PROGRESS®
ORBIX®

Orbix IBM Tivoli Integration Guide

Version 6.3.5, July 2011

PROGRESS software
BUSINESS MAKING PROGRESS™

© 2011 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Soft ware Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

Actional, Apama, Artix, Business Empowerment, Business Making Progress, DataDirect (and design), DataDi rect Connect, DataDirect Connect64, DataDirect Technologies, DataDirect XML Converters, DataDirect XQuery, DataXtend, Dynamic Routing Architecture, EdgeXtend, Empowerment Center, Fathom, Fuse Media tion Router, Fuse Message Broker, Fuse Services Framework, IntelliStream, IONA, Making Software Work Together, Mindreef, ObjectStore, OpenEdge, Orbix, PeerDirect, POSSENET, Powered by Progress, Pow erTier, Progress, Progress DataXtend, Progress Dynamics, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress OpenEdge, Progress Profiles, Progress Results, Progress Software Business Making Progress, Progress Software Developers Network, Progress Sonic, ProVision, PS Select, Savvion, SequeLink, Shadow, SOAPscope, SOAPStation, Sonic, Sonic ESB, SonicMQ, Sonic Orchestration Server, SpeedScript, Stylus Studio, Technical Empowerment, WebSpeed, Xcalia (and design), and Your Software, Our Technology-Experience the Connection are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. AccelEvent, Apama Dashboard Studio, Apama Event Manager, Apama Event Modeler, Apama Event Store, Apama Risk Firewall, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Cache-Forward, CloudEdge, DataDirect Spy, DataDirect SupportLink, Fuse, FuseSource, Future Proof, GVAC, High Performance Integration, Object Store Inspector, ObjectStore Performance Expert, OpenAccess, Orbacus, Pantero, POSSE, ProDataSet, Progress Arcade, Progress CloudEdge, Progress Control Tower, Progress ESP Event Manager, Progress ESP Event Modeler, Progress Event Engine, Progress RFID, Progress RPM, PSE Pro, SectorAlliance, SeeThinkAct, Shadow z/Services, Shadow z/Direct, Shadow z/Events, Shadow z/Presentation, Shadow Studio, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, Smart Frame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Sonic Business Integration Suite, Sonic Process Manager, Sonic Collaboration Server, Sonic Continuous Availability Architecture, Sonic Database Service, Sonic Workbench, Sonic XML Server, The Brains Behind BAM, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks con tained herein may be trademarks of their respective owners.

Third Party Acknowledgements:

Progress Orbix v6.3.5 incorporates Jakarata-struts 1.0.2 from the Apache Software Foundation (http://www.apache.org). Such Apache Technology is subject to the following terms and conditions: The Apache Soft ware License, Version 1.1 Copyright (c) 1999-2001 The Apache Software Foundation. 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 copy right notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "The Jakarta Project", "Struts", and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MER CHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DIS CLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBU TORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUEN TIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUB STITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIA BILITY, 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. This software consists of voluntary contributions made by many individuals on behalf of the Apache Soft ware Foun dation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates Jakarta-bcel 5.0 from the Apache Software Foundation (http://www.apache.org). Such Apache Technology is subject to the following terms and conditions: The Apache Software License, Version 1.1 Copy right (c) 2001 The Apache Software Foundation. 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 docu mentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Apache" and "Apache Software Foundation" and "Apache BCEL" must not be used to endorse or promote products derived from this software with out prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", "Apache BCEL", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``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 APACHE SOFTWARE FOUNDATION OR ITS 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 BUSI NESS 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. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Founda tion. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates Jakarat-regexp 1.2 from the Apache Software Foundation (http://www.apache.org). Such Apache Technology is subject to the following terms and conditions: The Apache Software License, Version 1.1 Copyright (c) 1999 The Apache Software Foundation. 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. 3. The end-user documentation included with the redistri bution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "The Jakarta Project", "Jakarta -Regexp", and "Apache Software Foundation" and "Apache BCEL" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``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 APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIA BLE 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. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates the Jakarta-log4j 1.2.6 from the Apache Software Foundation (http://www.apache.org). Such Apache Technology is subject to the following terms and conditions: The Apache Software License, Version 1.1 Copyright (c) 1999 The Apache Software Foundation. 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 dis claimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in

the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "log4j" and "Apache Software Foundation" and "Apache BCEL" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written per mission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``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 APACHE SOFTWARE FOUNDATION OR ITS 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 LIABIL ITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUD ING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foun dation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates Ant 1.5 from the Apache Software Foundation (http://www.apache.org). Such technology is subject to the following terms and conditions: The Apache Software License, Version 1.1 Copyright (c) 2000-2002 The Apache Software Foundation. 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 fol lowing 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 end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Ant" and "Apache Software Foundation" and "Apache BCEL" must not be used to endorse or promote products derived from this software without prior writ ten permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MER CHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS 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. This software consists of voluntary contri butions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates Xalan-j 2.3.1 from the Apache Software Foundation (http://www.apache.org). Such Apache Technology is subject to the following terms and conditions: The Apache Soft ware License, Version 1.1. Copyright (c) 1999 The Apache Software Foundation. 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. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Xalan" and "Apache Software Foundation" and "Apache BCEL" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MER CHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS 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. This software consists of voluntary contri butions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates the Xerces-c++ 2.4 from the Apache Software Foundation (http://www.apache.org). Such Apache Technology is subject to the following terms and conditions: The Apache Software License, Version 1.1. Copyright (c) 1999-2001 The Apache Software Foundation. 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. Redis tributions 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 end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Xerces" and "Apache Software Foundation" and "Apache BCEL" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANT ABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE

APACHE SOFTWARE FOUNDATION OR ITS 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. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Founda tion. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates xerces-j 2.5 from the Apache Software Foundation (http://www.apache.org). Such Apache Technology is subject to the following terms and conditions: The Apache Software License, Version 1.1. Copy right (c) 1999-2002 The Apache Software Foundation. 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. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTIC ULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS 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. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates the Tomcat 4.0.4 from the Apache Software Foundation (http://www.apache.org). Such Apache Technology is subject to the following terms and conditions: The Apache Software License, Version 1.1. Copyright (c) 1999, 2000 The Apache Software Foundation. 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. Redis tributions 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 end-user documentation included with the

redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "The Jakarta Project", "Tomcat" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANT ABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS 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. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foun dation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates MCPP 2.6.4 from the MCPP Project. Such technology is subject to the following terms and conditions: Copyright (c) 1998, 2002-2007 Kiyoshi Matsui kmatsui@t3.rim.or.jp All rights reserved. This software including the files in this directory is provided under the following license. Redistribu tion 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 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, PROCURE MENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CON TRACT, 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.

Progress Orbix v6.3.5 incorporates Xalan c++ v1.7 from The Apache Software Foundation. Such technology is subject to the following terms and conditions: The Apache Software License, Version 1.1 Copyright (c) 1999-2004 The Apache Software Foundation. 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. Redis tributions in binary form must reproduce the above copyright notice, this list of conditions and the follow ing disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the follow-

ing acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Xalan" and "Apache Software Foundation" must not be used to endorse or promote prod ucts derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICU LAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS 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.
==========

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, Lotus Development Corporation., http://www.lotus.com. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress Orbix v6.3.5 incorporates Tcl 8.4.15 from Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties. Such technology is subject to the following terms and conditions: This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, 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 SPE CIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WAR RANTIES 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 MODIFI CATIONS. 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 (c) (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 (c) (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.

Progress Orbix v6.3.5 incorporates bzip2 1.0.2 from Julian Seward. Such Technology is subject to the following terms and conditions: This program, "bzip2" and associated library "libbzip2", are copyright (C) 1996-2002 Julian R Seward. 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. 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. 3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 4. 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 MERCHANT ABILITY 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. Julian Seward, Cambridge, UK.jseward@acm.org bzip2/libbzip2 version 1.0.2 of 30 December 2001.

Progress Orbix v6.3.5 incorporates zlib 1.2.3 from Jean-loup Gailly and Mark Adler. Such Technology is subject to the following terms and conditions: License /* zlib.h -- interface of the 'zlib' general purpose compression library version 1.2.3, July 18th, 2005 Copyright (C) 1995-2000 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 mis represented; 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 */

Progress Orbix v6.3.5 incorporates the MinML 1.7 from John Wilson. Such Technology is subject to the following terms and conditions: Copyright (c) 1999, John Wilson (tug@wilson.co.uk). 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 dis claimer in the documentation and/or other materials provided with the distribution. All advertising materials mention ing features or use of this software must display the following acknowledgement: This product includes software devel oped by John

Wilson. The name of John Wilson may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY JOHN WILSON "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 JOHN WILSON 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 LIABIL ITY, 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.

Progress Orbix v6.3.5 incorporates JDOM vbeta9 from JDOM. Such Technology is subject to the following terms and conditions: LICENSE.txt, v 1.10 2003/04/10 08:36:05 jhunter Exp $ Copyright (C) 2000-2003 Jason Hunter & Brett McLaughlin. All rights reserved. Redistribution and use in source and binary forms, with or with out 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. Redistribu tions in binary form must reproduce the above copyright notice, this list of conditions, and the dis claimer that follows these conditions in the documentation and/or other materials provided with the distribu tion. 3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <license AT jdom DOT org>. 4. Prod ucts derived from this soft ware may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <pm AT jdom DOT org>. In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the soft ware itself an acknowledgement equivalent to the following: "This product includes software developed by the JDOM Project (http://www.jdom.org/)." Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos. THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WAR RANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DIS CLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIA BLE 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 LIABIL ITY, 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. This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter AT jdom DOT org> and Brett McLaughlin <brett AT jdom DOT org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

Progress Orbix v6.3.5 incorporates OpenSSL 0.9.8i Copyright (c) 1998-2008 The OpenSSL Project Copyright (c) 1995-1998 Eric A. Young, Tim J. Hudson All rights reserved. Such Technology is subject to the following terms and conditions: The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to

OpenSSL please contact openssl-core@openssl.org. OpenSSL License - Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted pro vided that the following conditions are met: 1. Redistributions of source code must retain the above copy right 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. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)" 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. 5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. 6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)" THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``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 OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAM AGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERV ICES; 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. This product includes cryp tographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). - Original SSLeay License - Copyright (C) 1995-1998 Eric Young (eay@crypt soft.com) All rights reserved. This package is an SSL implementation written by Eric Young (eay@crypt soft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commer cial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com). Copy right remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package. Redistri bution and use in source and binary forms, with or with out modification, are permitted provided that the follow ing conditions are met: 1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of con ditions and the following dis claimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mention ing features or use of this software must display the following acknowledge ment: "This product includes crypto graphic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the rou tines from the library being used are not crypto graphic related :-). 4. If you include any Windows specific code (or a deriv ative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)" THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANT ABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPE CIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCURE MENT 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 POSSI BILITY OF SUCH DAMAGE. The licence and distribution terms for any publically available version or deriva tive of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Progress Orbix v6.3.5 incorporates PCRE v7.8 from the PCRE Project. Such Technology is subject to the following terms and conditions:

PCRE LICENCE

PCRE is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language. Release 7 of PCRE is distributed under the terms of the "BSD"licence, as specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself. The basic library functions are written in C and are freestanding. Also included in the distribution is a set of C++ wrapper functions.

THE BASIC LIBRARY FUNCTIONS

Written by: Philip Hazel

Email local part: ph10

Email domain: cam.ac.uk

University of Cambridge Computing Service,

Cambridge, England.

Copyright (c) 1997-2008 University of Cambridge

All rights reserved.

THE C++ WRAPPER FUNCTIONS

Contributed by: Google Inc.

Copyright (c) 2007-2008, Google Inc.

All rights reserved.

