PROGRESS®
ORBIX®

CORBA Session Management Guide, Java

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

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The licence and distribution terms for any publically available version or derivative 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 PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE 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 software 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 DAMAGES 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 trademarks and registered trademarks mentioned herein are the property of their respective owners.

Updated: 13-Jul-2011

Contents

Preface

This book describes the Orbix session management capability, which is based on the Orbix leasing plug-in.

Audience

This guide is aimed at developers of Orbix applications. Before reading this guide, you should be familiar with the Object Management Group IDL and the Java language.

Additional resources

The IONA knowledge base (http://www.iona.com/support/knowledge_base/index.xml) contains helpful articles, written by IONA experts, about the Orbix and other products. You can access the knowledge base at the following location:

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

Typographical conventions

This guide 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 guide may use 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

Using the Leasing Plug-In

This chapter describes what the leasing plug-in does and how to use the leasing plug-in on the client-side and the server-side of your application.

In this chapter

The following topics are discussed in this chapter:

The Leasing Framework

Overview

The leasing plug-in is an add-on feature for Orbix that manages server-side resources by detecting when client processes have ceased using a server. This is done using a leasing framework. When a client starts up, it can acquire a *lease* for a particular server, renewing it periodically. When the client terminates, it automatically releases the lease. If the client crashes, the server later detects that the lease has expired. In this manner, both graceful and ungraceful client process terminations are detected.

What is session management?

It is a common requirement in many CORBA systems to know when a client process terminates, in order to clean up resources that are used only by that client. On the server side, session-based applications allocate resources to cater for client requests. To prevent servers from bloating, it is necessary to detect when clients are finished dealing with the server. CORBA does not provide a native solution to this problem.

Features

The leasing framework has the following features:

- Zero impact on client application code.
- Zero impact on existing application IDL interfaces.
- Easy to implement.
- CORBA compliant.
- Completely configurable.

Server side behavior

On the server side, the leasing framework operates as follows:

Stage	Description
1	When a server starts up, it automatically loads the leasing plug-in.
2	During initialization, the server advertises the lease, which causes a `LeaseCallback` object to be bound in the naming service.

Stage	Description
3	Whenever the server exports object references (IORs), the plug-in automatically adds leasing information to the IOR in a CORBA-compliant manner.

Client side behavior

On the client side, the leasing framework operates as follows:

Stage	Description
1	When the client starts up, it automatically loads the leasing plug-in.
2	If the plug-in detects that the client is going to invoke on an object using an IOR containing leasing details, the plug-in automatically initiates a session with the target server by acquiring a lease.
3	The plug-in automatically renews the lease when needed.
4	Upon client shut down: • If the client shuts down gracefully, the plug-in automatically releases the lease with the server. • If the client crashes, the server-side plug-in later realizes that the client has not recently renewed the lease. The lease expires, allowing the server to clean up appropriately.

Lease acquisition

A client initiates a session by acquiring a lease from a leasing server, as shown in Figure 1.

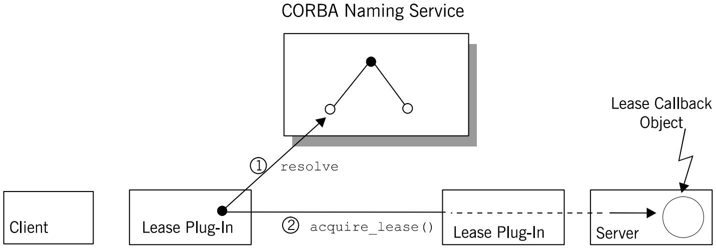

Figure 1: *The Client Acquires a Lease*

The client session is initiated by the leasing plug-in, as follows:

1. The client's leasing plug-in obtains an `IT_Leasing::LeaseCallback` object reference by resolving a name in the CORBA naming service.
2. The client's leasing plug-in initiates a session by calling `acquire_lease()` on the `LeaseCallback` object.

Lease renewal

After acquiring a lease, the client renews the lease at regular intervals, as shown in Figure 2

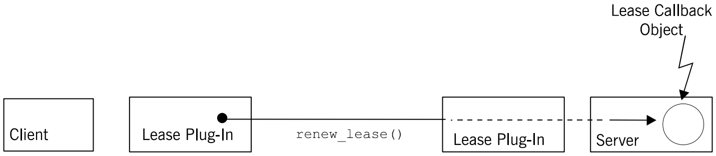

Figure 2: *The Client Renews the Lease*

The period between lease renewals is specified by the `plugins:lease:lease_ping_time` configuration variable.

