

Oracle9iAS TopLink

Troubleshooting Guide

Release 2 (9.0.3)

August 2002

Part No. B10068-01

Copyright © 2002, Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle*MetaLink*, Oracle Store, Oracle9i, Oracle9iAS Discoverer, SQL*Plus, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xix
Preface	xxi
Intended Audience	xxi
Documentation Accessibility	xxii
Structure	xxii
Related Documents.....	xxiii
Conventions.....	xxiv
1 Runtime and Development Exceptions	
TopLinkExceptions	1-1
Runtime Exceptions	1-1
Development Exceptions.....	1-1
Format of Exceptions	1-2
BuilderException	1-2
Format	1-2
CommunicationException	1-3
ConcurrencyException	1-3
Format	1-3
ConversionException	1-3
Format	1-4
DatabaseException	1-4
Format	1-4
DescriptorException	1-4

Format	1-5
OptimisticLockException	1-5
Format	1-5
QueryException	1-5
Format	1-6
TopLinkException	1-6
ValidationException	1-6
Format	1-6

2 Error Codes and Descriptions

Descriptor Exceptions (1 – 176)	2-1
Error code: 1	2-1
Error code: 2	2-1
Error code: 6	2-2
Error code: 7	2-2
Error code: 8	2-2
Error code: 9	2-2
Error code: 10	2-3
Error code: 11	2-3
Error code: 12	2-3
Error code: 13	2-3
Error code: 14	2-4
Error code: 15	2-4
Error code: 16	2-4
Error code: 17	2-4
Error code: 18	2-5
Error code: 19	2-5
Error code: 20	2-5
Error code: 21	2-5
Error code: 22	2-5
Error code: 23	2-6
Error code: 24	2-6
Error code: 25	2-6
Error code: 26	2-6
Error code: 27	2-7

Error code: 28	2-7
Error code: 29	2-7
Error code: 30	2-7
Error code: 31	2-8
Error code: 32	2-8
Error code: 33	2-8
Error code: 34	2-9
Error code: 35	2-9
Error code: 36	2-9
Error code: 37	2-9
Error code: 38	2-10
Error code: 39	2-10
Error code: 40	2-10
Error code: 41	2-10
Error code: 43	2-10
Error code: 44	2-11
Error code: 45	2-11
Error code: 46	2-11
Error code: 47	2-11
Error code: 48	2-12
Error code: 49	2-12
Error code: 50	2-12
Error code: 51	2-12
Error code: 52	2-13
Error code: 53	2-13
Error code: 54	2-13
Error code: 55	2-13
Error code: 56	2-13
Error code: 57	2-14
Error code: 58	2-14
Error code: 59	2-14
Error code: 60	2-14
Error code: 61	2-15
Error code: 62	2-15
Error code: 63	2-15

Error code: 64.....	2-15
Error code: 65.....	2-15
Error code: 66.....	2-16
Error code: 67.....	2-16
Error code: 68.....	2-16
Error code: 69.....	2-16
Error code: 70.....	2-16
Error code: 71.....	2-17
Error code: 72.....	2-17
Error code: 73.....	2-17
Error code: 74.....	2-17
Error code: 75.....	2-18
Error code: 77.....	2-18
Error code: 78.....	2-18
Error code: 79.....	2-18
Error code: 80.....	2-18
Error code: 81.....	2-19
Error code: 82.....	2-19
Error code: 83.....	2-19
Error code: 84.....	2-19
Error code: 85.....	2-20
Error code: 86.....	2-20
Error code: 87.....	2-20
Error code: 88.....	2-20
Error code: 89.....	2-21
Error code: 90.....	2-21
Error code: 91.....	2-21
Error code: 92.....	2-21
Error code: 93.....	2-21
Error code: 94.....	2-22
Error code: 96.....	2-22
Error code: 97.....	2-22
Error code: 98.....	2-22
Error code: 99.....	2-23
Error code: 100.....	2-23