THE "BSD" LICENCE

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 University of Cambridge nor the name of

Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFT WARE IS PRO VIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRAN TIES OF MERCHANT ABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDI RECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCURE MENT 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.

Progress Orbix v6.3.5 incorporates IDL Compiler Front End 1 from Sun Microsystems, Inc. Copyright 1992, 1993, 1994 Sun Microsystems, Inc. Printed in the United States of America. All Rights Reserved. Such tech nology is subject to the following terms and conditions: This product is protected by copyright and distrib uted under the following license restricting its use. The Interface Definition Language Compiler Front End (CFE) is made available for your use provided that you include this license and copyright notice on all media and documentation and the software program in which this product is incorporated in whole or part. You may copy and extend functionality (but may not remove functionality) of the Interface Definition Language CFE without charge, but you are not authorized to license or distribute it to anyone else except as part of a product or program developed by you or with the express written consent of Sun Microsystems, Inc. ("Sun"). The names of Sun Microsystems, Inc. and any of its subsidiaries or affiliates may not be used in advertising or publicity per taining to distribution of Interface Definition Language CFE as permitted herein. This license is effective until termi nated by Sun for failure to comply with this license. Upon termination, you shall destroy or return all code and documentation for the Interface Definition Language CFE. INTERFACE DEFINITION LANGUAGE CFE IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. INTERFACE DEFINITION LANGUAGE CFE IS PROVIDED WITH NO SUPPORT AND WITHOUT ANY OBLIGATION ON THE PART OF Sun OR ANY OF ITS SUBSIDIARIES OR AFFILIATES TO ASSIST IN ITS USE, CORREC TION, MODIFICATION OR ENHANCEMENT. SUN OR ANY OF ITS SUBSIDIARIES OR AFFILIATES SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY INTERFACE DEFINITION LANGUAGE CFE OR ANY PART THEREOF. IN NO EVENT WILL SUN OR ANY OF ITS SUBSIDIARIES OR AFFILIATES BE LIABLE FOR ANY LOST REVENUE OR PROFITS OR OTHER SPECIAL, INDIRECT AND CONSE QUENTIAL DAMAGES, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Use, duplication, or disclosure by the government is subject to restrictions as set forth in subpara graph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19. Sun, Sun Microsystems and the Sun logo are trademarks or registered trademarks of Sun Microsys tems, Inc. SunSoft, Inc. 2550 Garcia Avenue, Mountain View, California 94043 NOTE: SunOS, Sun Soft, Sun, Solaris, Sun Microsystems or the Sun logo are trademarks or registered trademarks of Sun Micro systems, Inc.

Progress Orbix v6.3.5 incorporates LibXML2 2.4.24 from Daniel Veillard. Such Technology is subject to the following terms and conditions: Except where otherwise noted in the source code (trio files, hash.c and

list.c) covered by a similar license but with different Copyright notices: Copyright (C) 1998-2002 Daniel Veillard. All Rights Reserved. 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 with out limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Soft ware, 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 MERCHANTA BILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIA BILITY, 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. Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

=== trio.c, trio.h: Copyright (C) 1998 Bjorn Reese and Daniel Stenberg. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THIS SOFTWARE IS PROVIDED "AS IS" AND WITH OUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIB UTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER. ==== triop.h: Copyright (C) 2000 Bjorn Reese and Daniel Stenberg. Permission to use, copy, modify, and dis tribute this software for any purpose with or without

fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTIC ULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

==== hash.c: Copyright (C) 2000 Bjorn Reese and Daniel Veillard. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permis sion notice appear in all copies. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHAN TIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

===== list.c: Copyright (C) 2000 Gary Pennington and Daniel Veillard. Permission

to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER. ===

triodef.h, trionan.c, trionan.h: Copyright (C) 2001 Bjorn Reese Permission to use, copy, modify, and distribute this soft ware for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permis sion notice appear in all copies. THIS SOFTWARE IS PROVIDED ``AS IS'' AND

WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MER CHANTIBILITY AND FITNESS FOR A PARTICULAR PUR-POSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIV-ABLE MANNER.

==== triostr.c, triostr.h: Copyright (C) 2001 Bjorn Reese and Daniel Stenberg.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, pro vided that the above copyright notice and this permission notice appear in all copies. THIS SOFTWARE IS PRO VIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PUR POSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

Progress Orbix v6.3.5 incorporates ICU library 2.6 from IBM. Such Technology is subject to the following terms and conditions: Copyright (c) 1995-2009 International Business Machines Corporation and others. All rights reserved. Per mission is hereby granted, free of charge, to any person obtaining a copy of this soft-ware and associated documenta tion files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Soft ware is fur nished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. 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 PARTICU LAR PUR POSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDI RECT OR CONSEQUENTIAL DAMAGES, OR ANY DAM-AGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TOR TIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or other wise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder. All trade-marks and registered trademarks mentioned herein are the property of their respective owners.

Updated: 18-Jul-2011

Contents

List of Figures xix

Preface xxi
- What is covered in this book xxi
- Who should read this book xxi
- Organization of this book xxi
- Related documentation xxii
- Suggested path for further reading xxii
- Additional resources for help xxiii
- Document conventions xxiii

Chapter 1 Integrating with IBM Tivoli™ 1
- Introduction 2
- The IONA Tivoli Integration 5

Chapter 2 Configuring your IONA Product 9
- Setting up your Artix Environment 10
- Setting up your Orbix Environment 14

Chapter 3 Configuring your Tivoli Environment 19
- Creating a Tivoli Installation Bundle 20
- Installing the Resource Model in the Tivoli Server 22
- Pushing the Resource Model out to your Host 26
- Configuring the Resource Model for your Endpoint 28

Chapter 4 Extending to a Production Environment 31
- Configuring an Artix Production Environment 32
- Configuring an Orbix Production Environment 36

Chapter 5 Using the IONA Tivoli Integration 41
- Detecting Common Server Problems 42
- Tracking Server Performance Metrics 44

Stopping, Starting, and Restarting Servers 45

Appendix A IONA Tivoli Resource Model 47
Thresholds 48
Events 50
Parameters 51
WBEM/CIM Definition 52

Glossary 55

Index 59

List of Figures

Figure 1: Overview of the IONA Tivoli Integration 4
Figure 2: Example IONA Tivoli Deployment 7
Figure 3: Deployment Bundle Wizard 11
Figure 4: Run Deployer Dialog 12
Figure 5: Orbix Configuration GUI 14
Figure 6: Selecting Tivoli Agent Configuration 15
Figure 7: Selecting Performance Logging 16
Figure 8: Tivoli Profile Manager 23
Figure 9: Edit Resource Model 24
Figure 10: Contents of the IONA Server Task Library 27
Figure 11: The configure_provider Task 33
Figure 12: The configure_provider Task 38

Preface

What is covered in this book

IONA's products support integration with Enterprise Management Systems such as IBM Tivoli™, HP OpenView™, CA Unicenter™, and BMC Patrol™. This book explains how to integrate Orbix and Artix with IBM Tivoli.

Who should read this book

This book is aimed at system administrators using IBM Tivoli to manage distributed enterprise environments, and developers writing distributed enterprise applications. Administrators do not require detailed knowledge of the technology that is used to create distributed enterprise applications.

This book assumes that you already have a good working knowledge of the IBM Tivoli Management Framework and IBM Tivoli Monitoring (formerly known as Distributed Monitoring).

Organization of this book

This book contains the following chapters:

- Chapter 1 introduces Enterprise Management Systems, and IONA's integration with IBM Tivoli.
- Chapter 2 describes how to configure your IONA product for integration with IBM Tivoli.

- Chapter 3 describes how to configure your IBM Tivoli environment for integration with IONA products.
- Chapter 4 describes how to extend your integration from a test environment into a production environment.
- Chapter 5 explains how to perform common tasks such as tracking server metrics or starting a server.
- Appendix A lists the contents of the IONA Tivoli resource model, describing its thresholds, events and parameters.

Related documentation

The Orbix library includes the following related books:

- *Orbix Management User's Guide*
- *Orbix Administrator's Guide*
- *Orbix Management Programmer's Guide*

The Artix library includes the following related books:

- *Deploying and Managing Artix Solutions*
- *Designing Artix Solutions with Artix Designer*
- *IONA BMC Patrol Integration Guide*

For the latest versions of all IONA product documentation, see the IONA web site:

http://www.iona.com/support/docs

Suggested path for further reading

If you are new to Orbix, you should read the documentation in the following order:

1. *Orbix Management User's Guide*

   This guide explains how to use the IONA Administrator management tool, and how to configure the Orbix management service for integration with other management tools.

2. *Orbix Administrator's Guide*

   This guide explains in detail how to configure and manage the components of an Orbix environment.

3. *Orbix Management Programmer's Guide*

   This guide explains how programmers can enable applications to be managed by IONA Administrator.

Additional resources for help

The IONA Knowledge Base (http://www.iona.com/support/knowledge_base/index.xml) contains helpful articles, written by IONA experts, about Artix and other products.

The IONA Update Center (http://www.iona.com/support/updates/index.xml) contains the latest releases and patches for IONA products.

If you need help with this or any other IONA products, go to IONA Online Support (http://www.iona.com/support/index.xml).

Comments on IONA documentation can be sent to docs-support@iona.com .

Document conventions

This book uses the following typographical and keying conventions.

Typographical conventions

This book uses the following typographical conventions:

`Constant width`

Constant width (courier font) in normal text represents portions of code and literal names of items such as classes, functions, variables, and data structures. For example, text might refer to the `CORBA::Object` class.

Constant width paragraphs represent code examples or information a system displays on the screen. For example:

```
#include <stdio.h>
```

Italic	Italic words in normal text represent *emphasis* and *new terms*. Italic words or characters in code and commands represent variable values you must supply, such as arguments to commands or path names for your particular system. For example: `% cd /users/`*your_name* **Note:** Some command examples may use angle brackets to represent variable values you must supply. This is an older convention that is replaced with *italic* words or characters.

Keying conventions

This book uses the following keying conventions:

No prompt	When a command's format is the same for multiple platforms, a prompt is not used.
`%`	A percent sign represents the UNIX command shell prompt for a command that does not require root privileges.
`#`	A number sign represents the UNIX command shell prompt for a command that requires root privileges.
`>`	The notation > represents the DOS or Windows command prompt.
. . . . . .	Horizontal or vertical ellipses in format and syntax descriptions indicate that material has been eliminated to simplify a discussion.
[ ]	Brackets enclose optional items in format and syntax descriptions.
{ }	Braces enclose a list from which you must choose an item in format and syntax descriptions.
|	A vertical bar separates items in a list of choices enclosed in { } (braces) in format and syntax descriptions.

CHAPTER 1

Integrating with IBM Tivoli™

This chapter introduces the integration of IONA products with the IBM Tivoli™ Enterprise Management System (EMS).

In this chapter

This chapter contains the following sections:

Introduction	page 2
The IONA Tivoli Integration	page 5

Introduction

Overview

IONA's products support integration with Enterprise Management Systems such as IBM Tivoli. This section includes the following topics:

- "The application life cycle".
- "Enterprise Management Systems".
- "IONA EMS integration".
- "IONA Tivoli integration tasks".
- "Integration overview".

The application life cycle

Most enterprise applications go through a rigorous development and testing process before they are put into production. When applications are in production, developers rarely expect to manage those applications. They usually move on to a new project while the day-to-day running of the applications are managed by a production team. In some cases, the applications are deployed in a data center that is owned by a third party, and the team that monitors the applications belong to a different organization.

Enterprise Management Systems

Different organizations have different approaches to managing their production environment, but most will have at least one *Enterprise Management System* (EMS).