Client shutdown

When the client shuts down, the lease is released as shown in Figure 3

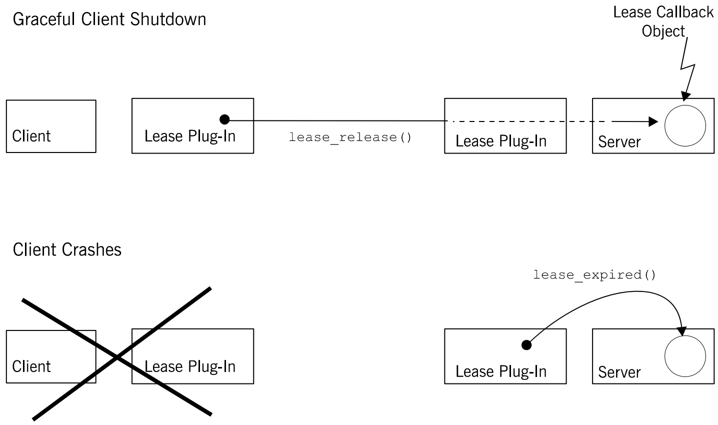

Figure 3: *The Lease is Released When the Client Shuts Down*

The following shutdown scenarios can occur:

- *Graceful client shutdown*—if the client shuts down gracefully, the plug-in automatically calls `lease_release()` to end the session.
- *Client crashes*—if the client crashes, the server-side plug-in calls `lease_expired()` on the `LeaseCallback` object after a period of time specified by the `plugins:lease:lease_reap_time` configuration variable.

A Sample Leasing Application

Location

Source code and build instructions for a sample leasing application are located in the asp/6.2/demos/corba/standard/session_management directory of your Orbix installation.

The LeaseTest IDL module

The sample leasing application is based on a server that supports a simple factory pattern for creating transient Person objects:

```
//IDL
module LeaseTest {
    exception PersonAlreadyExists { };

    interface Person {
        string name();
    };

    interface PersonFactory {
        Person create_person(in string name)
            raises (PersonAlreadyExists);
    };
};
```

Purpose

The purpose of this example is to show that no matter how many clients create Person objects, and no matter how those client processes terminate, the server is notified when it can safely clean up the objects. Therefore, the server is able to keep its memory usage down.

Client-server interaction

Clients interact with the LeaseTest server as follows:

Stage	Description
1	A client creates new Person objects by calling the create_person() operation, with unique name arguments for each Person.
2	When a client terminates, the Person objects it created no longer need to be held inside the server memory and are deleted.

Using the Leasing Plug-In on the Client Side

Prerequisites

The client plug-in makes periodic `resolve()` calls to the Naming Service during its lifetime. Therefore, your Orbix domain should have a properly configured locator, activator, and naming service ready before running a leasing client.

How to use the plug-in

The only thing that needs to be changed in a client deployment that uses the leasing framework is its configuration. Specifically, the plug-in must be added to the list of ORB plug-ins and be configured to participate in bindings.

Configuration variables

The following basic configuration variables are needed to configure and activate the client-side plug-in:

Table 1: *Configuration Variables Used on the Client Side*

Configuration Variable	Purpose
`plugins:lease:ClassName`	Identifies the lease plug-in class name.
`orb_plugins`	The ORB plug-in list is modified to ensure that the lease plug-in is automatically loaded when the client ORB is initialized.
`binding:client_binding_list`	The client binding list is modified to ensure that the plug-in can participate in request processing.

The complete set of leasing plug-in configuration variables is given in "Leasing Plug-In Configuration Variables" on page 23.

Configuring for colocated CORBA objects

In the `client_binding_list`, a binding description containing the `POA_Coloc` interceptor name *must* appear before the first binding description that contains a `LEASE` interceptor name. This is to ensure that a leasing application does not attempt to lease a colocated CORBA object.

Example configuration

In an Orbix file-based configuration, the client-side plug-in might be configured as follows:

```
# Orbix Configuration File
plugins:lease:ClassName =
   "com.iona.corba.plugin.lease.LeasePlugIn";
orb_plugins = ["local_log_stream", "lease", "iiop_profile",
   "giop", "iiop"];
binding:client_binding_list = ["POA_Coloc", "LEASE+GIOP+IIOP",
   "GIOP+IIOP"];
```

Using the Leasing Plug-In on the Server Side

The IT_Leasing module

Servers wishing to act as leasing servers interact with the plug-in to advertise leases. The interfaces used by leasing servers are declared in the `IT_Leasing` module, which is defined in the `leasing.idl` file:

```
//IDL
module IT_Leasing
{
    ...
    interface LeaseCallback
    {
        LeaseID acquire_lease()
        raises (CouldNotAcquireLease);
        void lease_expired(in LeaseID lease_id);
        void lease_released(in LeaseID lease_id);
        void renew_lease(in LeaseID  lease_id)
            raises (LeaseHasExpired);
    };
    local interface ServerLeaseAgent
    {
        void advertise_lease(
            in LeaseCallback lease_callback
        ) raises (CouldNotAdvertiseLease);
        LeaseID manufacture_lease_id();
        void withdraw_lease();
        void lease_acquired(in LeaseID lease_id);
        void lease_released(in LeaseID lease_id);
    };
    local interface Current : CORBA::Current
    {
        exception NoContext {};
        LeaseID get_lease_id() raises (NoContext);
    };
    ...
};
```

The complete listing for the `IT_Leasing` module is in “Leasing IDL Interfaces” on page 31.