Error code: 101	2-23
Error code: 102	2-23
Error code: 103	2-23
Error code: 104	2-24
Error code: 105	2-24
Error code: 106	2-24
Error code: 108	2-24
Error code: 109	2-25
Error code: 110	2-25
Error code: 111	2-25
Error code: 112	2-25
Error code: 113	2-25
Error code: 114	2-25
Error code: 115	2-26
Error code: 116	2-26
Error code: 118	2-26
Error code: 119	2-26
Error code: 120	2-26
Error code: 122	2-27
Error code: 125	2-27
Error code: 126	2-27
Error code: 127	2-27
Error code: 128	2-28
Error code: 129	2-28
Error code: 130	2-28
Error code: 131	2-28
Error code: 133	2-29
Error code: 135	2-29
Error code: 138	2-29
Error code: 139	2-29
Error code: 140	2-30
Error code: 141	2-30
Error code: 142	2-30
Error code: 143	2-30
Error code: 144	2-30

Error code: 145.....	2-31
Error code: 146.....	2-31
Error code: 147.....	2-31
Error code: 148.....	2-31
Error code: 149.....	2-31
Error code: 150.....	2-31
Error code: 151.....	2-32
Error code: 152.....	2-32
Error code: 153.....	2-32
Error code: 154.....	2-32
Error code: 155.....	2-32
Error code: 156.....	2-32
Error code: 157.....	2-33
Error code: 158.....	2-33
Error code: 159.....	2-33
Error code: 160.....	2-33
Error code: 161.....	2-33
Error code: 162.....	2-34
Error code: 163.....	2-34
Error code: 164.....	2-34
Error code: 165.....	2-34
Error code: 166.....	2-34
Error code: 168.....	2-34
Error code: 169.....	2-35
Error code: 170.....	2-35
Error code: 171.....	2-35
Error code: 172.....	2-35
Error code: 173.....	2-35
Error code: 174.....	2-36
Error code: 175.....	2-36
Error code: 176.....	2-36
Builder Exceptions (1001 – 1042)	2-36
Error code: 1001.....	2-36
Error code: 1002.....	2-36
Error code: 1003.....	2-37

Error code: 1004	2-37
Error code: 1005	2-37
Error code: 1006	2-38
Error code: 1007	2-38
Error code: 1008	2-38
Error code: 1009	2-38
Error code: 1010	2-39
Error code: 1011	2-39
Error code: 1012	2-39
Error code: 1013	2-39
Error code: 1015	2-40
Error code: 1016	2-40
Error code: 1017	2-40
Error code: 1018	2-40
Error code: 1019	2-40
Error code: 1020	2-41
Error code: 1021	2-41
Error code: 1022	2-41
Error code: 1023	2-41
Error code: 1024	2-42
Error code: 1025	2-42
Error code: 1026	2-42
Error code: 1027	2-42
Error code: 1028	2-43
Error code: 1029	2-43
Error code: 1030	2-43
Error code: 1031	2-43
Error code: 1032	2-43
Error code: 1033	2-44
Error code: 1034	2-44
Error code: 1035	2-44
Error code: 1036	2-44
Error code: 1037	2-44
Error code: 1039	2-44
Error code: 1040	2-45

Error code: 1041.....	2-45
Error code: 1042.....	2-45
Concurrency Exceptions (2001 – 2004)	2-45
Error code: 2001.....	2-45
Error code: 2002.....	2-45
Error code: 2003.....	2-46
Error code: 2004.....	2-46
Conversion Exceptions (3001 – 3007)	2-46
Error code: 3001.....	2-46
Error code: 3003.....	2-46
Error code: 3004.....	2-46
Error code: 3005.....	2-47
Error code: 3006.....	2-47
Error code: 3007.....	2-47
Database Exceptions (4001 – 4018)	2-47
Error code: 4002.....	2-47
Error code: 4003.....	2-47
Error code: 4005.....	2-48
Error code: 4006.....	2-48
Error code: 4007.....	2-48
Error code: 4008.....	2-48
Error code: 4009.....	2-48
Error code: 4011.....	2-49
Error code: 4014.....	2-49
Error code: 4015.....	2-49
Error code: 4016.....	2-50
Error code: 4017.....	2-50
Error code: 4018.....	2-50
Optimistic Lock Exceptions (5001 – 5007)	2-50
Error code: 5001.....	2-50
Error code: 5003.....	2-50
Error code: 5004.....	2-51
Error code: 5006.....	2-51
Error code: 5007.....	2-51
Error code: 5008.....	2-51

