below the menu bar is a tabbed interface with tabs for "TaskerGroups", "Taskers", "Tasker HW", "Tasker Resources", "Who", "Running Jobs", "Running Commands", "Running Details", "Resources", "Queued Jobs", "Queue Buckets", and "FairShare". The "Resources" tab is active, displaying a table with the following data:

	Type:Name	Total	InUse	Rsrvd	ooQ	Available	%Util.	MapsTo
1	License:AL002_HWPoll	unlim	0	0	0	unlim		
2	Limit:user0265_justo	1	0	0	0	1	99.12%	
3	License:AL002_HFSoli	100000	0	0	0	100000	0.00%	
4	License:AL003_HWPSO_	unlim	0	0	0	unlim		
5	Limit:user0418_justo	1	0	0	0	1	99.79%	
6	Limit:user0354_justo	1	0	0	0	1	97.64%	
7	Limit:user0100_justo	1	0	0	0	1	98.94%	
8	License:AL003_HWPSO_	unlim	0	0	0	unlim		
9	License:SolverNode	100000	0	0	0	100000	0.00%	
10	Limit:user1312_justo	1	0	0	0	1	98.16%	
11	License:AL003_HWPano	unlim	0	0	0	unlim		
12	License:AL003_SIMLAB	100000	0	0	0	100000	0.00%	
13	License:AL001_HWActi	unlim	0	0	0	unlim		

Figure 1:

Invoke the GUI

Job execution can be monitored with `nc gui`.

This command opens a monitoring tool; no interactive capabilities (such as configuration or running jobs) are provided. Interactive capabilities are available with `nc cmd vovconsole`.

nc gui

Show a grid view of the jobs in a specified set.

```
vnc: Usage Message

NC GUI:
  Show a grid view of the jobs in a specified set.
USAGE:
  % nc gui [OPTIONS] &
OPTIONS:
  With no options, the GUI shows all jobs of the current
  user.

  -all
  -a                -- Show all jobs.
  -u <user>        -- Show jobs for specified user.
  -s <SETNAME>
  -set <SETNAME>
  -setname <SETNAME> -- Show specified set.
  -timeout <TIMESPEC> -- Stop async update after this time (default 2h).

  -ss                -- Use subserver, if available.
  -noss              -- Do not user subserver, even if available. (Default)
  -submit            -- Activate job submission dialog
  -limitGui <N>     -- Override the limit of 3 max GUI per user.

  -batch <file>     -- Execute specified file after the GUI is ready

  -metrics           -- Show scheduler metrics.
  -metricsConfig <file> -- Use specified metrics configuration file.
  -slaves            -- Show compact slaves monitor.
  -taskers           -- Show compact taskers monitor.
  -fontsize <size>  -- Specify the normal font size. Default is 10.
                    Legal range is 3 to 36.

  -title <title>    -- Choose title of X11 window.

EXAMPLES:
  % nc gui &                -- Show all my jobs
  % nc gui -all &           -- Show all jobs.
  % nc gui -set SomeSetName -- Show specified set.

  % nc gui -submit          -- Job submission dialog.
  % nc gui -limitGui 5      -- Allow you to run up to 5 "nc gui" (default
3)
```

```
% nc gui -metrics & -- Show the scheduler metrics.
```

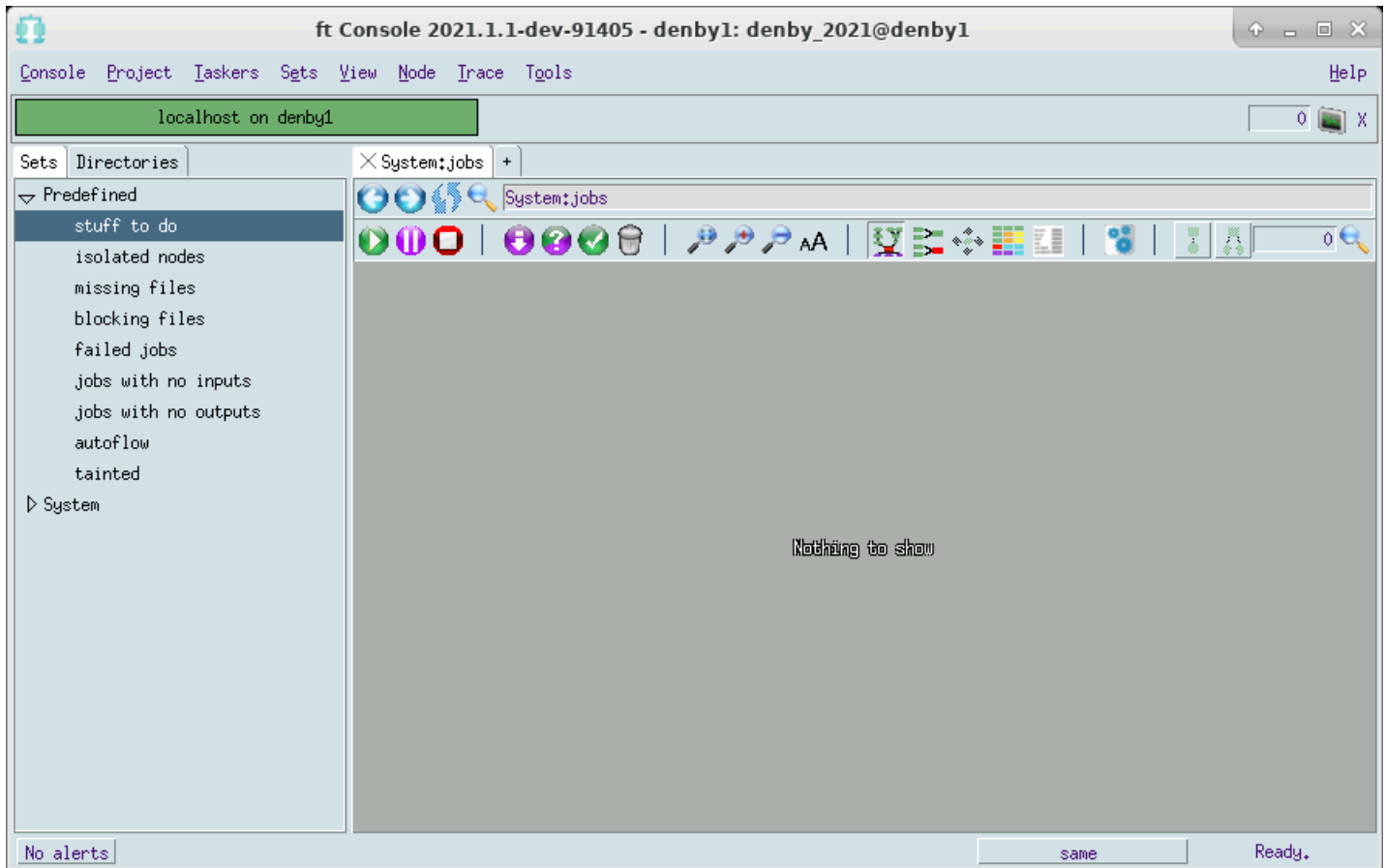


Figure 2: GUI that opens after entering nc cmd vovconsole &

Use the Web Browser

Find the URL for Accelerator

```
% nc cmd vovbrowser  
http://comet:6271/project
```

Use the Browser Interface

Enter the URL found above into your browser's location box. You will need to login unless your administrator has turned off authentication.

`http://your_host:your_port_number/project` is the home page of Accelerator. From this page, you can easily navigate to many other useful pages. Here we list some monitor pages that are similar to the ones from the GUI monitors.

Tasker page	From the home page, click Taskers
Workload pages (job queue and running jobs):	From the home page, click Workload For running jobs, under Workload, click Running jobs
Resources page	From the home page, click Resources
FairShare page	From the home page, under Workload, click Fairshare

Among the many useful pages, two report pages may be especially helpful.

Job queue report	From the home page, click Workload > Job Queue Reports
Resources reports	From the home page, click Resources > Resource Reports

Troubleshooting

This section covers typical problems that come up in Accelerator. The command `nc summary` will be useful here, as it tells you how many jobs are failed, queued, or idle, and what the queued jobs are waiting for. For example:

```
% nc summary
NC Summary For User bkring
TOTAL JOBS          0      Duration: 0s
Done                0
Idle                0
Queued              0
Running            0
Failed              0
```

My job won't start!

Often, your job won't start because it is waiting for a resource, usually a license, CPU, or memory.

To diagnose this, use the command:

```
nc info jobid
```

A job will only start if all resources requested by the job are available. If any resource is missing, the job will not start. You can look at the resources of all available vovtaskers to see if there is any that can run the job with the command:

```
nc host
```

Additionally, a vovtaskers must either be `READY` or `WRKNG` (have a free job slot) to accept jobs. Any other condition will prevent the vovtaskers from taking the job.

My job failed!