The LeaseCallback interface

Your server must provide an implementation of the `IT_Leasing::LeaseCallback` interface to receive notifications of lease-related events from the leasing plug-in. For example, when leases expire, the plug-in calls `IT_Leasing::LeaseCallback::lease_expired()`.

The server lease agent interface

The implementation of the `ServerLeaseAgent` interface is provided by the leasing plug-in. Your server communicates with the leasing plug-in by calling the operations defined on this interface. For example, the server can initialize the leasing plug-in by calling `IT_Leasing::ServerLeaseAgent::advertise_lease()`.

The Current interface

For a leasing server to react correctly to the *ending* of a lease, it must know which resources are relevant to that lease. In other words, the server must maintain an association between the resources that it has created and the clients that are currently using them.

This problem is solved as follows. When your server needs to figure out which leasing client invoked a particular operation, you can extract lease information from an object of `IT_Leasing::Current` type, which is derived from `CORBA::Current`, an interface specifically used for retrieving meta-information about CORBA invocations. Once the `IT_Leasing::Current` object is obtained, you can call `get_lease_id()` on it to find the lease ID relevant to that call.

If the call is made from a non-leasing client (or a non-Orbix client), the `IT_Leasing::Current::NoContext` user exception is thrown.

Implementing the server

To use the plug-in on the server side, perform the following steps:

Step	Action
1	Implement the LeaseCallback Interface.
2	Use IT_Leasing::Current to Track Client Sessions.
3	Advertise the Lease.
4	Configure the Server-Side Plug-In.

Implement the LeaseCallback Interface

Overview

You must implement the `LeaseCallback` interface to receive notification of leasing events from the plug-in.

The following example shows a code extract from the `LeaseTest` demonstration, where the `LeaseCallback` interface is implemented by the `LeaseCallbackImpl` class.

Object instances

The following two object instances are used by the `LeaseCallbackImpl` class:

Table 2: *Object Instances Used in the LeaseCallbackImpl Class*

Object Instance	Description
`m_lease_obj`	An `IT_Leasing::ServerLeaseAgent` object reference. This object is used to communicate with the leasing plug-in.
`m_factory`	A reference to a `PersonFactoryImpl` object. This object is used to create new instances of `Person` CORBA objects.

Implementation code

The `IT_Leasing::LeaseCallback` interface is implemented by the `LeaseCallbackImpl` Java class, as shown in Example 1.

Example 1: *The LeaseCallbackImpl Class (Sheet 1 of 2)*

```
//Java
package demos.session_management.LeaseTest;
//--JDK Imports--
import java.io.*;
//--IONAImports--
import demos.session_management.LeaseTest.*;
import com.iona.corba.IT_Lease_Component.*;
import com.iona.corba.IT_Lease_Logging.*;
import com.iona.corba.IT_Leasing.*;
import com.iona.corba.plugin.*;
import com.iona.corba.util.SystemExceptionDisplayHelper;
```

Example 1: *The LeaseCallbackImpl Class (Sheet 2 of 2)*

```
    class LeaseCallbackImpl extends LeaseCallbackPOA
    {
      private PersonFactoryImpl m_factory = null;
      private ServerLeaseAgent m_lease_obj = null;

      // Constructor (not shown)
      ...
      // IDL operations
1     public String acquire_lease()
      {
        // We could throw CouldNotAcquireLease here if we
        // wanted to refuse the lease
        if (m_lease_obj == null)
        {
          System.err.println(
    "ERROR: The Lease callback object has not been set correctly.");
          System.exit(1);
        }
        String new_lease = m_lease_obj.manufacture_lease_id();
        m_lease_obj.lease_acquired(new_lease);
        return new_lease;
      }

2     public void lease_expired(String lease_id)
      {
        m_factory.owner_has_gone_away(lease_id);
      }

3     public void lease_released(String lease_id)
      {
        m_lease_obj.lease_released(lease_id);
        m_factory.owner_has_gone_away(lease_id);
      }

4     public void renew_lease(String lease_id)
      {
        // Nothing to do, since the plugin has already intercepted
        // this request and knows that the lease has been renewed.
      }
    }
```

The code can be explained as follows:

1. The `LeaseCallbackImpl.acquire_lease()` method is called by client lease plug-ins when they need to acquire a lease with your server. The sample implementation asks the lease plug-in for a new unique lease ID, and then informs the plug-in that it has accepted the lease acquisition request by calling `lease_acquired()` on the `ServerLeaseAgent` object. You could also create the lease ID yourself—however, you are then required to ensure its uniqueness within the server process.
2. The `LeaseCallbackImpl.lease_expired()` method is called by the plug-in when a particular lease has expired—that is, if the lease has not been renewed within the configured reap time (see "Leasing Plug-In Configuration Variables" on page 23). This can occur if the client crashes or if the network link is lost between the client and the server.