The main Enterprise Management Systems are IBM Tivoli, HP OpenView™, CA Unicenter™, or BMC Patrol™. These systems are popular because they give a top-to-bottom view of every part of the IT infrastructure. This means that if an application fails because the `/tmp` directory fills up on a particular host, for example, the disk space is reported as the fundamental reason for the failure. The various application errors that arise are interpreted as symptoms of the underlying problem with disk space. This is much better than being swamped by an event storm of higher level failures that all originate from the same underlying problem. This is the fundamental strength of integrated management.

IONA EMS integration

IONA's Orbix and Artix products are designed to integrate with Enterprise Management Systems. IONA's common management instrumentation layer provides a base that can be used to integrate with any EMS.

In addition, IONA provides packaged integrations that provide out-of-the-box integration with major EMS products. This guide describes IONA’s integration with the IBM Tivoli products.

IONA Tivoli integration tasks

The IONA Tivoli integration performs key enterprise management tasks (for example, posting an event if a server dies). This enables automated recovery actions to be taken.

The IONA Tivoli integration also tracks key server metrics (for example, number of invocations received; and average, maximum and minimum response times). Events can be generated when any of these parameters go out of bounds.

In addition, you can also perform an extensible set of actions on servers. The default actions are start, stop and restart.

Integration overview

In the IONA Tivoli integration, these key server performance metrics are logged by the IONA performance logging plugins. Log file interpreting utilities are then used to analyze the logged data. Figure 1 shows a simplified overview of the IONA Tivoli integration at work. In this example, a restart command is issued to an unresponsive server (for example, locator or naming service).

The IONA performance logging plugins collect data relating to server response times and log it periodically in the performance logs. The IONA Tivoli resource model executes periodically on each host and uses the IONA log file interpreter to collect and summarize the logged data. It compares the response times and other values against user defined thresholds. If these values exceed the threshold, an event is fired. This event can be used to trigger an option from the Tivoli task library to restart the unresponsive server.

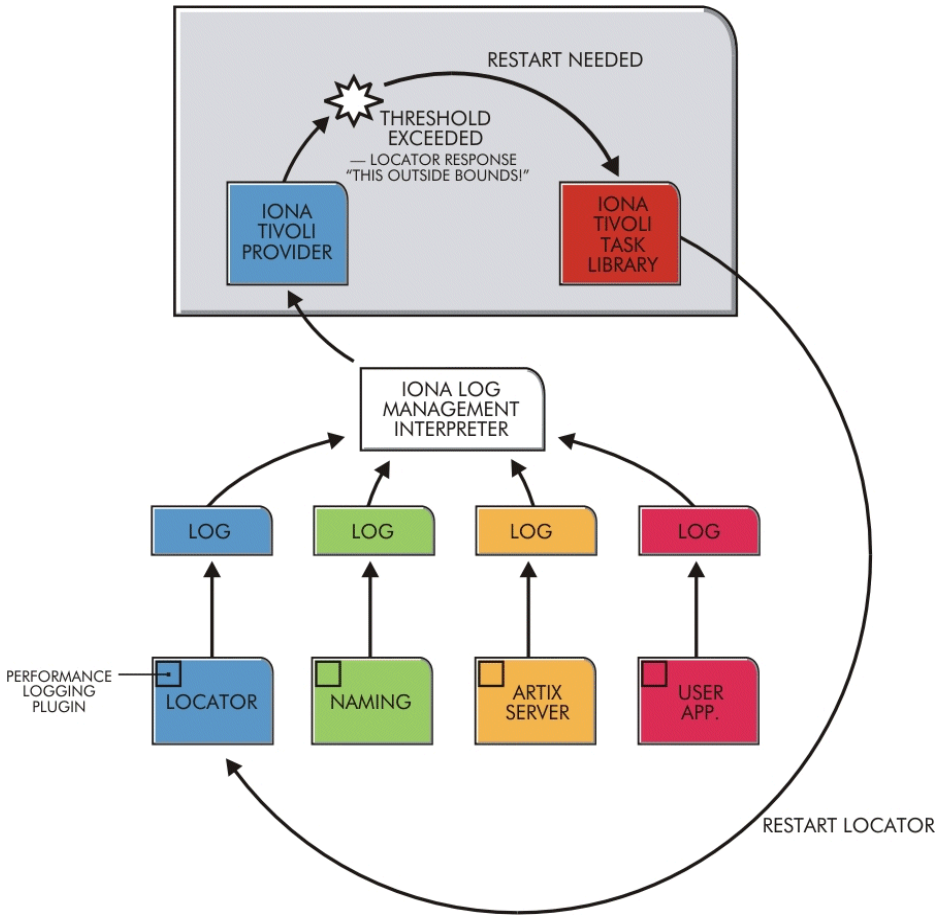

Figure 1: *Overview of the IONA Tivoli Integration*

The IONA Tivoli Integration

Overview

This section describes the requirements and main components of IONA's Tivoli integration. This section includes the following topics:

- "IONA requirements".
- "Tivoli requirements".
- "Main components".
- "IONA Tivoli resource model".
- "IONA Tivoli task library".
- "Integration and setup utilities".
- "Example IONA Tivoli deployment".

IONA requirements

IONA's Artix and Orbix products are fully integrated with IBM Tivoli. You must have at least one of the following installed:

- Artix 2.0.1
- Orbix 6.1, 6.2 or 6.3

Tivoli requirements

IONA's products are fully integrated with IBM Tivoli Management Framework and IBM Tivoli Monitoring.

To use the IONA Tivoli integration, you must have at least the following versions installed:

- IBM Tivoli Management Framework 4.1 or higher.
- IBM Tivoli Monitoring 5.1.1 (Fix Pack 04) or higher.

Main components

The IONA Tivoli integration package contains three main parts:

- A Tivoli Monitoring resource model.
- A Tivoli task library.
- Integration and setup utilities.

IONA Tivoli resource model

For an introduction to Tivoli resource models, see the IBM *Tivoli Monitoring User Guide*. The IONA Tivoli resource model enables Tivoli to track key attributes of Artix and Orbix services and customer-built servers that are based on Artix and Orbix. These attributes include:

- Server liveness.
- Number of incoming invocations received by the server.
- Maximum, average, and minimum response times of the server.

The resource model defines events that fire when a server's liveness cannot be verified, or when any of the other attribute values go beyond thresholds that can be set by the user.

The IONA Tivoli resource model is described in detail in Appendix A.

IONA Tivoli task library

The IONA Tivoli task library contains a set of tasks that can be used to configure and check the IONA Tivoli integration.

This task library can also be used to start, stop, or restart monitored servers. It can also be extended to perform any number of actions on a monitored server. These actions can be performed automatically as a result of receiving an event. For example, if an event fires to indicate that a server is no longer alive, you can configure Tivoli to use the IONA Tivoli task library to issue a restart for that server.

Integration and setup utilities

Both the IONA Tivoli resource model and task library must be installed and configured to work correctly. The IONA Tivoli integration package contains a number of setup utilities that help you achieve this task. These utilities are described in detail in "Configuring your Tivoli Environment" on page 19.

Example IONA Tivoli deployment

The high-level overview in Figure 2 shows a typical deployment of an IONA Tivoli integration. This deployment is explained as follows:

1. The IONA Tivoli resource model and task library are installed on the Tivoli region server.
2. The administrator customizes a monitoring profile based on the IONA Tivoli resource model.
3. The monitoring profile is distributed through the gateways to each of the Tivoli endpoints (managed hosts). In this example, there are three Tivoli endpoints—two based on Windows, and one on Solaris.

4. The monitoring profile executes inside the Tivoli Monitoring Agent, periodically checking the status and response times of the IONA services and IONA-based applications.

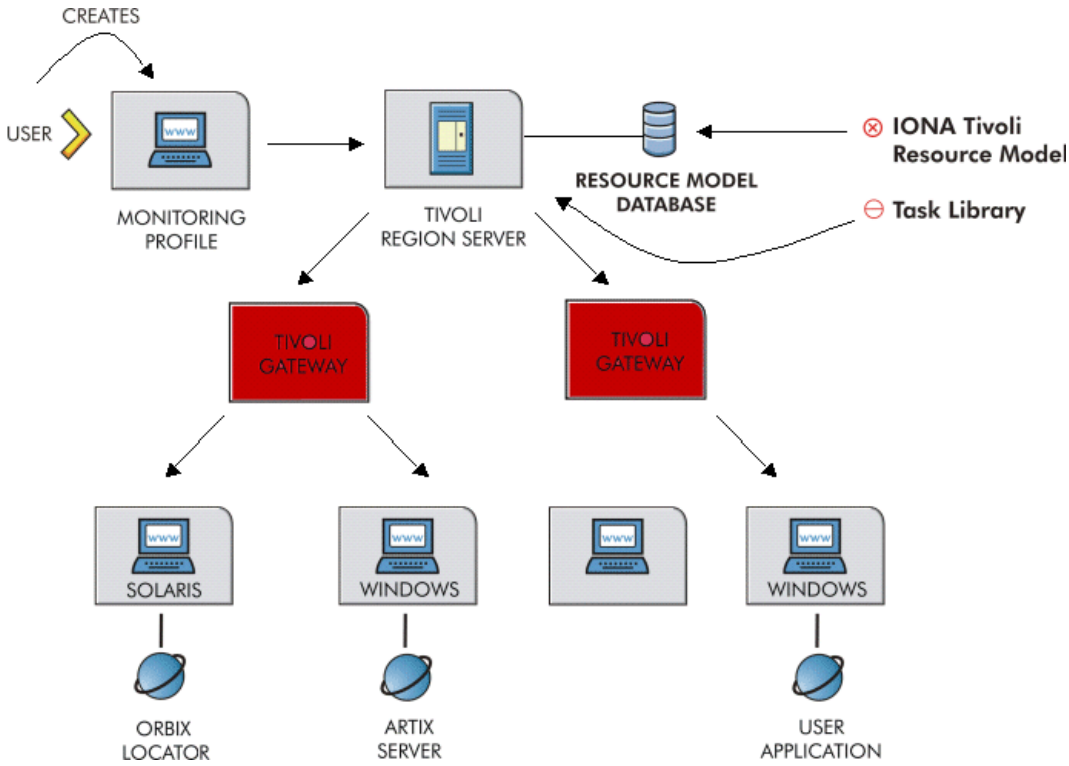

Figure 2: *Example IONA Tivoli Deployment*

These steps are explained in more detail in "Configuring your Tivoli Environment" on page 19 and "Extending to a Production Environment" on page 31.

CHAPTER 2

Configuring your IONA Product

This chapter explains the steps that you need to perform in Artix or Orbix so that they can be managed using IBM Tivoli.

In this chapter

This chapter contains the following sections:

Setting up your Artix Environment	page 10
Setting up your Orbix Environment	page 14

Setting up your Artix Environment

Overview

The best way to learn how to use the IONA Tivoli integration is to start with a host that has both Tivoli and Artix installed. This section explains the configuration steps in your Artix environment. It includes the following topics:

- "Enabling management".
- "Generating EMS configuration files".
- "The servers.conf file".
- "The server_commands.txt file".
- "Stopping Artix applications on Windows".
- "Further information".

Enabling management

You can use the **Artix Designer** GUI tool to enable management for your Artix applications. To enable management, perform the following steps:

1. Select **File|New Deployment Profile** and follow the steps in the wizard. This creates a platform-specific deployment profile.

   Typically, you would have a separate profile for each deployment machine (for example, Windows or UNIX).

2. Select **File|New Deployment Bundle**, and follow the steps in the wizard. In the **Bundle Details** panel, select the **Enable Management** checkbox, as shown in Figure 3.

   You can create as many deployment bundles as you like, but they must all be associated with one deployment profile.

For more detailed information about deployment bundles and profiles, and using the **Artix Designer** tool, see *Designing Artix Solutions*.