You can find the reason for job failure with the command:

```
nc info jobid
```

Some common failure conditions include:

- The job failure has nothing to do with Accelerator. Run the job without Accelerator to verify this.
- The job command doesn't exist, possibly because of a typo.
- You are using a wrong, nonexistent, or incomplete environment with the `-e`. In this case, `nc info jobid` will tell you that it cannot switch to the environment.
- You have failed to specify (or specified the wrong) architecture or memory usage. This can be done with the `-r` option. For example, `-r linux64` for Linux 64 bit or `-r RAM/2000` for 2GB of ram.

Altair Accelerator Administrator Tutorials

In the following tutorial, you will experiment with most issues that an Accelerator administrator will need to address, including starting/stopping the server, configuring FairShare, resources, taskers and environments.

This chapter covers the following:

- [Verify the License File](#) (p. 68)
- [Start/Stop the RLM License Server](#) (p. 70)
- [Start a Test Queue](#) (p. 71)
- [Start/Stop Accelerator](#) (p. 74)
- [Browser-based Setup](#) (p. 76)
- [Configure Policy - FairShare and Other Parameters](#) (p. 77)
- [Advanced Policy Configuration](#) (p. 79)
- [Configure Resources](#) (p. 81)
- [Configure Security](#) (p. 84)
- [Configure Taskers](#) (p. 86)
- [Configure an Environment](#) (p. 91)
- [Logical Names \(Equivalences\)](#) (p. 96)
- [Resource Management](#) (p. 99)
- [Upgrade Accelerator](#) (p. 103)

You will start a test queue to make the experiment non-disruptive to the default Accelerator queue.

Verify the License File

Altair supports two kinds of licensing:

- Altair Keyfile - read directly by vovserver, with hostid of the NHost
- Reprise License Manager (RLM) - provides floating licenses, with hostid of the license server host

A keyfile license contains a comment section and encrypted data representing the features. It should be placed in the `.swd` of the corresponding vovserver, or in the following location:

```
$VOVDIR/local/license.{nc,lm}.key
```

1. Verify your keyfile license information by using the command `vovreadlicutil`:

```
% vovreadlicutil -p nc -P 6271 -f $VOVDIR/local/license.nc.key
License file: $VOVDIR/local/license.key
Primary host: alpaca
-----
-- VOV License Key Information --
-----
License Number: 201104281356
VovVersion: 2020.01

Product: nc
Licensed to: Altair Engineering
For host id: MAC:00:d0:b7:75:b9:19
For host: panther
For port: 6339
Feature: ALL
License for: 6 slots
StartDate: Wed Apr 27 22:00:00 2019
Expiration: Sat Dec 31 21:00:00 2021
-----
(license information)
```

`vovreadlicutil` displays the license information of the license file specified by the `-f` option, with respect to the product and port number.

2. View the license file itself.

```
% more $VOVDIR/../../vnc/vnc.swd/license.key
#
# -----
# -- VOV License Key Information --
# -----
# License S/N: 201104281356
# VovVersion: 2020.1
# Product: nc
# Licensed to: Altair - Engineering
# For host id: MAC:00:14:2a:38:80:5b
# For host: croc
# For port: 0
# License for: 6 slots
# StartDate: Wed Apr 27 22:00:00 2019
# Expiration: Sat Dec 31 21:00:00 2021
# -----
#
# EFIFFFegjdEECEmdedceCGodDDLEGGlefeEEBCfeIDDE@EOGad
# ieieneC@feO@meo`LDHGFEGcKEaaadjaBEgaJEidngB@fdEDKB
# maceNE`adma`cldfaG@baIFB@`acaEFNAfcCDiehfbdEDCEdc
# ld`ahgm`ldj`@GB@neD@KFaajdL@IEJA`dg`NEndCCkdAEIDLE
```

```
FBhdd`fb@@EBG@ic``ifkdCDie```ebeBFneLDIGEABE``ddad  
meng@EAEJE`addneBDefifbdDBHDGA`edangI@ABJcF`EDLBBD  
g`ED@@D@ECAABBkaGCh`eemcicfdAEDDFEecdk@aea`neJAfd  
EAB@j`JAialeoaHAEEG@eekdECAEBEndM@FDdec`M@AFGDddf`  
JGgb@AI@IBLCJ@cdA@IEFAbfCGaaceNBefm`dengFEfaGEfa@D  
FBjeEFGEjfBEo`IEFFedddee`edfgeBDedbeKEkeBAgdKEid`a  
gaL@FabdLDAEBDfd@AIGNG`e
```

Start/Stop the RLM License Server

If you are using an RLM license file instead of a keyfile, you will need to start the license server on the host for which the license is made. The other Altair Accelerator programs locate the RLM license server by referring to the RLM_LICENSE environment variable.

Usually the RLM server directory will contain the platform-specific binaries for the license server host, including `rlm`, `rlmutil`, `rtdad`, and symbolic links from `rlmutil` to the usual command names like `rlmstat`, etc.

Disable the built-in web server and broadcast (starting in RLM 10)

```
% cd /opt/rlm
% ./rlm -nows -noudp -c rtda.lic -dlog reprise.log &
```

```
% printenv RLM_LICENSE
7070@buffalo
% rlmstat -avail
```

```
% cd /opt/rlm
% ./rlmutil rlmdown RLM
rlmdown v11.2
Copyright (C) 2006-2020, Reprise Software, Inc. All rights reserved.

shutting down RLM - are you sure? [y/n]: n
```

Start a Test Queue

At a given site, a single queue is recommended, such as a single Accelerator setup, by default called "vnc". Such a setup is called a *cluster* by other systems. The scheduler in Accelerator does not have queues in the sense used by other batch systems.

In this tutorial, you will start a temporary Accelerator queue for our testing. This allows you to experiment with most of the administration tasks without disturbing your production Accelerator queue.

Find the Server Working Directory

The Accelerator configuration directory for the default queue named "vnc" is in the product hierarchy. The server working directory for this queue is:

```
$VOVDIR/../../vnc/vnc.swd
```

This directory contains the server configuration files and server/tasker logs, etc.

Start a Queue

The command to start a queue is `ncmgr start`. By default, this command will start the default queue in the directory of `$VOVDIR/../../vnc`.

1. Get usage of this command:


```
vncmgr: Usage Message

USAGE:
    % ncmgr start [options]
OPTIONS:
    -h                This help.
    -force            Do not ask confirmation.
    -block           Do not return to the shell or command prompt
                    after starting. This is only useful, and
                    required, when starting Accelerator as a
                    Windows service.
    -port <port|mode> Specify port number, port number list (colon
                    separated) or port mode. Modes are:
                    automatic - hash queue name into port number,
                    do not start if port is
                    unavailable. The default queue
                    name 'vnc' hashes
                    to port 6271.
                    any - hash queue name into port number, try
                    additional ports in increments of 1
                    until an available one is found. The
                    default queue name 'vnc' hashes to
                    beginning port 11437.
                    Default: any
    -webport <port|mode> Specify a dedicated web interface port for
```

```
HTTP and HTTPS protocols. This port must be
configured to enable REST API v3 interface,
to enable the dashboard web UI page, and to
enable SSL. A value of 0 directs VovServer
not to open a web interface port. Port
number lists, 'automatic', and 'any' are not
supported.
Default: 0
-webkitprovider <provider> Specify the provider for
HTTP and HTTPS protocols.
This must be either "internal" or "nginx".
Default: "nginx"
-roport <port|mode> Specify read-only guest access web interface
port. A value of 0 disables this interface,
requiring all web interface users to log in.
Default: 0
-q, -queue <name> Name for queue (default is vnc).
-dir <dir> Directory of the server
(default $VOVDIR/../../vnc).
-dbhost <host> Host for database.
-dbroot <path> Path on database host for database files.
-dbport <port> Port of the database to listen for
connections.
-v Increase verbosity.
EXAMPLES:
% ncmgr start -port 6271
% ncmgr start -port 6271:6272:6273:any -force
% ncmgr start -q bigqueue -dir /remote/queues
```

2. Start the test queue in the home directory.
3. Name your queue `vncdextrin` (make sure you pick a name that does not conflict with any existing queue). The first three letters of the queue name should be formed by prefixing your login with `vnc`
For example, if your login is 'danny', use the name 'vncdanny' for your queue.

```
+ start a shell on the machine where your NC vovserver should run
% cd # Go to your home directory
% mkdir ncdadmin # Create a directory for our testing queue
% ncmgr start -dir ncdadmin -queue vncdextrin
message: Checking the license...
message: ... the license is good.
message: Starting NC
message: with name vncdextrin
message: on host alpaca
message: in directory /home/dextrin/ncdadmin
message: as user dextrin
Do you want to proceed? (yes/[no]) > yes
message: Updating config file '/remote/release/VOV/2013.09/linux64/local/
vncConfig/vncdextrin.tcl' message: Waiting for server to be ready ...
message: Sanity check...
message: NC vncdextrin@alpaca is ready.
```

 **Note:** Make sure the directory `$VOVDIR/local/vncConfig` is writable to you, because a config file needs to be written in that directory to start a new queue. The Altair Accelerator installer should have set this.