   The sample implementation informs the `Person` factory that a particular owner of `Person` objects has disappeared, by calling `owner_has_gone_away()`. The `Person` factory is then free to remove any `Person` objects belonging to that client. The sample `PersonFactory` removes the `Person` objects from a hash table, which allows the garbage collector to free the associated memory. Alternatively, a server could *evict* the transient objects by persisting their data before removing them from the hash table.
3. The `LeaseCallbackImpl.lease_released()` method is called by client lease plug-ins when the client shuts down gracefully. The implementation of this method is typically almost identical to the implementation of `lease_expired()`, because they are both caused by client terminations. The sample code delegates to the `PersonFactory` servant, informing it that a particular client has shut down.

   There is one important difference between `lease_released()` and `lease_expired()`, however. When `lease_released()` is invoked, you should inform the plug-in of the event, so that it stops managing that particular lease and checking for its expiration. Do this by calling `ServerLeaseAgent::lease_released()`, as in the example code.
4. The `LeaseCallbackImpl.renew_lease()` method is the ping method that the client plug-ins call periodically to renew their leases. You can leave this function body empty. By virtue of the call reaching this point, it has

already been intercepted and examined by the server-side plug-in. During the interception, the lease is timestamped with the current time as its *last renewed time*. You might want to perform some logging here.

Use IT_Leasing::Current to Track Client Sessions

Overview

The server has to track the resources associated with each client and this is done with the help of the `IT_Leasing::Current` interface. In the `LeaseTest` example, the associated resources are `Person` objects. Whenever a `Person` object is created (using the `LeaseTest::PersonFactory` interface) the server associates the new `Person` object with the current client session.

The current client session is identified by the current lease ID, which is obtained from the `IT_Leasing::Current` interface.

Implementation code

The `LeaseTest::PersonFactory` interface is implemented by the `PersonFactoryImpl` Java class as shown in Example 2.

Example 2: *The PersonFactoryImpl Class (Sheet 1 of 5)*

```
//Java
package demos.session_management.LeaseTest;
//--JDK Imports--
import java.io.*;
import java.util.*;
//--OMG Imports--
import org.omg.CORBA.*;
import org.omg.CORBA.ORBPackage.*;
import org.omg.PortableServer.*;
import org.omg.PortableServer.POAPackage.*;
//--IONAImports--
import com.iona.corba.util.SystemExceptionDisplayHelper;
import com.iona.corba.IT_Leasing.*;
import com.iona.corba.IT_Leasing.CurrentPackage.*;
class PersonFactoryImpl extends PersonFactoryPOA
{
  // The set of People that the Factory is currently managing
  private Hashtable m_people = new Hashtable();
  private ORB m_orb;
  private POA m_poa;

  // Constructor
  ... // (not shown)
  public Person create_person(String name)
    throws PersonAlreadyExists
  {
    Person result = null;
    try
    {
      System.out.println("LeaseTest.create_person("+name+")");
1     String owner = "<unknown>";

      try
      {
2       org.omg.CORBA.Object objref =
          m_orb.resolve_initial_references("LeaseCurrent");
```

Example 2: *The PersonFactoryImpl Class (Sheet 2 of 5)*

```
      if (objref != null)
      {
        com.iona.corba.IT_Leasing.Current current
          = com.iona.corba.IT_Leasing.CurrentHelper.narrow(
                objref
            );
3       owner = current.get_lease_id();
      }
    }
    catch (NoContext nc)
    {
      System.err.println(
          "Couldn't find the relevant ServiceContext data.");
    }
    catch (InvalidName in)
    {
      System.err.println("Caught InvalidName exception.");
    }
    catch (SystemException se)
    {
      System.err.println("Unknown exception"
           + SystemExceptionDisplayHelper.toString(se));
    }

    // Create a new Person servant and activate it
    PersonImpl newPersonServant;
    byte[] oid;
    org.omg.CORBA.Object tmp_ref = null;

    synchronized (this)
    {
      // check for Person existence within this process
      if (person_is_alive(name))
      {
        System.err.println("Person already exists!");
        throw new PersonAlreadyExists();
      }
      else
      {
        // Person does not exist, so it is created and
        // stored with the others, indexed by its name
4       newPersonServant = new PersonImpl(name, owner);
```