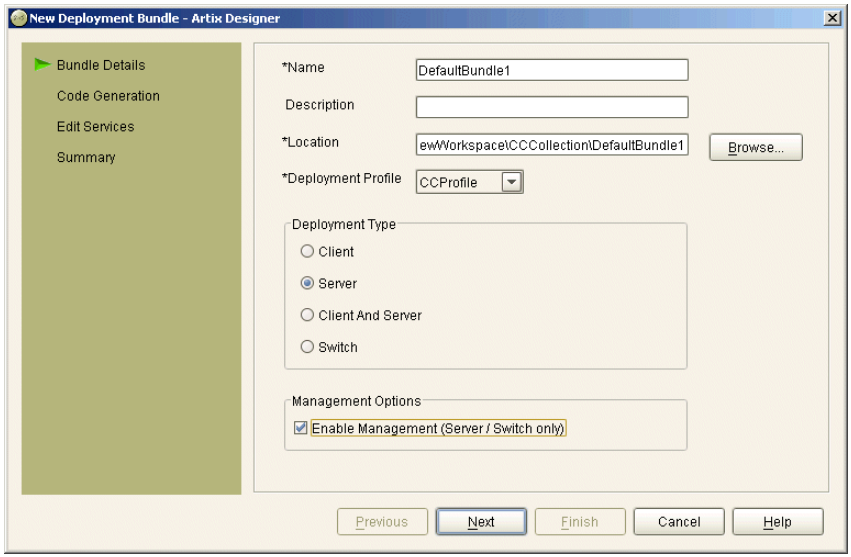

Figure 3: *Deployment Bundle Wizard*

Generating EMS configuration files

You can use **Artix Designer** to generate EMS configuration files for your Artix applications. To generate these files, perform the following steps:

1. Select **Tools**|**Generate Code**.
2. Select the **Generate** checkbox for **Environment Scripts**, as shown in Figure 4.
3. Select **OK**.

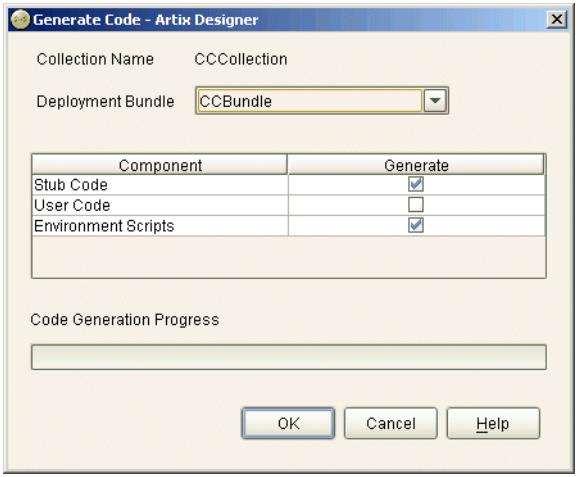

Figure 4: *Run Deployer Dialog*

The **Artix Designer** tool generates two files that are used to configure the Tivoli integration:

- `servers.conf`
- `server_commands.txt`

These generated files are created in the `etc` directory of your Artix deployment bundle, for example:

```
C:\artix\myWorkspace\myCollection\myBundle\etc
```

The servers.conf file

When you open the `servers.conf` file, you will see an entry such as the following:

```
myapplication, 1, /path/to/myproject/log/myapplication_perf.log
```

This example entry instructs Tivoli to track the `myapplication` server. It reads performance data from the following log file:

```
/path/to/myproject/log/myapplication_perf.log
```

There will be one of these files for each application that you want to monitor. The IONA Tivoli resource model uses the `servers.conf` file to locate these logs and then scans the logs for information about the server's key performance indicators.

The server_commands.txt file

When you open the `server_commands.txt` file, you will see entries like the following:

```
myapplication,start=/path/to/myproject/bin/start_myapplication.sh
myapplication,stop=/path/to/myproject/bin/stop_myapplication.sh
myapplication,restart=/path/to/myproject/bin/restart_myapplication.sh
```

Each entry in this file references a script that can be used to stop, start or restart the `myapplication` server. For example, when the IONA Tivoli task library receives an instruction to start `myapplication`, it looks up the `server_commands.txt` file, and executes the script referenced in this entry.

Stopping Artix applications on Windows

On Windows, stop scripts are not generated by default. While it is straightforward to terminate a process on UNIX by sending it a kill signal, there is no straightforward equivalent on most Windows platforms.

On Windows XP, you can use the `taskkill` command in your stop scripts. On older versions of Windows, you can write your own stop scripts based on a variety of methods. There are many options for implementing a stop script including adding a Web service interface to control the shutdown of your server, or simply making use of a utility such as `pskill` from `www.sysinternals.com`.

See also, the following article on the Microsoft support pages:

How to terminate an application cleanly in Win32 (`http://support.microsoft.com/default.aspx?scid=kb;EN-US;q178893`)

Further information

For details of how to manually configure servers to use the performance logging, see “Configuring an Artix Production Environment” on page 32.

For a complete explanation of performance logging configuration, see the *Orbix Management User’s Guide*.

Setting up your Orbix Environment

Overview

The best way to learn how to use the IONA Tivoli integration is to start with an Orbix installation on a host that is also a Tivoli endpoint. This section explains the configuration steps in your Orbix environment. It includes the following:

- "Creating an Orbix configuration domain".
- "Generating EMS configuration files".
- "Configuring performance logging".
- "Tivoli configuration files".
- "The servers.conf file".
- "The server_commands.txt file".
- "Further information".

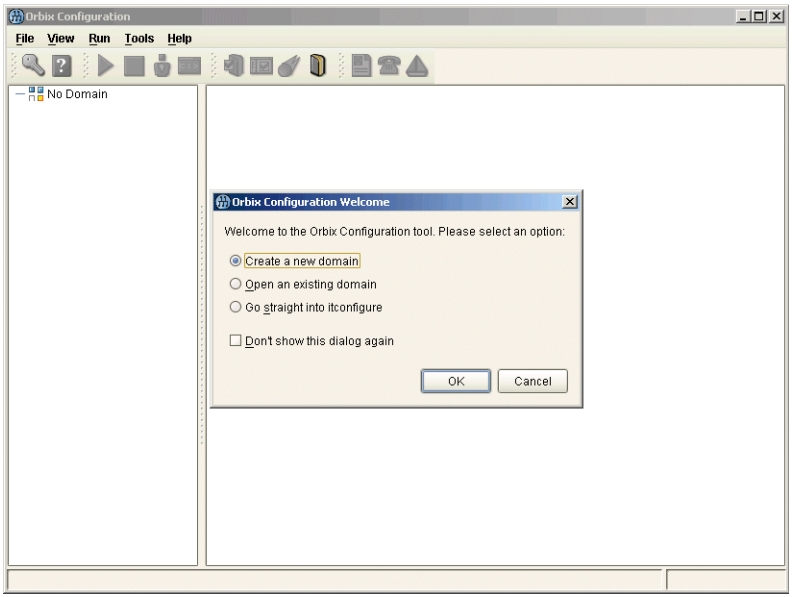

Figure 5: *Orbix Configuration GUI*

Creating an Orbix configuration domain

You must first create the Orbix configuration domain that you want to monitor using the **Orbix Configuration** GUI.

To start the **Orbix Configuration** GUI, enter `itconfigure` on the command line. The first screen is shown in Figure 5.

Generating EMS configuration files

To generate Tivoli agent configuration files, perform the following steps:

1. Click **Go straight into itconfigure** in the welcome dialog.

1. Select **File**|**New**|**Expert** from the GUI main menu. This displays the **Domain Details** screen, as shown in Figure 6.

2. Select the **Generate EMS Configuration Files** checkbox. This will generate configuration files required for your IONA Tivoli integration.

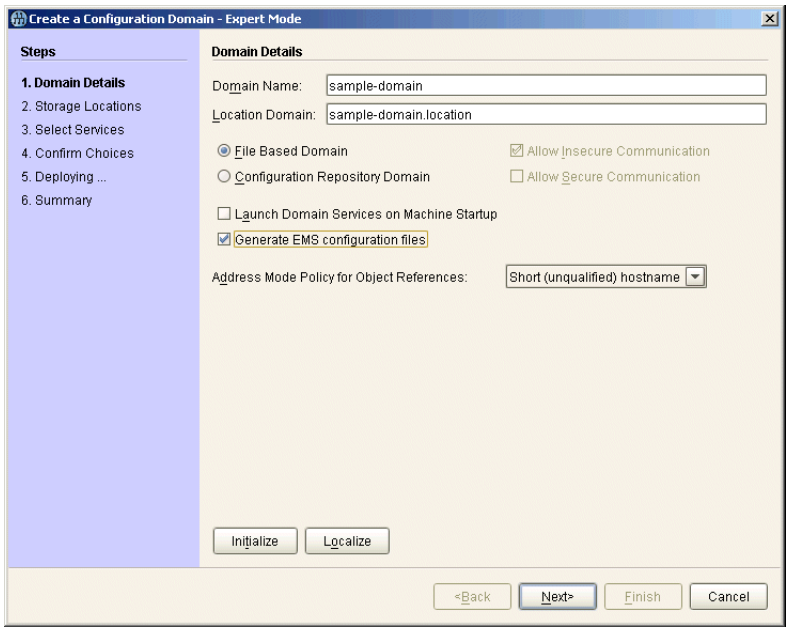

Figure 6: *Selecting Tivoli Agent Configuration*

3. Proceed as normal following the steps in the wizard until you get to the **Select Services** screen (see "Configuring performance logging").

Configuring performance logging

To configure performance logging, take the following steps:

1. In the **Select Services** screen, click **Settings** to launch the **Domain Defaults** dialog, shown in Figure 7.
2. Select the **Performance Logging** option in the **Other Properties** box, shown in Figure 7. This ensures that, by default, all your selected services are configured for monitoring.

   If you want to enable Tivoli to start, stop, or restart your servers, also select the **Launch Service on Domain Startup** option, in the **Service Launching** box.

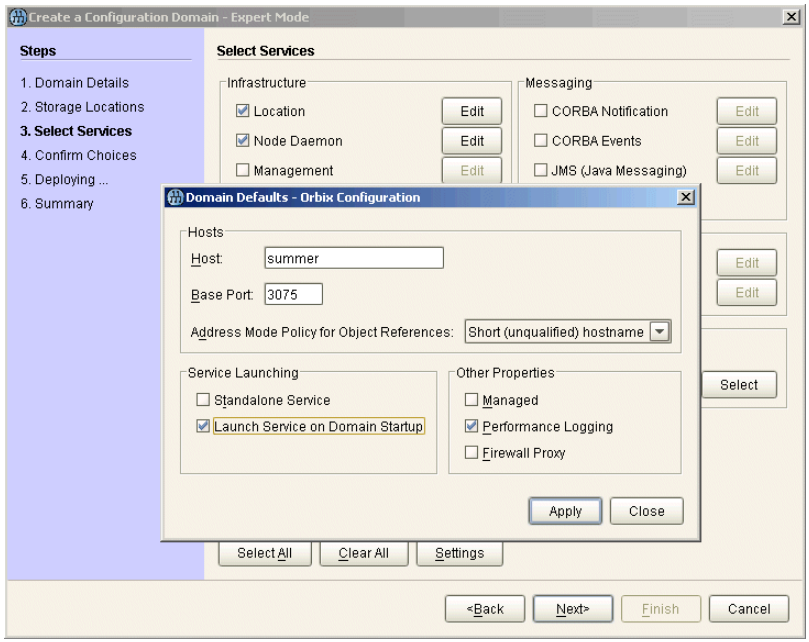

Figure 7: *Selecting Performance Logging*

Alternatively, you can configure these settings separately for each service by selecting the service, and clicking the **Edit** button.