Use a Specific Queue

Now that you have started a new queue, you have at least two Accelerator queues in your system. There are two ways to use a particular queue.

1. Use -q or -queue option:

```
# Submit a job to queue "vncdextrin"
% nc -q vncdextrin run sleep 60

# List my jobs in queue "vncdextrin"
% nc -q vncdextrin list

# Get info about queue "vncdextrin"
% ncmgr info -queue vncdextrin

# Reset all taskers for queue "vncdextrin"
% ncmgr reset -queue vncdextrin -taskers
```

2. Alternatively, you can set the environment variable NC_QUEUE:

You can also set an environment variable NC_QUEUE to the name of queue you want. Then you can execute all your `nc` and `ncmgr` commands in the context of that queue without having to use `-q` or `-queue` options.

```
% setenv NC_QUEUE vncdextrin

# Now submit a job to queue "vncdextrin"
% nc run sleep 61


# etc.
```

This way, you can easily switch between all the queues you have.

Start/Stop Accelerator

The commands involved in this tutorial are `ncmgr info`, `ncmgr start` and `ncmgr stop`.

You should use `ncmgr`, especially when starting Accelerator, because it checks for and applies any daemon and DB schema changes when moving to a new version.

 **Note:** Please remember to use the shell where you set the `NC_QUEUE` environment variable to the name of your training tutorial queue.

View Accelerator Status

Use command `ncmgr info` to get a summary information of all taskers and current running jobs.

```
% ncmgr info
ncmgr: message: NC vncdexin@alpaca
1 001 alpaca 1/0 146198 Unlim. IDLE
2 002 cheetah 1/0 235294 Unlim. IDLE
3 004 bison 1/1 99206 Unlim FULL
   2s:      vw vwrap sleep 60 > vnc long
4 000 pluto 1/0 75312 Unlim. IDLE
5 003 rhino 1/0 181818 Unlim. IDLE
```

Start Accelerator

1. Execute `ncmgr` to see if Accelerator is already running.

```
% ncmgr start
ncmgr: USER ERROR: NC vncdexin@alpaca already running.
```

2. If not already running, start Accelerator. In production use, it is important to check the number of file descriptors available to `vovserver`.

```
% ncmgr start
ncmgr: message: Checking the license...
ncmgr: message: ... the license is good.
ncmgr: message: Starting NC
ncmgr: message: with name          vncdexin
ncmgr: message: on host            alpaca
ncmgr: message: in directory       /home/dexin/ftadmin
ncmgr: message: as user           dextrin
ncmgr: message: with             1024 file descriptors
Do you want to proceed? (yes/[no]) > yes
ncmgr: message: Waiting for server to be ready ...
ncmgr: message: Sanity check...
ncmgr: message: NC vncdexin@alpaca is ready.
```

Stop Accelerator

Run the following:

```
% ncmgr stop
ncmgr: message: Checking if NC vncdextrin@alpaca is running...
ncmgr: message:
You are about to stop NC vncdextrin@alpaca in directory /home/dextrin/ftadmin
    Would you like to proceed (yes/[no]) ? > yes
ncmgr: message: Stopping taskers and server ...
ncmgr: message: NC vncdextrin@alpaca has been stopped.
```

Restart Your Accelerator Queue

You should restart your queue so that it will be running for later exercises.

Run the following:

```
% ncmgr start
ncmgr: message: Checking the license...
ncmgr: message: Checking the license...
ncmgr: message: ... the license is good.
ncmgr: message: Starting NC
ncmgr: message:   with name      vncdextrin
ncmgr: message:   on host        alpaca
ncmgr: message:   in directory   /home/dextrin/ftadmin
ncmgr: message:   as user        dextrin
    Do you want to proceed? (yes/[no]) > yes
ncmgr: message: Waiting for server to be ready ...
ncmgr: message: Sanity check...
ncmgr: message: NC vncdextrin@alpaca is ready.
```

Browser-based Setup

Accelerator provides a simplified user-friendly browser-based setup, which is especially useful for new users. To use this browser-base setup page, please first make sure Accelerator is running.

Find the URL

First find the Accelerator URL, using `vovbrowser` or `vsi`:

```
% nc cmd vovbrowser
http://yourhost:6295/project
% nc cmd vsi
(more detailed output, including the Accelerator vovserver URL)
```

The setup script is available at the URL `/cgi/setup.cgi` (for example, in this case, it is `http://alpaca:6295/cgi/setup.cgi`).

Setup Using the Web Page

1. Follow the instructions on the web page to finish the basic setup.
2. On the Taskers setup page, try to add at least one tasker for each type (Server, Workstation, Offhours).
3. View the tasker statuses at URL `/taskers`.
4. Try out some test jobs.



Note: You need a working remote-shell setup to start non-local vovtaskers for this step.

Configure Policy - FairShare and Other Parameters

Accelerator has a time-windowing multi-level FairShare mechanism. This allocates CPU cycles to the fsgroups at each level according to the assigned weights of each group until the leaf nodes of the FairShare tree are reached. Each node of the FairShare tree may have a separate time window. The word 'fsgroup' is used as an abbreviation here.

To configure FairShare, use the `vovfsgroup` command.

Access to FairShare groups is controlled by ACLs (Access Control Lists), which means you can configure them so only designated users can submit jobs in an fsgroup.

Locate Server Configuration Directory to Find `policy.tcl`

The test queue was start in `vncdextrin` in directory `~/ncadmin`. So the server configuration directory for this Accelerator queue is `~/ncadmin/vncdextrin.swd`. By default, the server configuration directory is `$VOVDIR/../../vnc/vnc.swd`.

If you forget, you can find it out by this command:

```
% nc cmd vovserverdir  
/home/dextrin/ncadmin
```

In this case, you will find `policy.tcl` file at `/home/dextrin/ncadmin/vncdextrin.swd/policy.tcl`.

You will find other configuration files in the same directory, for example, `taskers.tcl`, `resources.tcl`.

Configure `policy.tcl` for FairShare

In the following example, two groups, `production` and `regression`, are configured.

Execute the following:

```
# in policy.tcl  
% vovfsgroup create /production -weight 400 -window 8h  
% vovfsgroup create /regression -weight 100 -window 8h
```

Two groups are defined: `/production` and `/regression`. When there are jobs from both the `production` group and `regression` group, the target share ratio of CPUs will be 400:100.

Save the New Configuration

Changes made by the `vovfsgroup` command are only in the `vovserver`'s memory until the next save, and may be lost if the `vovserver` is restarted before then.

Save the configuration to a file so it can be reloaded.

```
% nc cmd vovfsgroup genconfig myfsconfig.tcl
```

Test the New Fairshare Configuration

1. You use `-g` option in command `nc run` to submit a job from a particular group.

```
% nc run -g regression -f listOfJobs  
% nc run -g production -f listOfJobs
```

2. Use the Monitors GUI or browser page to monitor the dynamic changes of FairShare.
 - Use the monitors: use command `nc monitor` to bring up the monitors and click on the **FairShare** tab.
 - Use the browser: use command `nc cmd vovbrowser` to find the Accelerator URL, then find the FairShare page at URL `/cgi/fairshare.cgi`.

Server Configuration Parameters

Certain parameters of the Accelerator `vovserver` are also configurable by means of entries in the `policy.tcl` file. The [Server Configuration](#) page describes these in detail.)

Here are some examples:

```
# This is part of the policy.tcl file.  
set config(maxQueueLength)      8000  
set config(httpSecure)          1  
set config(saveToDiskPeriod)    2h;  
set config(autoLogout)          1h;    # Logout from browser interface  
  
# Used by Accelerator for autoforget.  
set config(autoForgetValid)     1h  
set config(autoForgetFailed)    2d  
set config(autoForgetOthers)    2d  
set config(autoRescheduleThreshold) 2s
```

Advanced Policy Configuration

Besides the vovserver configuration file `policy.tcl`, the behavior of job submission to Accelerator can also be controlled by the file `$VOVDIR/local/vnc_policy.tcl`, which is used to define the following procedures:

<code>VncPolicyDefaultResources</code>	The default resources required by a job.
<code>VncPolicyValidateResources</code>	Make sure that the resource list for a job obeys any number of rules.
<code>VncPolicyDefaultPriority { user }</code>	Assign the default priority to a job based on the user.
<code>VncPolicyMaxPriority { user priority }</code>	Limit the priority based on the maximum allowed to the user.

In this tutorial, you will configure `VncPolicyDefaultResources` and `VncPolicyValidateResources`.