Example 2: *The PersonFactoryImpl Class (Sheet 3 of 5)*

```
          try
          {
            oid = m_poa.activate_object(newPersonServant);
            tmp_ref = m_poa.id_to_reference(oid);
          }
          catch (ServantAlreadyActive sae)
          {
            System.err.println(
                "Unexpected ServantAlreadyActive exception.");
          }
          catch (WrongPolicy wp)
          {
            System.err.println(
                "Unexpected WrongPolicy exception.");
          }
          catch (ObjectNotActive one)
          {
            System.err.println(
                "Unexpected ObjectNotActive exception.");
          }

          result = PersonHelper.narrow(tmp_ref);

          if (result == null)
          {
            System.err.println("Person is null error");
            System.exit(1);
          }

          // store the new servant with the others
          String temp_string = new String(name);
5         m_people.put(temp_string, newPersonServant);
          System.out.println("Created: " + name);
          dump_people_to_screen();
        }
      }
    }
    catch (PersonAlreadyExists pae)
    {
      throw pae;
    }
```

Example 2: *The PersonFactoryImpl Class (Sheet 4 of 5)*

```
      catch (SystemException se)
      {
        System.err.println("Unexpected system exception." +
     SystemExceptionDisplayHelper.toString(se));
      }

6     return result;
    }

7   void owner_has_gone_away(String owner)
    {
      // Iterate through the people map and evict any people
      // who were created by 'owner'.
      //
      Hashtable tmp_table = new Hashtable();
      tmp_table.putAll(m_people);

      Set the_set = tmp_table.keySet();
      String this_owner = null;

      if (!the_set.isEmpty())
      {
        Iterator the_iter = the_set.iterator();
        do
        {
          String key = (String)the_iter.next();
          PersonImpl the_person = (PersonImpl)tmp_table.get(key);
          this_owner = the_person.owner();

          // value may == null if this has already been evicted
          // while we are iterating through the list.
          if (owner.equals(this_owner))
          {
            try
            {
              // deactivate the servant before deleting it
              byte[] oid = m_poa.servant_to_id(the_person);
              // deactivate the servant with the corresponding
              // id on the POA
8             m_poa.deactivate_object(oid);
            }
```

Example 2: *The PersonFactoryImpl Class (Sheet 5 of 5)*

```
          catch(ObjectNotActive one)
          {
            System.err.println(
                "ERROR: Unexpected ObjectNotActive exception.");
          }
          catch(WrongPolicy wp)
          {
            System.err.println(
                "ERROR: Unexpected WrongPolicy exception.");
          }
          catch(ServantNotActive sna)
          {
            System.err.println(
               "ERROR: Unexpected ServantNotActive exception.");
          }
9         m_people.remove(key);
        }
      }
      while(the_iter.hasNext());
    }
    dump_people_to_screen();
  }
  ...
}
```

The code can be explained as follows:

1. If the factory cannot figure out the relevant lease ID, it assigns a default ID of `<unknown>` as the owner of the object. This happens if a non-leasing client (either a non-Orbix client or an Orbix client that did not load the plug-in) invokes the factory.
2. The factory checks to see if it can contact the `LeaseCurrent` object.
3. If a reference to a `LeaseCurrent` object can be obtained, the `get_lease_id()` method is called to get the lease ID (of string type) for this invocation.
4. A new `Person` object is created and activated. The `result` variable is set equal to the corresponding `Person` object reference.
5. The factory stores the new `Person` object in its own internal table of `Person` objects, `m_people`, using the lease ID, `temp_string`, as a key.
6. The `Person` object reference, `result`, is returned to the calling code.

7. The `owner_has_gone_away()` method is called by `LeaseCallback::lease_expired()` or `LeaseCallback::lease_released()` to clean up the resources (`Person` objects) associated with a client session identified by the `owner` string. The code iterates over all of the entries in the `m_person` table, searching for entries associated with the `owner` session.
8. Before removing a `Person` object from the hash table, the corresponding servant must be deactivated by calling `PortableServer.POA.deactivate_object()`.
9. The servant object is removed from the `m_people` hash table in this line of code. This allows the Java garbage collector to free the associated memory.

Advertise the Lease

Prerequisites

Advertising the lease causes the `LeaseCallback` object reference to be bound into the naming service. Therefore, you must have your Orbix locator, node daemon, and naming service properly configured and ready to run.

Where to advertise

Lease advertisement is an initialization step that is performed in the server `main()` method. This should be done before the server starts to process incoming CORBA requests (that is, before the server calls `ORB.run()` or `ORB.perform_work()`).