Query Exceptions (6001 – 6092)	2-52
Error code: 6001	2-52
Error code: 6002	2-52
Error code: 6003	2-52
Error code: 6004	2-52
Error code: 6005	2-53
Error code: 6006	2-53
Error code: 6007	2-53
Error code: 6008	2-54
Error code: 6013	2-54
Error code: 6014	2-54
Error code: 6015	2-54
Error code: 6016	2-54
Error code: 6020	2-55
Error code: 6021	2-55
Error code: 6023	2-55
Error code: 6024	2-55
Error code: 6026	2-55
Error code: 6027	2-56
Error code: 6028	2-56
Error code: 6029	2-56
Error code: 6030	2-56
Error code: 6031	2-56
Error code: 6032	2-57
Error code: 6034	2-57
Error code: 6041	2-57
Error code: 6042	2-57
Error code: 6043	2-57
Error code: 6044	2-57
Error code: 6045	2-58
Error code: 6046	2-58
Error code: 6047	2-58
Error code: 6048	2-58
Error code: 6049	2-58
Error code: 6050	2-59

Error code: 6051.....	2-59
Error code: 6052.....	2-59
Error code: 6054.....	2-59
Error code: 6055.....	2-59
Error code: 6056.....	2-60
Error code: 6057.....	2-60
Error code: 6058.....	2-60
Error code: 6059.....	2-60
Error code: 6060.....	2-60
Error code: 6061.....	2-61
Error code: 6062.....	2-61
Error code: 6063.....	2-61
Error code: 6064.....	2-61
Error code: 6065.....	2-62
Error code: 6066.....	2-62
Error code: 6068.....	2-62
Error code: 6069.....	2-62
Error code: 6070.....	2-62
Error code: 6071.....	2-63
Error code: 6072.....	2-63
Error code: 6073.....	2-63
Error code: 6074.....	2-63
Error code: 6075.....	2-63
Error code: 6076.....	2-64
Error code: 6077.....	2-64
Error code: 6078.....	2-64
Error code: 6079.....	2-64
Error code: 6080.....	2-64
Error code: 6081.....	2-64
Error code: 6082.....	2-65
Error code: 6083.....	2-65
Error code: 6084.....	2-65
Error code: 6085.....	2-65
Error code: 6086.....	2-65
Error code: 6087.....	2-66

Error code: 6088	2-66
Error code: 6089	2-66
Error code: 6090	2-66
Error code: 6091	2-66
Error code: 6092	2-66
Validation Exceptions (7001 – 7097)	2-67
Error code: 7001	2-67
Error code: 7002	2-67
Error code: 7003	2-67
Error code: 7004	2-67
Error code: 7008	2-68
Error code: 7009	2-68
Error code: 7010	2-68
Error code: 7011	2-68
Error code: 7012	2-69
Error code: 7013	2-69
Error code: 7017	2-69
Error code: 7018	2-69
Error code: 7023	2-69
Error code: 7024	2-70
Error code: 7025	2-70
Error code: 7027	2-70
Error code: 7028	2-70
Error code: 7030	2-70
Error code: 7031	2-70
Error code: 7032	2-71
Error code: 7033	2-71
Error code: 7038	2-71
Error code: 7039	2-71
Error code: 7040	2-71
Error code: 7042	2-72
Error code: 7043	2-72
Error code: 7044	2-72
Error code: 7047	2-72
Error code: 7048	2-72

Error code: 7051.....	2-72
Error code: 7052.....	2-73
Error code: 7053.....	2-73
Error code: 7054.....	2-73
Error code: 7055.....	2-73
Error code: 7056.....	2-74
Error code: 7058.....	2-74
Error code: 7059.....	2-74
Error code: 7060.....	2-74
Error code: 7061.....	2-74
Error code: 7062.....	2-75
Error code: 7063.....	2-75
Error code: 7064.....	2-75
Error code: 7065.....	2-75
Error code: 7066.....	2-75
Error code: 7068.....	2-75
Error code: 7069.....	2-76
Error code: 7070.....	2-76
Error code: 7071.....	2-76
Error code: 7073.....	2-76
Error code: 7074.....	2-76
Error code: 7075.....	2-76
Error code: 7076.....	2-77
Error code: 7077.....	2-77
Error code: 7078.....	2-77
Error code: 7079.....	2-77
Error code: 7080.....	2-77
Error code: 7081.....	2-78
Error code: 7082.....	2-78
Error code: 7083.....	2-78
Error code: 7084.....	2-78
Error code: 7085.....	2-78
Error code 7087:.....	2-78
Error code 7088:.....	2-79
Error code 7089:.....	2-79