Configure Default Resources

By default, Accelerator takes machine architecture as the resource of the job that you submit from a particular machine. This is controlled by the default setting of `VncPolicyDefaultResources`:

```
proc VncPolicyDefaultResources {} {  
    global env  
    return "$env(VOVARCH)"  
}
```

To change this behavior, you can create or edit the file `$VOVDIR/local/vnc_policy.tcl`, and add or edit the procedure that follows. This procedure will set the default resources to be the architecture and 50mb of memory. If you have more than one Accelerator setup, you can place the `vnc_policy.tcl` file in the `.swd` and it will apply only to that one.

```
proc VncPolicyDefaultResources {} {  
    global env  
    return "$env(VOVARCH) RAM/50"  
}
```

Enforce Job Resource Rules

You can also enforce some rules of job resources by overriding the procedure `VncPolicyValidateResources`. Here is the default behavior:

```
proc VncPolicyValidateResources { resList } {  
    return $resList  
}
```

But this process does nothing. Try the process described below:

To enforce a rule so that all jobs require a minimum RAM of 512 MB, create or edit the file `$VOVDIR/local/vnc_policy.tcl` and add or edit `VncPolicyValidateResources` procedure as follows:

```
proc VncPolicyValidateResources { resList } {  
    #  
    # This policy adds a minimum RAM requirement  
    # for all submitted jobs.  
    #  
  
    if [regexp "RAM/" $resList] {  
        # Already a RAM constaints.  
    } else {  
        lappend resList "RAM/512"  
    }  
    return $resList  
}
```


Configure Resources

Accelerator includes a sophisticated subsystem for the management of computing resources, which allows the design team to take into account all sorts of constraints regarding hardware and software resources, as well as site policy constraints. This mechanism is based on:

- The resources required by jobs
- The resources offered by taskers
- The ResourceMap, as described in the file `resources.tcl`

Next, you will configure the resource map in `resources.tcl`.

Find and View the resources.tcl File

You can find this file in the server configuration directory. For default `vnc queue`, it is `$VOVDIR/../../../../vnc/vnc.swd`. For the test queue, it is in directory `~/ncadmin/vncdixin.swd/`.

```
% cd ~/ncadmin/vncdixin.swd
% vi resources.tcl ; # Use the editor of your choice
```

```
# ... here we only show part of this file ...
vtk_resourcemap_set PRIORITY_LOW      1
vtk_resourcemap_set PRIORITY_NORMAL  10
vtk_resourcemap_set PRIORITY_HIGH    20
vtk_resourcemap_set PRIORITY_TOP     UNLIMITED
```

With above default configuration, there will be at most 1 low priority job running at any time, 10 for normal, 20 for high, and any number of top priority jobs could be running.

1. Run a simple test to verify that the above information is correct:

```
% nc run -f $VOVDIR/training/vnc/cmdlist.unix
```

2. You can use the Monitors window to monitor the "Running Jobs" and "Resources" to see the running jobs and resources usage. You can also use browser to get similar information from "Running Jobs" page and "Resources" page.

Configuration Examples

Example 1

For example, if you decide that no more than 4 normal priority jobs should be running at any time, you can edit `resources.tcl` and modify the value for normal priority to 4.

```
# ... here we only show part of this file ...
vtk_resourcemap_set PRIORITY_LOW      1
# Now we change this value to 4
```

```
vtk_resourcemap_set PRIORITY_NORMAL 4  
vtk_resourcemap_set PRIORITY_HIGH 20  
vtk_resourcemap_set PRIORITY_TOP UNLIMITED
```

Save your change and do a reread and test the new configuration:

```
% nc cmd vovproject reread  
% nc run -f $VOVDIR/training/vnc/cmdlist.unix
```

Example 2

You can configure your resources similarly. For example, you have 10 calibre license, you can configure this by adding the following line in `resources.tcl`:

```
# In resources.tcl  
vtk_resourcemap_set calibre_license 10
```

You can also associate a resource to other resource(s). For example, you have 4 licenses of spice that are only available on Linux. You can configure this by add the following line in `resources.tcl`:

```
# In resources.tcl  
vtk_resourcemap_set hspice_license 4 linux
```

Resource Monitor

To monitor resource activity, call a GUI monitor with the command:

```
nc mon
```

The screenshot shows a window titled "nc VOV Monitors 2021.1.1-dev-91405 vnc". The interface includes a menu bar with "File", "Options", "TaskersGroups", "Taskers", and "Resources". Below the menu is a tabbed interface with tabs for "TaskerGroups", "Taskers", "Tasker HW", "Tasker Resources", "Who", "Running Jobs", "Running Commands", "Running Details", "Resources", "Queued Jobs", "Queue Buckets", and "FairShare". The "Resources" tab is active, displaying a table with the following data:

	Type:Name	Total	InUse	Rsrvd	ooQ	Available	%Util.	MapsTo
1	License:AL002_HWPoll	unlim	0	0	0	unlim		
2	Limit:user0265_justo	1	0	0	0	1	99.12%	
3	License:AL002_HFSoli	100000	0	0	0	100000	0.00%	
4	License:AL003_HWPSO_	unlim	0	0	0	unlim		
5	Limit:user0418_justo	1	0	0	0	1	99.79%	
6	Limit:user0354_justo	1	0	0	0	1	97.64%	
7	Limit:user0100_justo	1	0	0	0	1	98.94%	
8	License:AL003_HWPSO_	unlim	0	0	0	unlim		
9	License:SolverNode	100000	0	0	0	100000	0.00%	
10	Limit:user1312_justo	1	0	0	0	1	98.16%	
11	License:AL003_HWPano	unlim	0	0	0	unlim		
12	License:AL003_SIMLAB	100000	0	0	0	100000	0.00%	
13	License:AL001_HWActi	unlim	0	0	0	unlim		

Figure 3:

Configure Security

Accelerator has 4 privilege levels: READONLY, USER, LEADER, ADMIN.

For detailed information about security, please refer to [Security](#).

Locate the Security Configuration File: security.tcl

This file is in the server configuration directory, default `$VOVDIR/../../vnc/vnc.swd/security.tcl` and in our test setup, that is `~/ncadmin/vncdexin.swd/security.tcl`.

Security Configuration Examples

Least Restrictive Security

The least restrictive security grants everybody full access from any host. This should not be used in production.

```
# All users (+) are administrators from all hosts (+).  
vtk_security + ADMIN +
```

Alternatively, a VovUserGroup may be utilized, to assign individuals in a group the ADMIN privilege.

```
# Members of mygroup are administrators from all hosts (+).  
vtk_security -group mygroup ADMIN +
```

Most Restrictive Security

```
# No rule defined gives only the owner of the project ADMIN privileges  
# on the server host.
```

Typical Case

The following example shows a typical security file, in which different privileges are granted to different users. Also notice the use of variables and VovUserGroups in this example.

In the example, `mary` is an administrator for any host, and `dan` is an administrator only for `reno` and `milano`. The user `pat` is a LEADER for her machine `elko`, and `fred` has USER privileges for 4 machines listed in the variable `$allhosts`. Members of the VovUserGroup "operators" have ADMIN rights on `$allHosts`.

```
set servers          { reno milano  }  
set allhostsset     { reno milano elko tahoe}  
  
vtk_security mary    ADMIN    +  
vtk_security john    ADMIN    tahoe  
vtk_security dan     ADMIN    $servers
```

```
vtk_security pat          LEADER  elko
vtk_security fred        USER    $allhosts
vtk_security  -group operators ADMIN  $allHosts
```

Configure Taskers

The file `taskers.tcl` describes the vovtaskers for Accelerator.

vovtasker Configuration

You can find the file `taskers.tcl` in server configuration directory, default at `$VOVDIR/../../vnc/vnc.swd/taskers.tcl`, and in our test case at `~/ncadmin/vncdixin.swd/taskers.tcl`.

This file is based on two commands (Tcl procedures), `vtk_tasker_define` and `vtk_tasker_set_default`:

```
# Set default behavior of vovtaskers
vtk_tasker_set_default [options]

# Define a vovtasker
vtk_tasker_define hostname [options]
```

If you set some options with `vtk_tasker_set_default` command, all the following `vtk_tasker_define` commands will use those options implicitly, and options set explicitly in `vtk_tasker_define` overwrite those set in `vtk_tasker_set_default`.