Implementation code

The code shown in Example 3 should be added to your server's `main()` method to advertise the lease:

Example 3: *Advertising the Lease in the main() Method (Sheet 1 of 2)*

```
//Java
package demos.session_management.LeaseTest;
// Imports (not shown)
...
class Server
{
  ...
  public static void main(String args[])
  {
    ...

    ServerLeaseAgent leaseObj = null;
    ...
    // Contact the Lease Plugin
    try
    {
1     tmp_ref = orb.resolve_initial_references(
                     "IT_ServerLeaseAgent"
                );
      leaseObj = ServerLeaseAgentHelper.narrow(tmp_ref);
    }
    catch (InvalidName in)
    {
        // Process the exception ...
    }
```

Example 3: *Advertising the Lease in the main() Method (Sheet 2 of 2)*

```
    catch (SystemException se)
    {
        // Process the exception ...
    }
    ...

    // Assume that we have already created and activated a
    // LeaseCallback servant and created a reference for it
    // called the_LeaseCallbackObject.
    ...
    // advertise a lease on the lease plugin
    try
    {
2     leaseObj.advertise_lease(the_LeaseCallbackObject);
    }
    catch (CouldNotAdvertiseLease cna)
    {
        // Process the exception ...
    }
    catch (DuplicateServerID dsid)
    {
        // Process the exception ...
    }
    catch (SystemException se)
    {
        // Process the exception ...
    }
    ...
  }
}
```

The code can be explained as follows:

1. The server obtains an initial reference to a `ServerLeaseAgent` object, which is created by the leasing plug-in.
2. The leasing plug-in is initialized by calling `advertise_lease()` on the `ServerLeaseAgent` object. The `advertise_lease()` operation takes a single parameter, `the_LeaseCallbackObject`, which causes the `LeaseCallback` object to be registered with the plug-in.

Configure the Server-Side Plug-In

Overview

Server-side configuration variables are used to initialize the server-side plug-in and to customize the behavior of the leasing plug-in. Some of these configuration variables are communicated to clients by inserting the information into IORs generated by the server.

Configuration variables

In addition to the client-side configuration variables, the following basic configuration variables are needed to configure the server-side plug-in:

Table 3: *Configuration Variables Used on the Client Side*

Configuration Variable	Purpose
`binding:server_binding_list`	The server binding list is modified, instructing the ORB to insert `LEASE` interceptors into server-side bindings.

`plugins:lease:`
`lease_name_to_advertise`
	The name under which the `LeaseCallback` object is bound in the naming service. This name must be unique per server.

`plugins:lease:lease_ping_time`	The time interval (in milliseconds) between successive ping messages sent by client-side plug-ins to renew the lease.
`plugins:lease:lease_reap_time`	If a particular client's lease is not pinged within `lease_reap_time`, the server resources associated with the client are released.

The complete set of leasing plug-in configuration variables is given in "Leasing Plug-In Configuration Variables" on page 23.

Example configuration

For a complete example of a client-side and server-side configuration, see "Sample Leasing Plug-In Configuration" on page 27.

APPENDIX A

Leasing Plug-In Configuration Variables

The following list describes the leasing plug-in configuration variables and their allowed values, ranges, and defaults.

In this appendix

This appendix contains the following sections:

Common Variables

List of variables

The following configuration variables apply to both clients and servers:

event_log:filters Specifies a list of logging filters. You can configure the plug-in to write to a log stream by appending the plug-in log stream to the list of filters (see the *CORBA Administrator's Guide* for more information on log stream configuration). The plug-in's log stream object is `IT_LEASE`. For example, to get full diagnostic output from the plug-in, set the variable `event_log:filters` equal to `["IT_LEASE=*"]`.

plugins:lease:lease_ns_context Identifies the naming service `NamingContext` where the leasing plug-in registers the `LeaseCallback` object. The name should be a valid `NamingContext` id (see the CORBA Naming Service specification). Since both leasing clients and leasing servers use this value, it should be set to the same value across your entire domain. The default is `IT_Leases`.

plugins:lease:shlib_name Identifies the base name of the leasing plug-in shared library. The `shlib_name` variable should be set to `it_lease`.

Server-Side Variables

List of Variables

The following configuration variables apply only to servers:

plugins:lease:allow_advertisement_overwrites Determines whether the server can re-advertise the same lease when it comes back up after a crash or disorderly shutdown. Internally, the plug-in uses `NamingContext::rebind()` if set to `true`, or `NamingContext::bind()` if set to `false`, when binding the `LeaseCallback` object in the naming service.