3. Click **Apply**, and then **Close**.
4. Click **Next** to view a **Confirmation** screen for your selected configuration.

5. Click **Next** to deploy your configuration.
6. Click **Finish** to exit.

Note: When configuring Tivoli integration, you must also configure performance logging. This step is not optional. However, you can configure performance logging without Tivoli integration. For full details, see the *Orbix Management User's Guide.*

Tivoli configuration files

When the domain is created, you can start it like any other domain, using the start script in your `<orbix-install>/etc/bin` directory. Selecting the performance logging feature has enabled some extra configuration and logging. In your `<orbix-install>/var/domain-name` directory, you will find the following Tivoli configuration files:

`servers.conf`	Used by the IONA Tivoli resource model.
`server_commands.txt`	Used by the IONA Tivoli task library.

The servers.conf file

When you open the `servers.conf` file, you will see a number of entries in the following form:

`servername, number, /path/to/a/log/file`

For example:

```
mydomain_locator_myhost, 1,
   /opt/iona/var/mydomain/logs/locator_myhost_perf.log
```

The `servers.conf` file lists the servers that you want Tivoli to monitor on a particular host. To begin with, assume that you are running all services in the domain on one host. For example, assume your `servers.conf` file has the above entry. When you have started your domain, you should see a log file in the following location:

`/opt/iona/var/mydomain/logs/locator_perf.log`

There will be one of these files for each server that you want to monitor. The IONA Tivoli resource model uses the `servers.conf` file to locate these logs and then scans the logs for information about the server's key performance indicators.

The server_commands.txt file

When you open the `server_commands.txt` file, you will see a number of entries of the form:

`servername,action=/path/to/script`

For example:

```
mydomain_locator_myhost,start
=/opt/iona/var/mydomain/locator_myhost_start.sh
```

Each entry in this file contains a pointer to a script that implements an action on a particular server. In this example, the action is a start action for the server `mydomain_locator_myhost`. When the IONA Tivoli task library receives an instruction to start the locator in a domain named `mydomain` on a host named `myhost`, it looks up the `server_commands.txt` file on `myhost`, and execute the script pointed to in this entry.

Further information

For details of how to manually configure servers to use the performance logging plugins, see "Extending to a Production Environment" on page 31.

For a complete explanation of performance logging configuration, see the *Orbix Management User's Guide*.

CHAPTER 3

Configuring your Tivoli Environment

This chapter explains the steps that you must perform in your IBM Tivoli environment. It assumes that you already have a good working knowledge of the IBM Tivoli Management Framework and IBM Tivoli Monitoring (formerly known as Distributed Monitoring).

In this chapter

This chapter contains the following sections:

Creating a Tivoli Installation Bundle	page 20
Installing the Resource Model in the Tivoli Server	page 22
Pushing the Resource Model out to your Host	page 26
Configuring the Resource Model for your Endpoint	page 28

Creating a Tivoli Installation Bundle

Overview

Your Tivoli integration comes in a `.tar` file called `tivoli_integration.tar`. This section explains how to create a Tivoli install bundle from the `tivoli_integration.tar` file. You will create an install bundle named `tivoli_install.tar`.

Creating an install bundle

To create a Tivoli install bundle, perform the following steps:

1. Untar the `tivoli_integration.tar` file into any directory on the host that you want to monitor, using the following command:

   ```
   tar xvf tivoli_integration.tar
   ```

   There should be three subdirectories:

   ```
   bin
   resource-model
   task-library
   ```

2. Go into the `bin` directory and run the `create_tivoli_install_bundle` shell script.

   Note: This is a bash script. On Windows (with Tivoli installed), you must use the bash environment that is installed with Tivoli. If you invoke the script with no arguments, it prints out a page of instructions.

The `create_tivoli_install_bundle` script takes the following arguments:

Configuration directory	The configuration directory where the `servers.conf` and `server_commands.txt` files are located: **Artix** Your Artix workspace `etc` directory (for example, `C:/Documents and Settings/user-name/etc`). **Orbix** `<install-dir>/var/domain-name` **Note:** On Windows, you must use a forward slash character (`/`) when specifying this location.
Region name	The name of the Tivoli administrative region that you want this host/application to be in.
Profile manager	The name of the Tivoli profile manager that you want the IONA profile to be installed in.

3. Decide which region to use in your Tivoli deployment, and which profile manager you want the IONA profile to be installed in.
4. Run the `create_tivoli_install_bundle` shell script with all three values specified. This results in a new tar file called `tivoli_install.tar`.

Installing the Resource Model in the Tivoli Server

Overview

This section explains how to install the IONA Tivoli resource model from the `tivoli_install.tar` file that you created.

Installing the IONA Tivoli resource model

To install the IONA Tivoli resource model and task library into your Tivoli server, perform the following steps:

1. Transfer the `tivoli_install.tar` file to your Tivoli region server, and untar it to a temporary location, using the following command:

   ```
   tar xvf tivoli_install.tar
   ```

2. Start a Tivoli shell environment (see your Tivoli documentation for details). On Windows, type `bash`, to run in a bash shell. Change to your temporary location, and you will see a new directory structure starting with a directory named `iona`.
3. Change directory into `iona/bin`. This contains the following shell scripts:

   ```
   import_tll.sh
   create_profile.sh
   ```

4. Run the `create_profile.sh` script. This adds the IONA Tivoli resource model to the resource model database and creates a new profile named `IONAProfile`.
5. Open the Tivoli **Desktop** and select the region that you specified when you created the install bundle, followed by the profile manager that you specified. In the **Profile Manager** GUI, you will see a new profile called `IONAProfile`, as shown in Figure 8.

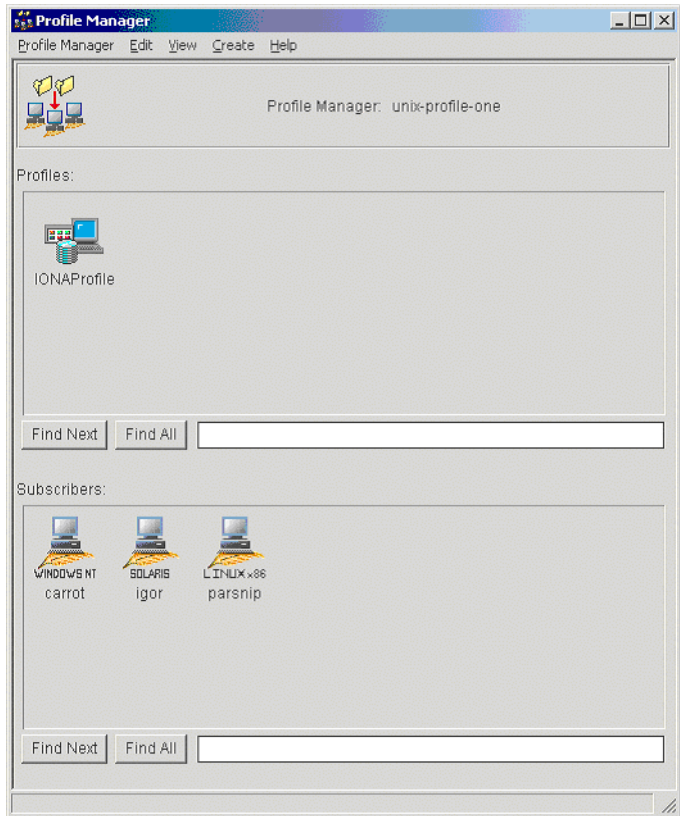

Figure 8: *Tivoli Profile Manager*

6. Open the `IONAProfile`, and then open the resource model `IONAServer Monitor`. You will see a resource model with default thresholds and indications, as shown in Figure 9.

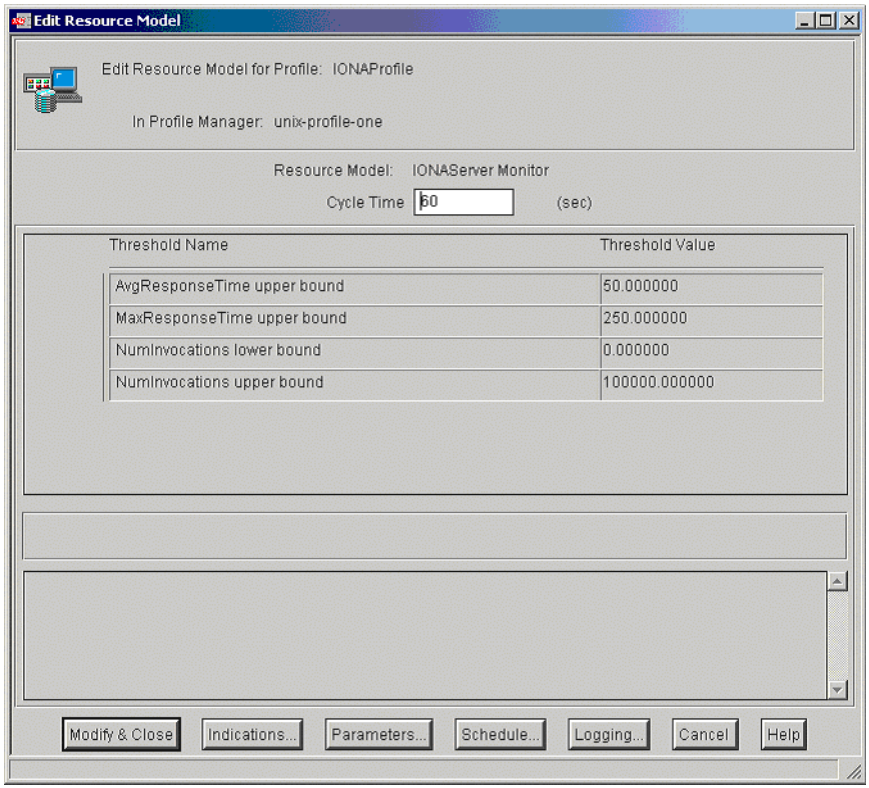

Figure 9: *Edit Resource Model*

Figure 9 shows that the profile has been initialized with default threshold values. Appendix A describes these thresholds in detail; you do not need to be concerned with these now.

7. If you want Tivoli to log historical data on the attributes of each server, click the **Logging...** button for the profile, and then check the box marked **Enable Data Logging** to put logging into effect. This will record historical data for each attribute.
8. Click **Modify & Close**.

Pushing the Resource Model out to your Host

Overview

This section explains how to push the IONA Tivoli resource model out to the endpoint where your IONA product is running (Orbix or Artix).

Pushing out the resource model

To push the IONA Tivoli resource model out to the endpoint where your IONA product is running, perform the following steps:

1. Add the Tivoli endpoint where your IONA product is installed as a subscriber to the profile manager, and distribute the `IONAProfile` to this endpoint.

   The resource model should now be running on the endpoint, but it will not yet be able to collect any meaningful data because it needs to be pointed to the `servers.conf` file.
2. Return to the directory where you untarred `tivoli_install.tar`, and change directory to `iona/bin`.
3. Run the `import_tll.sh` script. This installs the task library.
4. Reopen the Tivoli region that you are using on the desktop. You should now see a task library called `IONAServerTaskLibrary`.
5. Open the task library. It contains the following four tasks (also shown in Figure 10):

   ```
   check_deployment
   configure_provider
   list_server_commands
   server_command
   ```
6. Run the `check_deployment` task on the endpoint that contains your correctly configured Artix or Orbix installation. It prints out diagnostics to indicate that it has found your `servers.conf` file and your `server_commands.txt` file. This task also verifies the contents of these files.