For example, to declare 3 vovtaskers on the hosts apple, orange, and pear, use:

```
# In taskers.tcl
vtk_tasker_define apple
vtk_tasker_define orange
vtk_tasker_define pear
```

or some equivalent code:

```
# In taskers.tcl
foreach host {apple orange pear} {
    vtk_tasker_define $host
}
```

To understand the use of `vtk_tasker_set_default`, define 3 vovtaskers as follows:

```
vtk_tasker_define apple -resources "@STD@ big_memory" -CPUS 2
vtk_tasker_define orange -resources "@STD@ big_memory" -CPUS 2
vtk_tasker_define pear -resources "@STD@ big_memory" -CPUS 4
```

`vtk_tasker_set_default` does the following:


```
vtk_tasker_set_defaults -resources "@STD@ big_memory" -CPUS 2
vtk_tasker_define apple
vtk_tasker_define orange
vtk_tasker_define pear -CPUS 4
```

Tasker Configuration Examples

The default `taskers.tcl` provides pretty good examples of configuring vovtaskers. Most of the time, you just need to plug in some host names in brackets to define vovtaskers. For example, here is one piece of code in `taskers.tcl`.

```
# ADD THE NAMES OF THE COMPUTE SERVERS TO THE FOLLOWING LIST
set mainComputeServers {}

foreach host $mainComputeServers {
    vtk_tasker_define $host -resources "VovResources::Standard @RAMTOTAL@ @SWAPFREE@"
}
```

 **Note:** `-resources "VovResources::Standard"` is equivalent to `-resources"@STD@"`

To add some "Server" class vovtaskers, you just need to add the names of those hosts into the list `mainComputerServers`, like the following:

```
set mainComputeServers { apple orange pear }
```

You can certainly add or modify the vovtasker definition as you want (subject to license restriction). For example, you have many dual CPU machines, and you would like to make the *maxload* of these vovtasker machines bigger, say, equal to 1.5 times of the CPUs, for example 3.0. Then you can do this:

```
set myDualCpuServers {}

foreach host $myDualCpuServers {
    vtk_tasker_define $host -maxload 3.0 -resources "VovResources::Standard
@RAMTOTAL@ @SWAPFREE@"
}
```

Add Workstation/Offhours vovtaskers

```
# Add a Workstation vovtasker that will only start to accept
# job after 10 minutes of idle and will only accept jobs
# with expected duration no longer than 5 minutes
vtk_tasker_define ftcsun44 -resources "VovResources::Workstation -minIdle 10m -
maxtime 5m @RAMTOTAL@"

# Add a Offhours vovtasker that's only available from 7pm to 6am
# every weekday and on weekends
vtk_tasker_define ftcsun66 -resources "VovResources::Offhours"
```

Start Newly Defined vovtaskers

Like other configurations, Accelerator will pick up the changes in `taskers.tcl` when the server is stopped and restarted. This is often too disruptive and not favorable, especially when there are jobs running in Accelerator. Here are some other ways to apply your changes:

1. To start the vovtaskers that you just defined, use command:

```
% ncmgr reset -taskers
```

2. To start vovtaskers that you just modified, you must stop and start them.

Start/Stop vovtaskers (Advanced)

You can start/stop vovtaskers from the GUI and the browser.

The following is an advanced command of FlowTracer `vovtaskermgr` and how to run any command in the context of Accelerator using `nc cmd`.

```
# List all vovtaskers defined in taskers.tcl
# Also checks the taskers.tcl file for syntax errors
% nc cmd vovtaskermgr list

# Start all vovtaskers defined in taskers.tcl
% nc cmd vovtaskermgr start

# Start some vovtasker(s)
% nc cmd vovtaskermgr start tasker1 tasker2

# Stop all vovtaskers
% nc cmd vovtaskermgr stop

# Stop some vovtasker(s)
% nc cmd vovtaskermgr stop tasker1 tasker2

# Show detail information of all vovtaskers
% nc cmd vovtaskermgr show

# Get usage of all vovtaskermgr commands
% nc cmd vovtaskermgr
```

Start a vovtasker from Command Line (Advanced)

You can also start a vovtasker on the fly from the command line using the `vovtasker` binary. This can sometimes be handy, for example, for debugging. This is the command that Accelerator uses to start the vovtaskers after it reads the `taskers.tcl` configuration file.

1. To start a vovtasker on a particular host, you need to go to that host and use command `nc cmd vovtasker` with appropriate options, which are similar to the options of `vtk_tasker_define`.
2. Try the following examples and monitor the vovtaskers using the Monitor GUI or browser Tasker page.
 - Get usage of this command:

```
% nc cmd vovtasker
usage: vovtasker [-A startupLogFile] [-a name] [-b capabilities] [-B]
               [-c coefficient] [-C cpus] [-d] [-D integer] [-e reserveExpr]
               [-E] [-f tclfile] [-F <file>]
```



```
[ -g taskergroup ] [ -G group ] [ -h host ] [ -H HEALTHCHECKFLAGS ]
[ -I 0|1 ] [ -I tcclfile ] [ -j ] [ -k d|n|v ] [ -l rootofDailyLogFile ]
[ -L <loadSensor> ] [ -m <integer> ] [ -M max_load ] [ -n <integer> ]
[ -N ] [ -o local resource ] [ -p project ]
[ -P <double> ] [ -r resources ] [ -R resources ] [ -s ] [ -S
resources ] [ -t timeout ] [ -T capacity/max_capacity ]
[ -U updateInterval ] [ -v number ] [ -V ncName@ncHost[:port] ]
[ -w WX properties ] [ -z <timeSpec> ] [ -Z <timeSpec> ] [ -g name ]
[ -u name ]

-A: The name of the startup log file
-a: Name this tasker. The name may contain only letters, numbers,
dash(-) and underscore(_), or the expressions @HOST@ and @PID@ that get
expanded on the fly
-b: Comma-separated list of capabilities, case insensitive:
    symbolic: FULL NC LM
    normal: PROCINFO NC LM
    short: P N X R
-B: Show BPS tasker objects. Default to not show.
-c: Tasker coefficient (positive, default 1.0)
-C: Number of CPU's in this machine (automatic on win64). Use 0
to specify default value

...OMITTED...
```

- Start a normal vovtasker (with all default settings):

```
% nc cmd vovtasker -N
vovtasker Jan 10 13:44:42
Copyright © 1995-2020, Altair Engineering Linux/7.1 Jan 10 2019 10:00:59
vnc@alpaca
vovtasker Jan 10 13:44:42 Test 1: INTEGER OPS W= 1.00 Reps= 500 T= 5.00ms
vovtasker Jan 10 13:44:42 Test 1: DOUBLE OPS W= 1.00 Reps= 25 T= 10.00ms
vovtasker Jan 10 13:44:42 Test 1: CHAR OPS W= 0.10 Reps= 10 T= 10.00ms
vovtasker Jan 10 13:44:42 ---- Weighted time: 16.00ms
vovtasker Jan 10 13:44:42 Test 2: INTEGER OPS W= 1.00 Reps= 500 T= 5.00ms
vovtasker Jan 10 13:44:42 Test 2: DOUBLE OPS W= 1.00 Reps= 25 T= 10.00ms
vovtasker Jan 10 13:44:42 Test 2: CHAR OPS W= 0.10 Reps= 10 T= 20.00ms
vovtasker Jan 10 13:44:42 ---- Weighted time: 17.00ms
vovtasker Jan 10 13:44:42 Test 3: INTEGER OPS W= 1.00 Reps= 500 T= 5.00ms
vovtasker Jan 10 13:44:42 Test 3: DOUBLE OPS W= 1.00 Reps= 25 T= 10.00ms
vovtasker Jan 10 13:44:42 Test 3: CHAR OPS W= 0.10 Reps= 10 T= 20.00ms
vovtasker Jan 10 13:44:42 ---- Weighted time: 17.00ms
vovtasker Jan 10 13:44:42 Best weighted time: 16.00ms
```

- Start a vovtasker with name "myvovtasker", max load 4.0, and offers standard resources "@STD@" and resource "special_license":

```
% nc cmd vovtasker -a myvovtasker -M 4.0 -r "@STD@ special_license"
vovtasker Jan 10 13:51:00
Copyright © 1995-2020, Altair Engineering Linux/7.1 Jan 10 2019 10:00:59
vnc@alpaca
vovtasker Jan 10 13:51:00 Test 1: INTEGER OPS W= 1.00 Reps= 500 T= 5.00ms
vovtasker Jan 10 13:51:00 Test 1: DOUBLE OPS W= 1.00 Reps= 25 T= 5.00ms
vovtasker Jan 10 13:51:00 Test 1: CHAR OPS W= 0.10 Reps= 10 T=
20.00ms
vovtasker Jan 10 13:51:00 ---- Weighted time: 12.00ms
vovtasker Jan 10 13:51:00 Test 2: INTEGER OPS W= 1.00 Reps= 500 T= 5.00ms
vovtasker Jan 10 13:51:00 Test 2: DOUBLE OPS W= 1.00 Reps= 25 T= 5.00ms
vovtasker Jan 10 13:51:00 Test 2: CHAR OPS W= 0.10 Reps= 10 T=
20.00ms
vovtasker Jan 10 13:51:00 ---- Weighted time: 12.00ms
```

```
vovtasker Jan 10 13:51:00 Test 3: INTEGER OPS W= 1.00 Reps= 500 T= 5.00ms
vovtasker Jan 10 13:51:00 Test 3: DOUBLE OPS W= 1.00 Reps= 25 T= 5.00ms
vovtasker Jan 10 13:51:00 Test 3: CHAR OPS W= 0.10 Reps= 10 T=
20.00ms
vovtasker Jan 10 13:51:00 ---- Weighted time: 12.00ms
```

Configure an Environment

VOV includes a sophisticated subsystem to manage environments. In Accelerator, an environment is a named collection of environment variables and their values. Many tools expect certain environment variables to be set for correct operation, and the variable `PATH` is used by most UNIX shells to locate executable programs.