The default is `false`, but in a real deployment scenario the recommended setting is `true`.

plugins:lease:lease_name_to_advertise Determines the lease name used when registering the `LeaseCallback` object in the naming service. This name should be configured to be unique among all your leasing servers. The name should be a valid `NamingContext` id (see the CORBA naming service specification). The default value is `default_lease_name`.

plugins:lease:lease_ping_time Determines the value inserted into `TAG_IONA_LEASE` IOR components for the lease ping time. Leasing clients using that IOR automatically renew the lease by pinging every `N` ms, where `N` is the value specified in this variable. The default value is 900,000 ms (15 minutes). Legal values are unsigned longs > 1. In addition, if the ping time is specified to be greater than the reap time, `lease_reap_time`, it is automatically changed to half the reap time.

plugins:lease:lease_reap_time Determines how often the server-side plug-in checks whether leases have expired. The value is specified in ms. If a particular lease has not been renewed (pinged) by its client in this amount of time, the lease expires. Legal values are unsigned longs > 2. The default value is 1,800,000 ms (30 minutes).

APPENDIX B

Sample Leasing Plug-In Configuration

This appendix shows the leasing plug-in configuration used in the session management demonstration.

Configuration file extract

The following listing is a sample valid configuration for a set of applications, `Server1`, `Server2`, and clients, using the leasing plug-in. This configuration is included in generated Orbix domains, *OrbixInstallDir*`/etc/domains/`*domain_name*`.cfg`, where *domain_name* is the name of your domain.

Example 4: *Configuration File Extract for Leasing Plug-In*

```
# Orbix Configuration File
...
demos {
    ...
    session_management
    {
        plugins:lease:shlib_name = "it_lease";
        plugins:lease:ClassName =
                      "com.iona.corba.plugin.lease.LeasePlugIn";
        orb_plugins = ["local_log_stream", "lease",
                       "iiop_profile", "giop", "iiop"];
        binding:client_binding_list = ["POA_Coloc",
                                       "LEASE+GIOP+IIOP",
                                       "GIOP+IIOP"];
        binding:server_binding_list = ["LEASE", ""];
        plugins:lease:allow_advertisement_overwrites = "true";
        # default is false
        event_log:filters = ["IT_LEASE=*"];
        server1 {
            # client must ping every 10 seconds
            plugins:lease:lease_ping_time = "10000";
            # leases will expire after 20 seconds of inactivity
            plugins:lease:lease_reap_time = "20000";
            plugins:lease:lease_name_to_advertise
                                          = "PersonFactorySrv1";
        };
        server2 {
            # client must ping every 20 seconds
            plugins:lease:lease_ping_time = "20000";
            # leases will expire after 40 seconds of inactivity
            plugins:lease:lease_reap_time = "40000";
            plugins:lease:lease_name_to_advertise
                                          = "PersonFactorySrv2";
        };
    };
    ...
};
```

APPENDIX C

Leasing IDL Interfaces

The complete IDL for the leasing plug-in.

The IT_Leasing IDL module

The IT_Leasing module is defined as follows:

Example 5: *The IT_Leasing IDL Module (Sheet 1 of 3)*

```
//IDL
#ifndef __IT_LEASING_IDL_
#define __IT_LEASING_IDL_
//
// @Copyright (c) 2000 IONA Technologies PLC. All Rights
// Reserved.

#include <omg/orb.idl>
#include <omg/IOP.idl>
#pragma prefix "iona.com"
module IT_Leasing
{
    // Type definitions
    typedef string LeaseID;

    // Possible error conditions
    exception LeaseHasExpired {};
```

Example 5: *The IT_Leasing IDL Module (Sheet 2 of 3)*

```
    enum LeaseAdvertisementError {
        NAMING_SERVICE_UNREACHABLE,
        LEASE_ALREADY_ADVERTISED,
        LEASE_ALREADY_BOUND_IN_NS,
        UNKNOWN_ERROR
    };
    exception CouldNotAdvertiseLease
    {
        LeaseAdvertisementError reason;
    };
    exception CouldNotAcquireLease {};

    // This is the maximum amount of time that a client leasing
    // plugin will wait before automatically renewing a
    // particular lease. The value is set in the server plugins'
    // configuration.
    typedef unsigned long IdleTimeBeforePing; // milliseconds
    // This interface must be implemented by servers that
    // wish to advertise leases.
    interface LeaseCallback
    {
        // Informs the server that a client wants a new lease.
        LeaseID acquire_lease()
        raises (CouldNotAcquireLease);
        // Informs the server that a lease not been renewed
        // (usually because the client has gone away)
        void lease_expired(
            in LeaseID  lease_id
        );
        // Informs the server that a client has explicitly
        // released a lease
        void lease_released(in LeaseID  lease_id);

        // renew_lease() is called by leasing plugins on the
        // client side to renew leases after some idle time.
        // This is semantically equivalent to a 'keepalive'
        // or 'heartbeat' method.
        void renew_lease(in LeaseID  lease_id)
        raises (LeaseHasExpired);
    };
    // This is the interface that leasing plugins will
    // expose on the server side. Server programmers must
    // interact with this interface to advertise leases.
    local interface ServerLeaseAgent
    {
```