Error code 7091	2-79
Error code 7092	2-79
Error code 7093	2-80
Error code 7095	2-80
Error code 7096	2-80
Error code 7097:	2-80
Error Code: 7099	2-80
Error Code: 7101	2-80
EJBQL Exceptions (8001 – 8009)	2-81
Error Code: 8001	2-81
Error Code: 8002	2-81
Error Code: 8003	2-81
Error Code: 8004	2-81
Error Code: 8005	2-81
Error Code: 8006	2-82
Error Code: 8009	2-82
Synchronization Exceptions (8050 – 8070)	2-82
Error Code: 8050	2-82
Error Code: 8051	2-82
Error Code: 8052	2-82
Error Code: 8053	2-82
Error Code: 8054	2-83
Error Code: 8055	2-83
Error Code: 8056	2-83
Error Code: 8057	2-83
Error Code: 8058	2-83
Error Code: 8059	2-84
Error Code: 8070	2-84

3 Mapping Workbench

Error Messages	3-1
Classpath Issues	3-12
Database Connections	3-13
Troubleshooting Descriptors	3-13

4 Frequently Asked Questions (FAQs)

General.....	4-1
Mappings.....	4-4
Sessions.....	4-12
Unit of Work.....	4-14
Querying.....	4-16
TopLink Mapping Workbench.....	4-19
Enterprise.....	4-21

5 Entity Deployment

Generating Deployment JARs.....	5-1
Common BEA WebLogic Deployment Exceptions.....	5-1
Assertion Error.....	5-2
Error Deploying Application.....	5-2
Exception 8001.....	5-2
Exception 8016.....	5-3
Cannot Startup Connection Pool.....	5-3
Error Message.....	5-4
EJBC Found Errors.....	5-4
EJB Deployment Exception.....	5-4
Deploying EJB Component.....	5-5
Cannot Startup Connection Pool.ejbPool.....	5-5
Other Errors.....	5-5
Problems on IBM WebSphere Server Startup or IBM WebSphere Bean Deployment.....	5-6
Class Not Found Exceptions.....	5-6
oracle.toplink.exceptions.DatabaseException.....	5-6
Exception [6066].....	5-7
Exception [7064].....	5-7
Exception [7066].....	5-8
Exception [7068].....	5-8
Exception [7069].....	5-8
Exception [7070].....	5-8
Exception [7079].....	5-9
Exception [7101].....	5-9
Exception [9002].....	5-9

Problems Occurring at Runtime Using TopLink for WebSphere Foundation Library.....	5-9
Exception [6026].....	5-9
Common WebSphere Exceptions.....	5-10
Class Not Found Exceptions.....	5-10
Common BEA WebLogic 6.1 Exceptions	5-10
Development Exceptions.....	5-10
Missing Persistence Type	5-10
Error Loading Persistence Resource	5-11
Wrong BEA WebLogic Version.....	5-11
Deployment/runtime Exceptions	5-11
Missing Persistence Type	5-11
Error Loading Persistence Resource	5-12
Wrong Persistence Version.....	5-12
Cannot Startup Data Source	5-13
Wrong WebLogic Version.....	5-13
Common WebLogic 7.0 Exceptions	5-14
Development-time Exceptions:.....	5-14
Missing Persistence Type	5-14
Error Loading Persistence Resource.....	5-14
Wrong WebLogic Version.....	5-15
Deployment/runtime Exceptions:	5-15
Missing Persistence Type	5-15
Error Loading Persistence Resource.....	5-16
Cannot Startup Data Source	5-16

Index

Send Us Your Comments

Oracle9iAS TopLink Troubleshooting Guide, Release 2 (9.0.3)

Part No. B10068-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: iasdocs_us@oracle.com
- FAX: 650-506-7407 Attn: Oracle9i Application Server Documentation Manager
- Postal service:
Oracle Corporation
Oracle9i Application Server Documentation
500 Oracle Parkway, M/S 2op3
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This document contains general information about TopLink's error handling strategy, the types of errors that can occur, and Frequently Asked Questions (FAQs).

This preface contains the following topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Structure](#)
- [Related Documents](#)
- [Conventions](#)

Intended Audience

This document is intended for developers who use TopLink or deploy TopLink-enabled projects.

This document assumes that you are familiar with the concepts of object-oriented programming, the Enterprise JavaBeans (EJB) specification, and with your own particular Java development environment.