A VOV environment definition consists of three scripts defined in either Tcl, C-shell, or Bourne-shell syntax. Environments that will be used on both UNIX and Windows must be written in Tcl. The names of these scripts must follow a naming convention. For example, an environment named `ENV` would use these three scripts:

- `ENV.start.tcl`
- `ENV.end.tcl`
- `ENV.doc`

The 'start' script is used when entering the environment, the 'end' script is used when leaving the environment, and the 'doc' script contains a one-line summary of the environment's purpose.

VOV provides commands for using environments from the command line:

<code>vel</code>	List available environments
<code>vesENV</code>	Switch to environment ENV
<code>vep</code>	Change prompt to show environment name
<code>veprestore</code>	Return to original prompt

Find and View Example Environment Scripts

It is interesting and instructive to examine some of the environment definitions which are shipped with VOV. One of the most important is the `BASE` environment.

The most common place for environment scripts is `$VOVDIR/local/environments`, but you can also specify additional directories where VOV will search. We will discuss this in more detail later in the lab.

The `BASE` definition is in the 'local' directory of the VOV software installation, at `$VOVDIR/local/environments/BASE.start.tcl`

```
% pushd $VOVDIR/local/environments
% view BASE.start.tcl

# --
# -- Completely reset the environment to bare bones.
# -- We assume that VOVDIR and VOVARCH are set.
# --

set VOVDIR $env(VOVDIR)
```

```
if { $::tcl_platform(platform) eq "windows" } {
#
# --
#
nt_preprocess_env
# set vovenv(debug) 1 set WINDIR $env(WINDIR)
setenv PATH "$WINDIR\\system32;$WINDIR"

set MKSDIR "c:/mksdemo"
if [file exists $MKSDIR] {
setenv ROOTDIR $MKSDIR
vovenv PATH ";" APPEND $MKSDIR/mksnt
}
#
# Depending on the installation, the binaries can be in
# different directories.
#
foreach b { bin bin-O bin-g bat local/bat } {
set p "$VOVDIR/$b"
#
# Eliminate double // as in w://bat
#
regsub -nocase {^([a-z])://} $p {\1:/} p

if [file exists $p] {
vovenv PATH ";" APPEND $p
}
}

set version [exec vovversion]
vovenv PATH ";" PREPEND c:/temp/vov/execache$version

# vovenv PATH ";" PREPEND c:/temp/vov/execache
} else {
#
# -- Unix BASE environment.
#

setenv PATH ""
foreach b { bin bin-O bin-g } {
if [file exists $VOVDIR/$b] {
vovenv PATH : APPEND $VOVDIR/$b
break
}
}
vovenv PATH : APPEND $VOVDIR/scripts
vovenv PATH : APPEND $VOVDIR/local/scripts

setenv MANPATH ""
foreach mp [list /usr/man /usr/share/man /usr/local/man /usr/openwin/man $VOVDIR/man]
{
if [file isdirectory $mp] {
vovenv MANPATH : APPEND $mp
}
}

setenv LD_LIBRARY_PATH ""
foreach lp [list /lib /usr/lib /usr/local/lib /usr/openwin/lib $VOVDIR/lib] {
if [file isdirectory $lp] {
vovenv LD_LIBRARY_PATH : APPEND $lp
}
}
}
```

```
}  
  
foreach xapp { /usr/openwin/lib/app-defaults /usr/lib/X11/app-defaults } {  
  if [file isdirectory $xapp ] {  
    vovenv XFILESEARHPATH : APPEND "$xapp/%N"  
  }  
}  
  
# -- /opt/aCC/bin is for HPUX  
foreach p {  
  /bin /usr/ucb /usr/bin /usr/local/bin  
  /usr/X11/bin  
  /usr/dt/bin  
  /usr/openwin/bin /usr/bin/X11 /opt/aCC/bin  
} {  
  if [file isdirectory $p] {  
    vovenv PATH : APPEND $p  
  }  
}  
}
```

The BASE environment includes just the regular system commands and VOV. This is an example of a complex environment setup script; most of the ones you will need to define will be much simpler.

A simple test to verify that:

```
% vep
```


Notice that the prompt has changed to show that you are in the DEFAULT environment. This is the environment which is defined by your regular system setup files, for example, `.cshrc`.

```
% ves BASE  
% printenv PATH
```

Notice that the PATH environment variable is now (probably) much shorter, and contains the VOV software directories.

```
% veprestore
```

Notice that the prompt has been changed back to the original.


 **Note:** This only changes the prompt. Your shell still has whatever environment was set most recently using `vep`.

Add Additional Environment Directories

When you are testing and developing environments, or when you want to have project-specific ones which are not put in the system wide directory at `$VOVDIR/local/environments`, you can use the environment variable `VOV_ENV_DIR` to specify additional directories.

You will use this capability in the tutorial, so that you don't need to put our environment scripts in the system's shared environment area. You should have a `~/ncadmin` directory from the previous exercises, which contains the server working directory of your private Accelerator queue.

```
% setenv VOV_ENV_DIR ~/ncadmin/vnc-user-name.swd/environments
% pushd $VOV_ENV_DIR
```

 **CAUTION:** The `environments` directory is generated by setup, but any user-made env-scripts placed in the directory will not be used unless `VOV_ENV_DIR` includes it.

Configuration Examples

In this example, you will create an environment which adds your personal bin directory to the path. You will put the definitions in your queue's environment directory.

Use Tcl for the environment scripts because it is more powerful than C-shell, and platform-independent.

Example 'start' script for JOHN environment

```
# add john bin directory to path
vovenv PATH : PREPEND /home/john/bin
```

This script adds the element `/home/john/bin` to the beginning of the PATH. There is also an APPEND operator which adds to the end of the PATH. If the exact element is already present in the path, it is left undisturbed. This avoids PATH overflow in C-shell.

Example 'end' script for JOHN environment

```
# remove john bin directory from path
vovenv PATH : DELETE /home/john/bin
```

This script removes the element `/home/john/bin` from the PATH.

Example 'doc' script for JOHN environment

```
prepend /home/john/bin to PATH
```

This file contains a one-line summary which is presented by the `vel` command.

Use the above examples to create similar files in your queue.

Compose Good Environment Scripts

There are several characteristics of a 'good' environment setup.

minimal	A minimal environment only sets the variables needed to accomplish a specific purpose.
composable	A composable environment respects pre-existing values of variables, and adds to them, rather than overwriting them. For example, when modifying <code>LM_LICENSE_FILE</code> , use <code>PREPEND</code> and <code>APPEND</code> rather than simply setting it. This allows you to use commands like <code>ves BASE+CADENCE+NASSDA</code> and have both Cadence and Nassda tools work.
reversible	A good environment definition uses the 'end' script to revert what it set in the 'start' script, so that the environment does not gradually become polluted with obsolete values.

Environments are Cached on the vovtasker

To provide very low job dispatch latency, Accelerator caches the eight most-recently used environments in the vovtasker process. This means that jobs which run in any of those environments may start immediately without sourcing the environment definition script. It also means that if you change an env-script, you must tell the vovtaskers by using the `vovtaskermgr refresh` subcommand.

Summary

- Most environments are defined by scripts stored in `$VOVDIR/local/environments`
- Environment scripts are in Tcl or C-shell syntax
- You can use the `VOV_ENV_DIR` environment variable to specify additional directories to search for environment scripts.
- Environments are cached on the vovtasker, and refreshed using `ncmgr reset -taskers`, or `nc cmd vovtaskermgr refresh`


Logical Names (Equivalences)

Most sites set up their machines so that they all have uniform mountpoints and view of the file systems. A common scheme is to place all user home directories under a single parent directory, often called `/home`. The actual file system is mounted using automount, and the physical file system is located by an NIS map.

Unlike some other batch/network computing systems, Accelerator does not require that all the compute machines have a uniform set of mountpoints. Accelerator can use what are called logical names or equivalences, to locate the data.

The default `equiv.tcl` file which is shipped with the software defines a logical name for the path to `VOVDIR`.

The `equiv.tcl` file in some versions also defines a logical name `HOMES` for whatever the parent of your home directory as set in the environment variable `HOME`. The default one shipped with an early version did not, so we will show the steps to define a logical name using the browser-based setup.

 **Note:** When doing the exercises, remember to give these commands from a shell where you have set the environment variable `NC_QUEUE` to the name of your training tutorial queue. Otherwise, you will need to use the `-queue` option.

Define a Logical HOME Directory Name

1. Use the browser-based setup to view the logical names that are defined. First, find the URL of the queue's server, and connect a web browser to it.

```
% nc cmd vovbrowser --> http://host:port/project
```

```
% firefox http://host:port/cgi/setup.cgi
```

The initial page will have a link on the left called FileSystems.