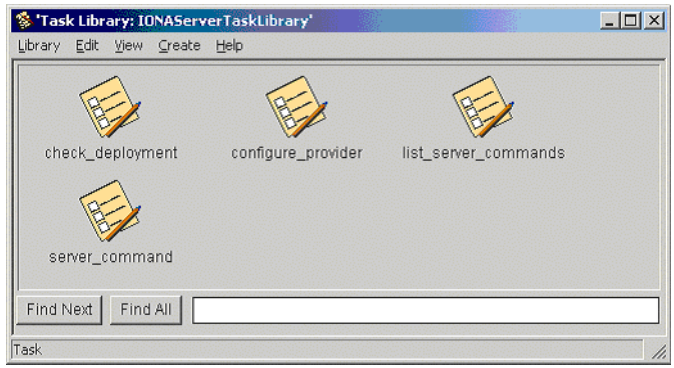

Figure 10: *Contents of the IONA Server Task Library*

7. If the `check_deployment` task runs successfully, try running `list_server_commands`. This shows a list of actions that you can run on each server, for example:

   ```
   mydomain_locator_myhost,stop
   ```

   Executing this command stops the Orbix locator in `mydomain`. You can execute any of these server commands by running the `server_command` task. This is an exercise for later (described in "Configuring the resource model" on page 28).

Configuring the Resource Model for your Endpoint

Overview

This section explains how to configure the IONA Tivoli resource model for your endpoint host, and how to test that your integration is configured correctly. It includes the following topics:

- "Configuring the resource model".
- "Testing your Tivoli integration".
- "Further information".

Configuring the resource model

To configure the resource model for your endpoint, perform the following steps:

1. First, verify that the `configure_provider` script uses the correct location for the `servers.conf` file, and then execute the `configure_provider` task.
2. You must restart the Tivoli Monitoring Engine on the endpoint to pick up this new information. You can do this using the following command:

   ```
   wdmcmd -e endpoint_name -stop
   ```

   Note: On Windows, you might need to allow some time before restarting. This is because it takes time to shut down the Tivoli M12 provider, which hosts the resource model.

3. When this process has finished, it is safe to execute a restart using the following command:

   ```
   wdmcmd -e endpoint_name -restart
   ```

4. View the status of your deployed resource model by opening the Tivoli **Web Health Console**, and view the data for your host. If all the servers in your domain are running, everything should be fine with no errors.
5. Verify that monitoring is working correctly by killing one of your servers. The effect is not immediately visible on the **Web Health Console**. The delay depends on cycle time setting in the profile. The default is `60` seconds. However, the **Web Health Console** can take longer to refresh.

The quickest way to check the status is by executing the following command:

```
wdmlseng -e endpoint_name -verbose
```

This shows the status of any errors in the deployed resource models.

6. You should be able to start the server again using the task library. Go to the task library and execute the `server_command` task. Fill in the name of the server and the action to perform on it (for example, in this case, `mydomain_locator_myhost, start`). You should see your server start and your health return to 100% in the **Web Health Console** soon after that.

Testing your Tivoli integration

When you have checked that you can start and stop servers and monitor their liveness, you can try some of the other available thresholds.

For example, the IONA Tivoli resource model provides a threshold called **NumInvocations upper bound**. This emits an event when the number of operations that a server receives exceeds a certain threshold, which can indicate that an overload is in progress. You can set the threshold, redistributing the profile. You can test this by writing clients to frequently contact the server in question until the threshold is exceeded.

Further information

For full details on how to use the Tivoli Monitoring product, see the *Tivoli Monitoring User Guide*.

CHAPTER 4

Extending to a Production Environment

This section describes how to extend an IONA Tivoli integration from a test environment to a production environment.

In this chapter

This chapter contains the following sections:

Configuring an Artix Production Environment	page 32
Configuring an Orbix Production Environment	page 36

Configuring an Artix Production Environment

Overview

When you have performed the basic setup steps, then you can move on to the deployment-based production tasks. These include:

- "Monitoring your own Artix applications".
- "Monitoring an Artix application on multiple hosts".
- "Monitoring multiple Artix applications on the same host".
- "Further information".

Monitoring your own Artix applications

Using the **Artix Designer** GUI to enable Tivoli to manage your applications is straightforward. **Artix Designer** generates all the correct configuration for you. For details, see "Setting up your Artix Environment" on page 10.

Manual configuration

If you do not use **Artix Designer**, you must add the following settings to your Artix server's configuration file:

```
my_application {

# Ensure that it_response_time_collector is in your orb_plugins list.
orb_plugins = [ ...,"it_response_time_collector"];

# Enable performance logging.
use_performance_logging = true;

# Collector period (in seconds). How often performance information is logged.
plugins:it_response_time_collector:period = "60";

# Set the name of the file which holds the performance log
plugins:it_response_time_collector:filename =
   "/opt/myapplication/log/myapplication_perf.log"

};
```

Note: The specified `plugins:it_response_time_collector:period` should divide evenly into your cycle time (for example, a `period` of `20` and a cycle time of `60`).

Monitoring an Artix application on multiple hosts

The same principles apply when monitoring your server on multiple hosts. Each host has one `servers.conf` file. In the following example, assume that you want to run the `prdserver` on an endpoint host called `dublin`:

1. Create the `servers.conf` and `server_commands.txt` files for the servers that you want to monitor on the `dublin` host. You can write these files manually or use **Artix Designer** (see "Setting up your Artix Environment" on page 10 for details).
2. Run the `configure_provider` task selecting the `dublin` endpoint. Enter the location of the `servers.conf` file on the `dublin` host, shown in Figure 11.
3. Restart the Tivoli Monitoring Engine on `dublin` as described in "Configuring the Resource Model for your Endpoint" on page 28.

Now you should be able to monitor `prdserver` on `dublin`.

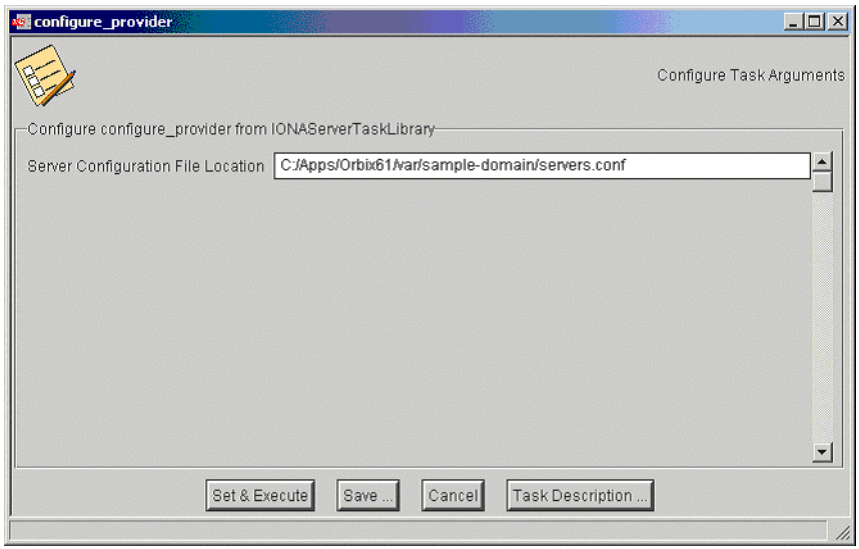

Figure 11: *The configure_provider Task*

Example task Suppose you want to execute the stop script for the `prdserver` on the `dublin` endpoint. Assuming your `server_commands.txt` file is complete, you can open the `server_command` task selecting the `dublin` endpoint. The script takes the following parameters:

```
Server Command File Location, (location of server_commands.txt)
Server Name,
Server Action,
Server Id
```

The `Server Name` and `Server Id` is `myapplication_prdserver`. The action is `stop`, but the `Server Command File Location` defaults to whatever this location was on the host where you first generated the `tivoli_install.tar`.

You must retype this location so that it points to the correct location on the `dublin` host. Use the same path for your `servers.conf` and `server_commands.txt` files on all hosts, if possible. If not, enter a new location each time that you want to invoke an action on a different host.

Alternatively, you can use the `server_command` task as a template for a new task. After changing the value of `server_commands.txt`, and filling in the other fields, instead of clicking **Set & Execute**, click **Save...**. You can rename this task as, for example, `stop_prdserver_on_dublin`.

If you want more flexibility in deciding which parameters to default and which to leave open, you can create a custom task library based on the IONA Tivoli task library. A description of how this is done is beyond the scope of this document. If you need to do this, contact IONA Professional Services.

Monitoring multiple Artix applications on the same host

Sometimes you may need to deploy multiple separate Artix applications on the same host. However, the **Artix Designer** only generates a `servers.conf` and `server_commands.txt` file for a single application.

The solution is to merge the `servers.conf` and `server_commands.txt` files from each of the applications into single `servers.conf` and `server_commands.txt` files.

For example, if the `servers.conf` file from the `UnderwriterCalc` application looks as follows:

```
UnderwriterCalc,1,/opt/myAppUnderwritierCalc/log/UnderwriterCalc_perf.log
```

And the `servers.conf` file for the `ManagePolicy` application looks as follows:

```
ManagePolicy, 1, /opt/ManagePolicyApp/log/ManagePolicy_perf.log
```

The merged `servers.conf` file will then include the following two lines:

```
UnderwriterCalc,1,/opt/myAppUnderwritierCalc/log/UnderwriterCalc_perf.log
ManagePolicy, 1, /opt/ManagePolicyApp/log/ManagePolicy_perf.log
```

Exactly the same procedure applies to the `server_commands.txt` file.

Further information

For full details on how to use the Tivoli Monitoring product, see your *Tivoli Monitoring User Guide*.

Configuring an Orbix Production Environment

Overview

When you have performed the basic setup steps, then you can move on to the deployment-based production tasks. These include:

- "Monitoring your own Orbix applications".
- "Monitoring your Orbix servers on multiple hosts".
- "Monitor multiple Orbix domains on the same host".
- "Further information".

Monitoring your own Orbix applications

You can use the **Orbix Configuration** tool to enable Tivoli management of Orbix services. Enabling Tivoli to manage your own Orbix applications involves the following steps:

1. You must configure your application to use performance logging (see the *Orbix Management User's Guide* for a full description). For example, suppose you have a server executable named `myapplication_prdserver` that executes with the ORB name `myapplication.prdserver`. The typical configuration would be as follows:

C++ applications

```
myapplication {
  prdserver {
     binding:server_binding_list = ["it_response_time_logger+OTS", ""];
     plugins:it_response_time_collector:period = "30";
     plugins:it_response_time_collector:server-id
        ="myapplication_prdserver";
     plugins:it_response_time_collector:filename =
        "/opt/myapplication/logs/prdserver/prdserver_perf.log";
     }
}
```

Java applications

```
myapplication {
  prdserver {
     binding:server_binding_list = ["it_response_time_logger+OTS", "..."];
    plugins:it_response_time_collector:period = "30";
    plugins:it_response_time_collector:server-id = "myapplication_prdserver";
     plugins:it_response_time_collector:log_properties = ["log4j.rootCategory=INFO, A1",
         "log4j.appender.A1=com.iona.management.logging.log4jappender.TimeBasedRollingFile
         Appender",
     "log4j.appender.A1.File=/opt/myapplications/logs/prdserver_perf.log",
     "log4j.appender.A1.layout=org.apache.log4j.PatternLayout",
     "log4j.appender.A1.layout.ConversionPattern=%d{ISO8601} %-80m %n"];
     }
}
```

Note: The specified `plugins:it_response_time_collector:period` should divide evenly into your cycle time (for example, a `period` of `20` and a cycle time of `60`).