The document also assumes that you are familiar with your particular operating system (such as Windows, UNIX, or other). The general operation of any operating system is described in the user documentation for that system, and is not repeated in this manual.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

This document includes the following chapters:

Chapter 1, "Runtime and Development Exceptions"

This chapter describes the TopLink exception classes.

Chapter 2, "Error Codes and Descriptions"

This section lists each TopLink error code, including its probable cause and recommended action.

Chapter 3, "Mapping Workbench"

This section contains information on troubleshooting the Mapping Workbench.

Chapter 4, "Frequently Asked Questions (FAQs)"

This section contains frequently asked questions and their answers about using TopLink and the Mapping Workbench.

Chapter 5, "Entity Deployment"

This section discusses some of the general troubleshooting issues surrounding entity bean configuration and deployment and lists many of the common exceptions and error messages.

Related Documents

For more information, see these Oracle resources:

Oracle9iAS TopLink: Getting Started

Provides installation procedures to install and configure TopLink. It also introduces the concepts with which you should be familiar to get the most out of TopLink.

Oracle9iAS TopLink: Tutorial

Provides tutorials illustrating the use of TopLink. It is written for developers who are familiar with the object-oriented programming and Java development environments.

Oracle9iAS TopLink: Foundation Library Guide

Introduces TopLink and the concepts and techniques required to build an effective TopLink application. It also gives a brief overview of relational databases and describes how TopLink accesses relational databases from the object-oriented Java domain.

Oracle9iAS TopLink: Mapping Workbench Reference Guide

Includes the concepts required for using the TopLink Mapping Workbench, a stand-alone application that creates and manages your descriptors and mappings for a project. This document includes information on each Mapping Workbench function and option and is written for developers who are familiar with the object-oriented programming and Java development environments.

Oracle9iAS TopLink: Container Managed Persistence for Application Servers

Provides information on TopLink container-managed persistence (CMP) support for application servers. Oracle provides an individual document for each application server specifically supported by TopLink CMP.

TopLink: Troubleshooting

Contains general information about TopLink's error handling strategy, the types of errors that can occur, and Frequently Asked Questions (FAQs). It also discusses troubleshooting procedures and provides a list of the exceptions that can occur, the most probable cause of the error condition, and the recommended action.

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/admin/account/membership.html>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/docs/index.htm>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Microsoft Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter <code>sqlplus</code> to open SQL*Plus. The password is specified in the <code>orapwd</code> file. Back up the datafiles and control files in the <code>/disk1/oracle/dbs</code> directory. The <code>department_id</code> and <code>location_id</code> columns are in the <code>hr.departments</code> table. Set the <code>QUERY_REWRITE_ENABLED</code> initialization parameter to <code>true</code> . Connect as <code>oe</code> user. The <code>JRepUtil</code> class implements these methods.
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>Uold_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	<code>DECIMAL (digits [, precision])</code>
{ }	Braces enclose two or more items, one of which is required.	<code>{ENABLE DISABLE}</code>

Convention	Meaning	Example
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
Other notation	You must enter symbols other than brackets, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>

Conventions for Microsoft Windows Operating Systems

The following table describes conventions for Microsoft Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Oracle Database Configuration Assistant, choose Start > Programs >
Case sensitivity and file and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt\" \"system32 is the same as C:\WINNT\SYSTEM32

Convention	Meaning	Example
	IMPORTANT NOTE: File names and directory names <i>are</i> case sensitive under UNIX. Where the name of a file or directory is mentioned and the operating system is a non-Windows platform, you must enter the names exactly as they appear unless instructed otherwise.	
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.	C:\oracle\oradata>
	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
INSTALL_DIR	Represents the Oracle home installation directory name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	SET CLASSPATH=INSTALL_DIR\jre\bin

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin95 for Windows 95 ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install Oracle9i release 1 (9.0.1) on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\ora90. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle9i Database Getting Starting for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

Runtime and Development Exceptions

This chapter describes the TopLink exception classes. Each type of exception corresponds to a range of error code (see [Chapter 2, "Error Codes and Descriptions"](#)).

TopLinkExceptions

All TopLink exceptions are descendents of `RuntimeException`. The `TopLinkException` class is the superclass of all runtime and development type exceptions.