2. Left-click on this link to display the logical names defined for the queue. You will see `VOVDIR` and `VNCSWD`.
3. Run a job in your home directory.

```
% cd  
% nc run sleep 10
```

This will often run, because the default is to require the architecture from which the job was submitted as a resource, so the job may run on your machine.

4. Now try the job, but request an architecture different from the machine on which you are working. This job will fail, because the directory on the remote machine does not have a logical name.

```
% nc run -r linux64 -e BASE pwd
```

```
Resources= linux64 Env = BASE  
Command = vw vwrap pwd  
Logfile = vnc_logs/20030108/143125.27906 JobId = 00000021  
vnc: message: Scheduled jobs: 1 Total estimated time: 0s
```

```
% nc list  
00000011 Done sleep 10  
00000021 Failed pwd
```

```
% nc info -l 21  
  
Log file is: '${VNCSWD}/vncjohn.swd/vnc_logs/20030108/143125.27906'  
vnc: message: File ${VNCSWD}/vncjohn.swd/vnc_logs/20030108/143125.27906 does not  
exist (yet)  
    (mapped to /usr2/home/john/ncadmin/vncjohn.swd/  
vnc_logs/20030108/143125.27906)
```

5. Run the job in the /tmp directory.

/tmp exists on all UNIX machines, but is not usually exported to the network.

```
% cd /tmp  
% nc run sleep 10
```

This fails, because /tmp is not a network path.

```
% nc run sleep 10  
  
WARNING:  
  The path to the current directory is not 'LOGICAL',  
  i.e. it does not begin with a '$' sign.  
EXPLANATION:  
  The current directory  
  /tmp  
  may not be a valid directory path for all hosts in the network.  
OPTIONS:  
  (1) Continue.  
      By choosing this option, you assert that the path  
      is valid everywhere. If it is not, the job is likely to fail,  
      because the remote host cannot reach the current directory.  
      This option causes the creation of a flag file called .vnc which has  
the  
      purpose of avoiding the repetition of this question for this directory  
and  
      its subdirectories  
  (2) Abort.  
      Please ask your NetworkComputer administrator to change the equiv.tcl  
file to  
      define the rules that give a logical name to the current directory.
```

If you respond **2**, the job will not be submitted.

If you respond **1**, a file named `.vnc` will be created in the directory, and the job will be submitted. In the future, you will not receive the warning so long as the `.vnc` file exists.

The Altair Accelerator software internally uses logical names to refer to files.

Advanced Commands

1. Resource the cache file.

```
ls ~/ncadmin/vncjohn.swd/equiv.caches  
cayman pluto  
cayman abbcanova@cayman DEFAULT+SUPPORT html/ftnctraining > cat !$/pluto
```

```
cat ~/ncadmin/vncjohn.swd/equiv.caches/pluto  
VOVDIR /remote/release/VOV/7.0u2/linux64  
VOVDIR /remote/release/VOV/7.0u2/linux64/../common
```

2. Run the `vovequiv` command.

```
% vovequiv -p . # show the logical name for this directory
```

Resource Management

This exercise will walk you through some of the more advanced and complex issues of resource management in Accelerator. You should already be familiar with Accelerator and basic resource management, covered in the [Configure Resources](#) tutorial.

With Accelerator, Altair Accelerator includes a simplified version of our Monitor product, called Monitor-basic, which interfaces between NC and licensing systems like FlexNet Publisher. Unlike the full LM product, LM-basic does not do history or denials.

Review

The following topics are review from the [Configure Resources](#) tutorial:

- Find and view `resources.tcl` file
- The `vtk_resourcemap_set` procedure
- Static Resource (PRIORITY_LOW) configuration example
- Configure resources map on the fly using `vtk_resourcemap_set`

Topics of this tutorial include:

- Monitor-basic setup
- `vtk_flexlm_monitor` procedure
- Browser-based setup
- Resource throttling

In this tutorial, we will start LM-basic, configure it, and use it to monitor licenses. There are two things necessary to monitor FlexNet Publisher features:

1. Configure and start the LM-basic `vovserver`.
2. Accept the default `vtk_flexlm_monitor_all`, or add specific `vtk_flexlm_monitor` statements to NC's `resources.tcl`

Altair Monitor-Basic Setup

Altair Accelerator uses a two-layer interface to FlexNet Publisher. There is a daemon `vovresourced` which is started by the NC server whenever there are `vtk_flexlm_monitor` statements in the `resources.tcl` file. `vovresourced` gets license information from `licmon` via HTTP, and uses it to maintain NC's License: `resources`.

In the case where many NC queues or FlowTracer projects are running, it greatly reduces the load on FlexNet Publisher and improves the consistency of the license information to run Monitor or Monitor-basic. Additionally, Monitor offers a browser interface which shows licenses in use and license utilization information. Altair Accelerator recommends that you run this whenever you need to monitor FlexNet Publisher licenses, even in cases where you only have one server running Accelerator.

Configure and Manage Monitor-basic

Both Monitor-basic and full Monitor are managed by the `lmmgr` command. It is essential that you use this instead of other ways to start the LM vovserver, because it also takes care of any DB schema updates, auxiliary daemon and startup changes needed by newer versions.

You use the configuration file of `vovflexlmd` in Monitor-basic to specify which FlexNet Publisher licenses you want the daemon to monitor. You do not need to monitor all licenses. In this exercise, we will monitor one. In most cases it is easier to use the browser UI to configure which license servers are monitored.

1. Change to the working directory of the daemon and see whether it is already running.
2. If you see an `info.tcl` file, then check whether the process named there exists on that system.

```
% cd $VOVDIR/local/vovflexlmd; ls
```

Example `info.tcl` file:

```
# This file is created automatically by vovlmd: DO NOT EDIT
# If you touch this file, the process 17400 will exit.
set host      "jaguar"
set pid       17400
set port      5555
set cwd       "/remote/proj9/cadmgr/licmon/licmon.swd/vovlmd"
set version   "2.0"
set timestamp 1441146545; # Tue Sep 01 15:29:05 PDT 2019
```



Note: If this directory already contains a file named `config.tcl`, your site may already be running. In this case, check for the file `info.tcl`. This file will contain the host name, process ID, and port of the daemon. You can verify that it is running by pointing your web browser to the URL `http://host:port` using the port and host in the info file.

Even if the daemon is not running, we will use our own subdirectory to run our instance Monitor-basic, rather than use the default directory.

3. Create a directory to run your own Monitor-basic instance.

```
% mkdir ~/ncadmin/licmon
% cd ~/ncadmin/licmon
```

4. Start Monitor-basic, picking a port which is different from the regular one, if you are running on the Accelerator server host. The default TCP/IP port number is 5555. You can find the ports in use by `vovproject list -a -l`. For this example, we use port 5575.

```
% cd ~/ncadmin/licmon
% lmmgr -name licmon<user> -dir . -port 5575
```

5. Prepare a configuration file with a text editor, which names one of the FlexNet Publisher license files at your site.

```
% vovproject enable licmon<user>
% cd `vovserverdir -p vovlmd`
```

```
% vi config.tcl
```

6. Edit the default configuration file `config.tcl`, and enter at least one license file to monitor, using the `add_LM_LICENSE_FILE` procedure. The license may be in any format acceptable as a `-c` parameter to `lmstat`, that is, a full pathname or in `port@host` notation. It is better to use `port@host` than file paths, so that NFS is not involved.

Example `config.tcl` file.

```
# config.tcl -- vovflexlmd FLEXlm configuration
add_LM_LICENSE_FILE -tag CDNS /remote/vendors/cadence/license/license.dat
add_LM_LICENSE_FILE -tag SNPS /remote/vendors/synopsys/license/pluto.dat
add_LM_LICENSE_FILE -tag SUNW 1726@saturn
```

7. Verify that LM-basic is running and supplying license information. Examine the `info.tcl` file created by the `vovlmd` daemon, and point your web browser to the URL as described above.
8. Stop the LM-Basic `vovserver`.

Most of the Accelerator daemons may be stopped by touching the `info` file that they create. When it starts, the daemon records the timestamp of the file. Each cycle, the daemon examines the timestamp, and exits if it has changed.

For Monitor-basic, the auxiliary daemons like `vovlmd` will stop when the Monitor-basic `vovserver` stops.

vtk_flexlm_monitor Procedures

Your Accelerator setup will get license in-use info from your Monitor or Monitor-basic setup via HTTP on Monitor's `/raw` interface. The default `resources.tcl` file for Accelerator includes the `vtk_flexlm_monitor_all` procedure. This converts every feature monitored by Monitor into an Accelerator resource. If the resource name is specified, then this name is the actual resource name used. If the resource name is not specified, the name defaults to `License:<feature>`.

If you do not wish to have a large number of resources, you can comment out the above and put calls to the `vtk_flexlm_monitor` procedure in your `resources.tcl` file to specify which FlexNet Publisher features Accelerator should monitor.