2. The most important configuration values are the `server-id` and the C++ `filename` or Java `log_properties` used by the `response_time_collector`. You can add these values to the `servers.conf` file to make the IONA Tivoli resource model aware of your application as follows:

```
myapplication_prdserver, 1,
    /opt/myapplication/logs/prdserver/prdserver_perf.log
```

3. Restart your endpoint. Now Tivoli will monitor the execution of the `myapplication_prdserver`.
4. To control the `myapplication_prdserver` server through the `server_command` task, edit the `server_commands.txt` file. For example you could add the following entries to the `server_commands.txt` file:

```
myapplication_prdserver,start =
    /opt/myapplication/scripts/prdserver_start.sh
myapplication_prdserver,stop =
    /opt/myapplication/scripts/prdserver_stop.sh
myapplication_prdserver,restart =
    /opt/myapplication/scripts/prdserver_restart.sh
```

The `prdserver_start.sh`, `prdserver_stop.sh` and `prdserver_restart.sh` scripts will be written by you.

Monitoring your Orbix servers on multiple hosts

The same principles apply when monitoring your Orbix servers on multiple hosts. Each host has one `servers.conf` file. In the following example, assume that you want to run the `prdserver` on an endpoint host called `dublin`:

1. Write the `servers.conf` and `server_commands.txt` files for the servers that you want to monitor on the `dublin` host (see "Setting up your Orbix Environment" on page 14 for details).
2. Run the `configure_provider` task selecting the `dublin` endpoint. Enter the location of the `servers.conf` file on the `dublin` host, shown in Figure 11.
3. Restart the Monitoring Engine on `dublin` as described in "Configuring the Resource Model for your Endpoint" on page 28.

Now you should be able to monitor `prdserver` on `dublin`.

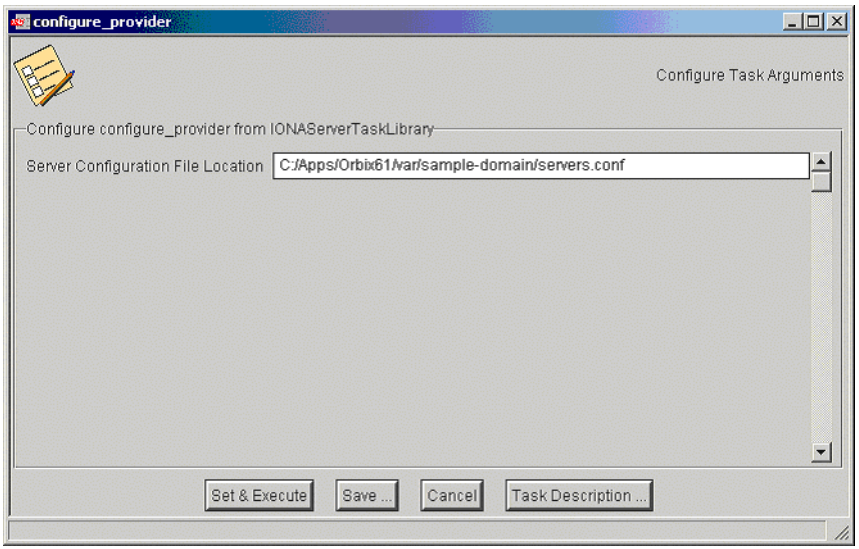

Figure 12: *The configure_provider Task*

Example task: Suppose you want to execute the stop script for the `prdserver` on the `dublin` endpoint. Assuming your `server_commands.txt` file is complete, you can open the `server_command` task selecting the `dublin` endpoint. The script takes the following parameters:

```
Server Command File Location, (location of server_commands.txt)
Server Name,
Server Action,
Server Id
```

The `Server Name` and `Server Id` is `myapplication_prdserver`. The action is `stop`, but the `Server Command File Location` defaults to whatever this location was on the host where you first generated the `tivoli_install.tar`.

You must retype this location so that it points to the correct location on the `dublin` host. Use the same path for your `servers.conf` and `server_commands.txt` files on all hosts, if possible. If not, enter a new location each time that you want to invoke an action on a different host.

Alternatively, you can use the `server_command` task as a template for a new task. After changing the value of `server_commands.txt`, and filling in the other fields, instead of clicking **Set & Execute**, click **Save...**. You can rename this task as, for example, `stop_prdserver_on_dublin`.

If you want more flexibility in deciding which parameters to default and which to leave open, you can create a custom task library based on the IONA Tivoli task library. A description of how this is done is beyond the scope of this document. If you need to do this, contact IONA Professional Services.

Monitor multiple Orbix domains on the same host

You may have more than one Orbix configuration domain running on the same host. Tivoli is not aware of concepts like Orbix configuration domains and the current solution for this is to have the IONA Tivoli resource model perform monitoring of all domains on the same host. This means having only one `servers.conf` or `server_commands.txt` file for each host.

This could potentially cause problems if you have servers on the same host that have the same ORB name and by extension the same default value for the following variable:

`plugins:it_response_time_collector:server-id`

This is why, by default, the server IDs are generated with the domain name added as prefix and the host name added as suffix (for example, `mydomain_locator_myhost`).

A typical `servers.conf` file with two domains (`mydomain` and `yourdomain`) would look as follows:

```
mydomain_locator, 1,
/opt/iona/var/domains/mydomain/logs/locator_myhost_perf.log
...
yourdomain_locator, 1,
/opt/iona/var/domains/yourdomain/logs/locator_yourhost_perf.log
```

Similarly for the task library:

```
mydomain_locator_myhost , start,
/opt/iona/etc/bin/mydomain_locator_start.sh
...
yourdomain_locator_yourhost , start,
/opt/iona/etc/bin/yourdomain_locator_start.sh
```

Further information

For full details on how to use the Tivoli Monitoring product, see your *Tivoli Monitoring User Guide*.

CHAPTER 5

Using the IONA Tivoli Integration

This chapter explains how to perform common tasks using the IONA Tivoli integration. For example, how to access historical data, or detect when a server is down.

In this chapter

This chapter contains the following sections:

Detecting Common Server Problems	page 42
Tracking Server Performance Metrics	page 44
Stopping, Starting, and Restarting Servers	page 45

Detecting Common Server Problems

Overview

This section explains how to detect common server problems using the IONA Tivoli integration. It includes the following:

- "Detecting possible server crashes".
- "Detecting problems with response times".
- "Detecting heavy traffic".
- "Enabling data logging for your servers".
- "Further information".

Detecting possible server crashes

An `Ev_IONAServer_ServerStatus_matches` event is sent when a server listed in `servers.conf` fails to log a `status=running` message since the beginning of the last cycle. The `Ev_IONAServer_ServerStatus_matches` event contains information about the identity of the server that has stopped running.

The cycle time can be set appropriately before you distribute your profile. It is important that the configured value of the `plugins:it_response_time_collector:period` is always less than the cycle time. Otherwise, you may get spurious events of this type. The specified period should divide evenly into your cycle time (for example, a period of `20` and a cycle time of `60`).

For more details on configuration variables, see "Extending to a Production Environment" on page 31.

Detecting problems with response times

If the average response time of a server exceeds the average response time threshold (`Thr_IONAServer_Resource_Model_AvgResponseTime_gt`), an event is emitted to warn the user. A higher than expected response time may indicate a heavy load or possibly a failure that is causing an unexpectedly slow response for users. This threshold should be set appropriately for the servers that you are monitoring. This can be done in a profile or a policy.

There is also a threshold for maximum response times (`Thr_IONAServer_Resource_Model_MaxResponseTime_gt`). The maximum response time refers to the slowest operation that took place on a server during the last collection cycle. Typically, this value can vary a lot more than the average response time, so you might want to set this threshold higher than the average response time.

Detecting heavy traffic

The `NumInvocations` parameter tracks the number of invocations being processed by the server during each cycle. You must treat this metric with caution because it is not normalized and can be prone to sampling errors.

For example, small differences in the actual cycle time could mean that you pick up an extra log entry during the lifetime of a particular cycle. This can lead to a spike in the data.

The effect of this is lessened when the ratio of cycle time/collector period increases. For example, if the performance logging plugin logs data every 60 seconds and the cycle time is 60 seconds, the error could be as much as +/- 100%. If the ratio of cycle time/collector period is 10, the error for this parameter is +/- 10%.

Enabling data logging for your servers

Before you distribute your `IONAProfile`, or indeed any profile based on the IONA Tivoli resource model, it is recommended that you enable logging in the profile as follows:

1. In the Tivoli **Monitoring Profile** window, double click on **IONAServer Monitor** in the top pane.
2. This launches an **Edit Resource Model** window, click on the **Logging…** button in this window.
3. Ensure that the **Enable Data Logging** button is checked.
4. Click **Apply Changes and Close**.
5. Click **Modify & Close** in the **Edit Resource Model** window.

If you do this before distributing the profile, the Tivoli Agent will track and summarize data for all of the attributes in the resource model. You can use these historical logs for a number of tasks (for example, server downtime, explained in the next section).

Further information

For descriptions of all the events, thresholds, and parameters in the IONA Tivoli resource model, see Appendix A.

Tracking Server Performance Metrics

Overview

This section explains how to track key server performance metrics (for example, server downtime and response time). It includes:

- "Examining server downtime".
- "Tracking other server performance metrics".

Examining server downtime

To examine server downtime, perform the following steps:

1. Open the **Web Health Console** and connect to a machine that is running your profile.
2. In the top pane (the one labelled **Resource Models on Hostname**), select the **Historical Data** radio button.
3. In the bottom pane, choose the `IONAServer_Resource_Model` in the left-hand drop-down box, and `IONAServer_Resource_Model_Availability` in the right-hand drop-down box.
4. In the left-hand selection, choose the name of the server that you want to examine.
5. In the right-hand selection, choose `ServerStatus`.

A table is displayed that shows when the server was running, and for what periods (if any) that its status was unknown. This will most likely be because the server was not running.

Tracking other server performance metrics

Follow steps 1-4 listed for "Examining server downtime". But this time, choose a different metric on the right.

For example, to view a history of the average response time of your server, choose `AvgResponseTime (AVG)`. The data is displayed in tabular form for the last hour, by default. However, you can choose to view data for longer periods. The range of graphical presentation options, such as line and bar charts, can give a useful insight into your server usage patterns.

Another metric of interest is `NumOperations`. This tracks the throughput of your server. Viewing the history can help you identify times when the server usage peaks.

Stopping, Starting, and Restarting Servers

Overview

This section explains how to use the `IONAServerTaskLibrary` to perform actions on servers (for example, stop, start, or restart). It includes:

- "Establishing which servers and operations are tracked".
- "Example of starting the locator service".

Establishing which servers and operations are tracked

The `IONAServerTaskLibrary` enables you to stop, start or restart your servers. To check what servers are recognized by the system and what operations are defined for them, perform the following steps:

1. Double click on the `IONAServerTaskLibrary`.
2. Double click on `list_server_commands`.
3. Click the **Display on Desktop** checkbox.
4. Click the endpoint on which to execute the task.
5. Select **Execute & Dismiss**.
6. Verify that the **Server Command File Location** is correct (this is the `server_commands.txt` file).
7. Click **Set & Execute**.

A list of recognized servers and the operations supported for those servers is displayed.

Example of starting the locator service

To start an Orbix locator service (for example, in the domain `foo`, on the host `patrick`) perform the following steps:

1. Double click on the `IONAServerTaskLibrary`.
2. Double click on `server_command`.
3. Click the **Display on Desktop** checkbox. Select the endpoint on which to execute the task.
4. Click **Execute & Dismiss**.

5. Verify that the **Server Commands File Location** is correct (this is for the `server_commands.txt` file)
6. Fill in the name and ID of the server (`foo_locator_patrick`) and the action (`start`).
7. Click **Set & Execute**.

APPENDIX A

IONA Tivoli Resource Model

This appendix describes the contents of the IONA Tivoli resource model. It includes descriptions of the thresholds, events, and parameters used in this model, along with a WBEM/CIM definition.

In this appendix

This chapter contains the following sections:

Thresholds	page 48
Events	page 50
Parameters	page 51
WBEM/CIM Definition	page 52

Thresholds

This section describes the thresholds in the IONA Tivoli resource model. It lists an internal name and description of each threshold.