Example 5: *The IT_Leasing IDL Module (Sheet 3 of 3)*

```
        // advertise_lease() is called by the server
        // to start the lease advertisement. The ping time
        // and ServerID values for the lease are obtained
        // from configuration.
        void advertise_lease(in LeaseCallback lease_callback)
        raises (CouldNotAdvertiseLease);
        // Helper function that generates a system defined
        // lease id, in case the server does not need to attach
        // any specific meaning to incoming leases.
        LeaseID manufacture_lease_id();
        // You may call this method at any time to withdraw your
        // lease, but note that the plugin will automatically
        // withdraw your lease at ORB shutdown time, so you
        // typically never need to call this method.
        void withdraw_lease();
        // Call this method if you wish the plugin to
        // detect that a particular lease has expired (usually
        // due to non-graceful client termination).
        // The typical place to call this is from your
        // implementation of LeaseCallback::acquire_lease().
        void lease_acquired(in LeaseID lease_id);
        // Call this method when you wish the plugin to stop
        // detecting that a particular lease has expired, usually
        // because a client has terminated gracefully and
        // released the lease themselves.
        // The typical place to call this is from your
        // implementation of LeaseCallback::lease_released().
        void lease_released(in LeaseID lease_id);
    };
    // This interface represents the lease details that will
    // be added to requests by leasing clients. The information
    // will be added as a ServiceContext and be available within
    // the servant implementations through the Current interface.
    local interface Current :
    CORBA::Current
    {
        exception NoContext {};
        LeaseID get_lease_id()
        raises (NoContext);
    };
    const IOP::ServiceId SERVICE_ID = 0x49545F43;
};
#endif /*__IT_LEASING_IDL_*/
```

Glossary

A

activator

A server host facility that is used to activate server processes.

ART

Adaptive Runtime Technology. IONA's modular, distributed object architecture, which supports dynamic deployment and configuration of services and application code. ART provides the foundation for IONA software products.

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 Orbix ORBs, services and applications use. Defines a set of common configuration settings that specify available services and control ORB behavior. This information consists of configuration variables and their values. Configuration domain data can be implemented and maintained in a centralised Orbix configuration repository or as a set of files distributed among domain hosts. Configuration domains let you organise ORBs 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 components within a specific configuration domain. See also configuration domain.

configuration repository

A centralised store of configuration information for all Orbix components within a specific configuration domain. See also configuration domain.

configuration scope

Orbix 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 organising configuration properties into various scopes, different settings can be provided for individual ORBs, or common settings for groups of ORB. Orbix 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.

CORBA naming service

An implementation of the OMG Naming Service Specification. Describes how applications can map object references to names. Servers can register object references by name with a naming service repository, and can advertise those names to clients. Clients, in turn, can resolve the desired objects in the naming service by supplying the appropriate name. The Orbix naming service is an example.

CORBA objects

Self-contained software entities that consist of both data and the procedures to manipulate that data. Can be implemented in any programming language that CORBA supports, such as C++ and Java.

D

deployment

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

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.

implementation repository

A database of available servers, it dynamically maps persistent objects to their server's actual address. Keeps track of the servers available in a system and the hosts they run on. Also provides a central forwarding point for client requests. See also location domain and locator daemon.

interceptor

An implementation of an interface that the ORB uses to process requests. Abstract request handlers that can implement transport protocols (such as IIOP), or manipulate requests on behalf of a service (for example, adding transaction identity).

Interface Definition Language

See IDL.

invocation

A request issued on an already active software component.

IOR

Interoperable Object Reference. See object reference.

L

location domain

A collection of servers under the control of a single locator daemon. Can span any number of hosts across a network, and can be dynamically extended with new hosts. See also locator daemon and node daemon.

locator daemon

A server host facility that manages an implementation repository and acts as a control center for a location domain. Orbix clients use the locator daemon, often in conjunction with a naming service, to locate the objects they seek. Together with the implementation repository, it also stores server process data for activating servers and objects. When a client invokes on an object, the client ORB sends this invocation to the locator daemon, and the locator daemon searches the implementation repository for the address of the server object. In addition, enables servers to be moved from one host to another without disrupting client request processing. Redirects requests to the new location and transparently reconnects clients to the new server instance. See also location domain, node daemon, and implementation repository.

N

naming service

See CORBA naming service.

node daemon

Starts, monitors, and manages servers on a host machine. Every machine that runs a server must run a node daemon.

O

object reference

Uniquely identifies a local or remote object instance. Can be stored in a CORBA naming service, in a file or in a URL. The contact details that a client application uses to communicate with a CORBA object. Also known as interoperable object reference (IOR) or proxy.

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.

POA

Portable Object Adapter. Maps object references to their concrete implementations in a server. Creates and manages object references to all objects used by an application, manages object state, and provides the infrastructure to support persistent objects and the portability of object implementations between different ORB products. Can be transient or persistent.

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