Resource Throttling

In some cases, you may wish to restrict the number of a feature which are consumed by Accelerator jobs, to leave some for 'fast' or interactive use.

To accomplish this, you can use an auxiliary resource called `throttle_<resource>` to restrict the number of Accelerator jobs. For example, you have 15 licenses of `hsim`, and want to leave 1 free for jobs that bypass the Accelerator system. You would create a resource `'throttle_hsim'` with a count of 14, by adding statements like this. Map `hsim_license` to `throttle_hsim`, so that when

the throttle resource is exhausted, jobs that need hsim will queue. Often such resources are named Limit:<something>.

```
vtk_flexlm_monitor hsim hsim_license throttle_hsim  
vtk_resource_map_set throttle_hsim 14
```

Upgrade Accelerator

There are two types of updates for Accelerator, patches and new versions. Patches replace specific files in an installation and save an archive of the changed files so you can revert. A new release is a complete set of files, usually installed in a directory that is a sibling of your current version.

The simplest method to change to a different server is to use the following steps (shutdown and restart on new version):

- Install new software
- Notify users that Accelerator will be shutdown
- At an idle time, shut down the Accelerator server with `ncmgr stop`
- Change startup files (`.vovrc`) to refer to the new software installation
- Start the server, running the new version, with `ncmgr start`
- Notify users that Accelerator is available again

In some cases, for example, it may be impractical to shut down the Accelerator server for even a short time:

- you have a very busy installation, and Accelerator is never idle
- you have jobs running and you do not want to lose them
- you want to keep the order of jobs in the queue. In such cases you can use one of the following methods.
 1. Switch `vovserver` and `vovtaskers` separately, using `ncmgr stop -freeze`.
 2. Start a new Accelerator queue on new version, sending new jobs to it, shutting down the current Accelerator queue after all jobs have been retired.

Here are the steps for the `ncmgr stop -freeze` method.

1. From a shell with your current Altair Accelerator version, stop `vovserver` with `ncmgr stop -freeze`. This instructs `vovtaskers` with jobs to wait for a new `vovserver` and reconnect to it.
2. `vovtaskers` with no jobs should exit right away, and ones with jobs will enter the SUSP state. Their names will change to indicate they are stopping (and to permit new `vovtaskers` to start with the regular names).
3. From a shell with the new Altair Accelerator version, start the `vovserver` using `ncmgr start`.
4. Check on the browser UI Admin page that `vovserver` is running the expected version.

Here are the steps for the overlapping queues method.

1. Create a new queue, running the new server software
2. Direct `nc run` to send new jobs to the new queue
3. After some time, all jobs in the old queue have completed
4. The old queue may be now be shut down

Overlapping Queues

Please read these steps to the end and understand them before doing the procedure. This procedure is preliminary, and may need to be adapted to your Accelerator configuration.

1. Install and verify the new software version. The `vovcheck` command may be helpful.
2. Start a new queue, for example with name `vnc2`.


```
% ncmgr start -queue vnc2
```

3. Copy configuration files (`policy.tcl`, `resources.tcl`, `taskers.tcl` etc.) from the configuration directory of the old queue to that of the new queue in such a way that the two queues are functionally the same.
4. Get the new queue ready.

```
% ncmgr reset -queue vnc2 -full
```


5. In the `setup.tcl` file of the new queue, set the environment variables `NC_OLDQUEUE` and `NC_OLDVERSION` to point to the old queue.

```
# This is a fragment of the setup.tcl file for the new queue  
# USE THESE FOR MIGRATION FROM AN OLD QUEUE.  
setenv NC_OLDQUEUE  
vnc setenv NC_OLDVERSION 5.4.7
```

 **Note:** This goes in the `setup.tcl` file so that these environment variables will always be set in the context of `vnc2`.

6. Test the new queue. By setting the environment variable `NC_QUEUE` to `vnc2`, the Accelerator commands will use that as the default queue.


```
% setenv NC_QUEUE "vnc2"  
% nc run sleep 10  
% nc list  
% nc mon
```

 **Note:** You may use the `-queue` option as an alternative to setting `NC_QUEUE`. For example, you might use

```
% nc -queue vnc list -a      # Show all jobs in old queue.  
% nc -queue vnc2 run sleep 10 # Submit a sleep job to new queue.  
% nc -queue vnc2 list      # Show your jobs (only) in new  
queue.
```

7. Once you are satisfied that the second queue is ready, make it the default queue. The default queue is always named `vnc`, and is where the newly-submitted jobs will be placed.

```
% cd $VOVDIR/local/vncConfig  
% mv vnc.tcl vnc1.tcl; ln -s vnc2.tcl vnc.tcl
```


 **Note:** In the example above, `vnc.tcl` is moved to `vnc1.tcl` to preserve its contents, in case you want to reactivate the original queue. The two commands are on the same line, separated by a semicolon, so that the file `vnc.tcl` will be unavailable for the shortest possible time. Check carefully!

Now all the newly-submitted jobs will go to `vnc2`, while commands such as `nc info`, `nc stop`, `nc forget`, etc. will be run on both queues.

8. Error messages are filtered out, because a job ID which is valid in the first queue will not be valid in the second, and conversely. To see the results with messages, query the queues separately by using the `-queue` option of Accelerator commands.
9. When all jobs in the old queue have been retired, you can shut down the old queue.

Index

Special Characters

.cshrc [42](#)
.profile [42](#)

A

access to help [40](#)
add additional environment directories [93](#)
add workstation/offhours vovtaskers [87](#)
advanced commands [98](#)
advanced policy configuration [79](#)
Altair Accelerator Administrator Tutorials [67](#)

B

bin/bash [42](#)
bin/csh [42](#)
bin/tcsh [42](#)
browser-based setup [76](#)

C

compose good environment scripts [94](#)
configuration examples [81](#), [94](#)
configure an environment [91](#)
configure and manage Monitor-basic [100](#)
configure default resources [79](#)
configure FairShare [77](#)
configure policy.tcl for FairShare [77](#)
configure resources [81](#)
configure security [84](#)
configure taskers [86](#)
customize actions needed to enable access to Altair Accelerator products on Windows [47](#)

D

define a logical HOME directory name [96](#)

E

enable Altair Accelerator for non-interactive shells [43](#)
enable CLI access on UNIX [42](#)
enable CLI access on Windows [46](#)
enable the command prompt to communicate with a running product server [47](#), [48](#)
enable the shell to communicate with a running product server [44](#)
enforce job resource rules [80](#)

F

find and view example environment scripts [91](#)
find and view resources.tcl file [81](#)
find the server working directory [71](#)
find the URL [76](#)

G

get detailed information about a job [60](#)
get summary information [51](#)

H

help, Accelerator [40](#)

I

invoke the GUI [63](#)

J

job control [57](#)
jobs, show current information [63](#)

L

locate server configuration directory to find policy.tcl [77](#)
locate the security configuration file: security.tcl [84](#)
logical names (equivalences [96](#)

M

method 1: use windows explorer to set command line environment [46](#)
method 2: using Windows command prompt to set command line environment [46](#)
metrics, scheduler [63](#)
Monitor Basic setup [99](#)
monitoring jobs, taskers and resources [61](#)

N

nc_gui [63](#)

O

online help [40](#)
overlapping queues [104](#)

P

PDF, access [40](#)

R

- rerun jobs [59](#)
- resource management [99](#)
- resource monitor [82](#)
- resource throttling [101](#)
- restart the Accelerator queue [75](#)
- run basic jobs [49](#)
- run jobs with various options [54](#)

S

- save the new configuration [77](#)
- scheduler metrics [63](#)
- security configuration examples [84](#)
- server configuration parameters [78](#)
- setup using the web page [76](#)
- start a queue [71](#)
- start a test queue [71](#)
- start a vovtasker from the command line [88](#)
- start Accelerator [74](#)
- start newly defined vovtaskers [87](#)
- start/stop Accelerator [74](#)
- start/stop the RLM license server [70](#)
- start/stop vovtaskers [88](#)
- stop Accelerator [75](#)
- submit multiple jobs at once [54](#)

T

- test the new FairShare configuration [78](#)
- third party licenses [viii](#)
- troubleshooting [66](#)
- troubleshooting the UNIX setup [45](#)

U

- upgrade Accelerator [103](#)
- use a specific queue [73](#)
- use Accelerator help [40](#)
- use environments: -e <env> [54](#)
- use of resources: -r <res1 res2 ...> [55](#)
- use the web browser [65](#)

V

- verify access to Altair Accelerator products [43](#)
- verify the license file [68](#)
- view Accelerator status [74](#)

VOV_STDOUT_SPEC [40](#)
vovbrowser [40](#)
vovbuild [40](#)
vovconsole [63](#)
vovdoc [40](#), [40](#)
vovid [40](#)
vovserver [40](#)
vovtasker configuration [86](#)
vtk_flexlm_monitor procedures [101](#)

W

wait for jobs [55](#), [56](#)