Thr_IONAServer_Resource_Model_AvgResponseTime_gt

When the `AvgResponseTime` counter exceeds this threshold, the `Ev_IONAServer_Resource_Model_AvgResponseTime_too_high` event is generated.

The default value is `50`.

This threshold corresponds to the **AvgResponseTime upper bound** threshold displayed in the **Profile Manager** GUI.

Thr_IONAServer_Resource_Model_MaxResponseTime_gt

When the `MaxResponseTime` counter exceeds this threshold, the `Ev_IONAServer_Resource_Model_MaxResponseTime_too_high` event is generated.

The default value is `250`.

This threshold corresponds to the **MaxResponseTime upper bound** threshold displayed in the **Profile Manager** GUI.

Thr_IONAServer_Resource_Model_NumInvocations_lt

When the `NumInvocations` counter is lower than this threshold, the `Ev_IONAServer_Resource_Model_NumInvocations_too_low` event is generated.

The default value is `0`.

This threshold corresponds to the **NumInvocations lower bound** threshold displayed in the **Profile Manager** GUI. This threshold is useful for detecting server hangs when used in conjunction with a ping client that is run at regular intervals.

Thr_IONAServer_Resource_Model_NumInvocations_gt

When the `NumInvocations` counter exceeds this threshold the `Ev_IONAServer_Resource_Model_NumInvocations_too_high` event is generated.

The default value is `100000`.

This threshold corresponds to the **NumInvocations upper bound** threshold displayed in the **Profile Manager** GUI.

Events

This section describes the events in the IONA Tivoli resource model. It lists an internal name and description of each event.

Ev_IONAServer_Resource_Model_AvgResponseTime_too_high

This event is generated when the `AvgResponseTime` counter exceeds the **AvgResponseTime upper bound** threshold.

Ev_IONAServer_Resource_Model_MaxResponseTime_too_high

This event is generated when the `MaxResponseTime` counter exceeds the **MaxResponseTime upper bound** threshold.

Ev_IONAServer_Resource_Model_NumInvocations_too_low

This event is generated when the `NumInvocations` counter is lower than the **NumInvocations lower bound** threshold.

Ev_IONAServer_Resource_Model_NumInvocations_too_high

This event is generated when the server is receiving a large number of invocations, and when the `NumInvocations` counter exceeds the **NumInvocations upper bound** threshold. This can be an indication of overload.

Ev_IONAServer_Server_Status_matches

This event is generated when the status of the server is unknown.

Parameters

This section describes the parameter in the IONA Tivoli resource model. It lists an internal name and description.

Par_Problematic_Status_Values_eqs

This specifies values that indicate a problem with the server status. Possible values are as follows:

- `unknown`
- `shutdown_started`
- `shutdown_complete`

WBEM/CIM Definition

WBEM/CIM refers to Web-Based Enterprise Management/Common Information Model. The WBEM/CIM definition for the IONA Tivoli resource model is as follows:

```
#pragma namespace ("\\\\.\\ROOT\\CIMV2")

[
Dynamic,
M12_Instrumentation("Java.com.iona.management.provider.tivoli.AR
   TILTProviderImpl | | ENUM"),
Provider("M12JavaProvider")
]
class IONAServer
{
    [Key, Description("The unique name of an IONA Server
   Replica")]
    string Identifier;

    [
    Dynamic,

   M12_Instrumentation("Java.com.iona.management.provider.tivoli
   .ARTILTProviderImpl | | GET"),
    Provider("M12JavaProvider")
    ]
    uint32 NumInvocations;

    [
    Dynamic,

   M12_Instrumentation("Java.com.iona.management.provider.tivoli
   .ARTILTProviderImpl | | GET"),
    Provider("M12JavaProvider")
    ]
    uint32 MaxResponseTime;
```

```
   [
    Dynamic,

   M12_Instrumentation("Java.com.iona.management.provider.tivoli
   .ARTILTProviderImpl | | GET"),
    Provider("M12JavaProvider")
    ]
    uint32 MinResponseTime;

    [
    Dynamic,

   M12_Instrumentation("Java.com.iona.management.provider.tivoli
   .ARTILTProviderImpl | | GET"),
    Provider("M12JavaProvider")
    ]
    uint32 AvgResponseTime;

    [
    Dynamic,

   M12_Instrumentation("Java.com.iona.management.provider.tivoli
   .ARTILTProviderImpl | | GET"),
    Provider("M12JavaProvider")
    ]
    string ServerStatus;

};
```

Glossary

A

administration

All aspects of installing, configuring, deploying, monitoring, and managing a system.

C

CFR

See configuration repository.

client

An application (process) that typically runs on a desktop and requests services from other applications that often run on different machines (known as server processes). In CORBA, a client is a program that requests services from CORBA objects.

configuration

A specific arrangement of system elements and settings.

configuration domain

Contains all the configuration information that Artix and Orbix applications use. Defines a set of common configuration settings that specify available services and control their behavior. This information consists of configuration variables and their values. Configuration domain data can be implemented and maintained in a centralized Orbix configuration repository, or as a set of files distributed among domain hosts. Configuration domains let you organize configuration into manageable groups, thereby bringing scalability and ease of use to the largest environments. See also configuration file and configuration repository.

configuration file

A file that contains configuration information for Orbix or Artix components within a specific configuration domain. See also configuration domain.

configuration repository

A centralized store of configuration information for all Orbix components within a specific configuration domain. See also configuration domain. This component occurs in Orbix only, and is not present in Artix.

configuration scope

Orbix and Artix configuration is divided into scopes. These are typically organized into a root scope and a hierarchy of nested scopes, the fully-qualified names of which map directly to ORB names. By organizing configuration properties into various scopes, different settings can be provided for individual ORBs, or common settings for groups of ORB. Orbix and Artix services, such as the naming service, have their own configuration scopes.

CORBA

Common Object Request Broker Architecture. An open standard that enables objects to communicate with one another regardless of what programming language they are written in, or what operating system they run on. The CORBA specification is produced and maintained by the OMG. See also OMG.

D

deployment

The process of distributing a configuration or system element into an environment.

E

Enterprise Management System

An Enterprise Management System (EMS) is a set of integrated tools that enable system administrators to manage large-scale production environments. The main Enterprise Management Systems are BMC Patrol™, IBM Tivoli™, HP OpenView™, and CA Unicenter™. These systems give a top-to-bottom view of every part of the IT infrastructure, and enable you to track key server metrics and automate recovery actions if a server crashes.

H

HTTP

HyperText Transfer Protocol. The underlying protocol used by the World Wide Web. It defines how files (text, graphic images, video, and other multimedia files) are formatted and transmitted. Also defines what actions Web servers and browsers should take in response to various commands. HTTP runs on top of TCP/IP.

I

IDL
Interface Definition Language. The CORBA standard declarative language that allows a programmer to define interfaces to CORBA objects. An IDL file defines the public API that CORBA objects expose in a server application. Clients use these interfaces to access server objects across a network. IDL interfaces are independent of operating systems and programming languages.

IIOP
Internet Inter-ORB Protocol. The CORBA standard messaging protocol, defined by the OMG, for communications between ORBs and distributed applications. IIOP is defined as a protocol layer above the transport layer, TCP/IP.

installation
The placement of software on a computer. Installation does not include configuration unless a default configuration is supplied.

Interface Definition Language
See IDL.

invocation
A request issued on an already active software component.

O

OMG
Object Management Group. An open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications, including CORBA. See www.omg.com.

ORB
Object Request Broker. Manages the interaction between clients and servers, using the Internet Inter-ORB Protocol (IIOP). Enables clients to make requests and receive replies from servers in a distributed computer environment. Key component in CORBA.

P

protocol
Format for the layout of messages sent over a network.

S

server

A program that provides services to clients. CORBA servers act as containers for CORBA objects, allowing clients to access those objects using IDL interfaces.

Index

A
Apply Changes and Close 43
Artix Designer 10
Artix workspace 21
average response time 42, 44
AvgResponseTime 44
AvgResponseTime upper bound 48

B
binding:server_binding_list 36, 37
Bundle Details 10

C
C++ configuration 36
check_deployment task 26
configure_provider 28, 33, 38
configure_provider task 26
crash, server 42
create_profile.sh 22
create_tivoli_install_bundle 20, 21
cycle time 28, 37, 42

D
deployment
  bundle 10
  profile 10
  wizard 10
Display on Desktop 45
Domain Settings 15

E
Edit Resource Model 43
EMS 2
Enable Data Logging 25, 43
Enable Management checkbox 10
Enterprise Management System 2
events 50
Ev_IONAServer_Resource_Model_MaxResponseTime_too_high 50
Ev_IONAServer_Resource_Model_NumInvocations_too_high 50
Ev_IONAServer_Resource_Model_NumInvocations_too_low 50
Ev_IONAServer_Server_Status_matches 50
Ev_IONAServer_ServerStatus_matches 42
Execute & Dismiss 45

F
filename 37

G
Generate Code dialog 11
Generate EMS Configuration Files 15

I
import_tll.sh 22
IONAProfile 22, 43
IONAServer Monitor 24, 43
IONAServer_Resource_Model 44
IONAServer_Resource_Model_Availability 44
IONAServerTaskLibrary 26, 45
IONA Tivoli resource model 43
itconfigure tool 15
it_response_time_collector 32
it_response_time_logger 36, 37

J
Java configuration 37

L
Launch Service on Domain Startup 16
list_server_commands task 26
log file interpreter 3
Logging... 25, 43
log_properties 37

M
maximum response time 43
MaxResponseTime upper bound 48
Modify & Close 25, 43
Monitoring Profile 43

N
NumInvocations 43
NumInvocations lower bound 48
NumInvocations upper bound 29, 49, 50
NumOperations 44

O
Orbix Configuration tool 15, 36
orb_plugins 32
Other Properties 16

P
parameters 51
Par_Problematic_Status_Values_eqs 51
performance logging
  configuration 16
  plugins 3
plugins:it_response_time_collector:filename 32, 36
plugins:it_response_time_collector:log_properties 37
plugins:it_response_time_collector:period 32, 36, 37, 42
plugins:it_response_time_collector:server-id 36, 37, 39
Profile Manager 22, 48
pskill 13

R
resource model
  events 50
  parameters 51
  servers.conf 13
  thresholds 48
response time 3
  average 42
  maximum 43
response_time_collector 37

S
Save... 34, 39
server_command 34, 39
Server Command File Location 34, 39, 45
server_commands.txt 13, 17, 26, 34, 35, 39, 45
server_command task 26, 29, 37
server crash 42
server-id 37
servers.conf file 12, 17, 26, 33, 35, 38
ServerStatus 44
Service Launching 16
Set & Execute 34, 39, 45
setup utilities 6
shutdown_complete 51
shutdown_started 51
stopping
  applications on Windows 13

T
taskkill 13
threshold
  average response time 42
  maximum response time 43
thresholds 48
Thr_IONAServer_Resource_Model_AvgResponseTime_gt 42, 48
Thr_IONAServer_Resource_Model_MaxResponseTime_gt 43, 48
Thr_IONAServer_Resource_Model_NumInvocations_gt 49
Thr_IONAServer_Resource_Model_NumInvocations_lt 48
Tivoli Desktop 22
tivoli_install.tar 21, 34, 39
tivoli_integration.tar 20
Tivoli Management Framework 5
Tivoli Monitoring 5
Tivoli profile manager 21
Tivoli region name 21
Tivoli task library 6, 13
Tivoli Web Health Console 28

U
unknown 51
use_performance_logging 32

W
WBEM/CIM definition 52
wdmcmd 28
wdmlseng 29
Web Health console 44
Windows
  stopping applications 13
workspace, Artix 21