Runtime Exceptions

Runtime exceptions indicate error conditions at runtime, and do not necessarily indicate fatal errors. Instead, they indicate that runtime conditions are invalid, such as the loss of database connection. All of these exceptions should be handled in a try-catch block.

The following exceptions can be thrown at runtime:

- `DatabaseException`
- `OptimisticLockException`
- `CommunicationException`

Development Exceptions

Development exceptions indicate that a certain fragment of code is invalid. All development exceptions do not depend on runtime conditions and must therefore be solved before deploying the application. For example, the `DescriptorException` is thrown the first time an application is executed if an

erroneous descriptor or mapping property is set. Development exceptions are useful as a debugging tool to find inconsistencies in the descriptor. Because development exceptions are not normal behavior, they must not be handled in a try-catch block.

The following exceptions are not dependent on runtime conditions. If one of these exceptions is thrown, the application code being tested is invalid and must be changed. Avoid handling these types of exceptions:

- `BuilderException`
- `ConcurrencyException`
- `ConversionException`
- `DescriptorException`
- `QueryException`
- `ValidationException`

Format of Exceptions

All exceptions return the name of the exception and a message describing what caused the exception. The message given is dependent on the type of exception.

Each TopLink exception thrown has the following information:

- The name of TopLink exception
- A description of the most probable cause of the error
- A native error code

BuilderException

A development exception that is raised when the Builder file format for the descriptor is not in a proper state. If possible, the message returned gives the line number of the appropriate descriptor file that caused the exception.

The source and line number are not printed only when it is possible for TopLink to detect the actual source of problem. Internal exception is printed when `BuilderException` was caused due to the exception thrown by the VM.

Format

```
EXCEPTION [TOPLINK - error code]: Exception name
```

```
EXCEPTION DESCRIPTION: Message
INTERNAL EXCEPTION: Message
SOURCE: The source to the descriptor file that caused the error.
LINE NUMBER: The line number that caused the exception to be raised. This is the
line number in the descriptor file.
```

Example 1–1 BuilderException

```
EXCEPTION [TOPLINK - 1038]: oracle.toplink.tools.builderreader.BuilderException
EXCEPTION DESCRIPTION: No such section token: ABC
```

CommunicationException

A runtime exception that wraps all RMI, CORBA, or IO exceptions that may occur.

ConcurrencyException

A development exception that is raised when a Java concurrency violation occurs. Internal Exception is shown only when the running thread was interrupted and VM threw `InterruptedException`.

Format

```
EXCEPTION [TOPLINK - error code]: Exception name
EXCEPTION DESCRIPTION: Message
INTERNAL EXCEPTION: Message
```

Example 1–2 ConcurrencyExcpetion

```
EXCEPTION [TOPLINK - 2004]: oracle.toplink.exceptions.ConcurrencyException
EXCEPTION DESCRIPTION: Signal attempted before wait on concurrency manager.
This normally means that an attempt was made to commit or rollback a transaction
before being started, or rolledback twice.
```

ConversionException

A development exception that is raised when a conversion error occurs by an incompatible type conversion. The message returned indicates which type cast caused the exception.

Format

```
EXCEPTION [TOPLINK - error code]: Exception name
EXCEPTION DESCRIPTION: Message
INTERNAL EXCEPTION: Message
```

Example 1-3 ConversionException

```
EXCEPTION [TOPLINK - 3006]: oracle.toplink.exceptions.ConversionException
EXCEPTION DESCRIPTION: object must be of even length to be converted to a
ByteArray
```

DatabaseException

A runtime exception that is raised when data read from the database or the data that is to be written to the database is incorrect. The exception also acts as a wrapper for `SQLException` in which case it contains a reference to the error code and error message. This exception can occur on any database type operation.

Internal exception and error code are printed when the exception is wrapping `SQLException`.

Format

```
EXCEPTION [TOPLINK - error code]: Exception name
EXCEPTION DESCRIPTION: Message
INTERNAL EXCEPTION: Message
ERROR CODE: Error code
```

Example 1-4 DatabaseException

```
EXCEPTION [TOPLINK - 4002]: oracle.toplink.exceptions.DatabaseExceptions
EXCEPTION DESCRIPTION: java.sql.SQLException: [INTERSOLV][ODBC dBase driver]
Incompatible datatypes in expression: >
INTERNAL EXCEPTION: java.sql.SQLException: [INTERSOLV][ODBC dBase driver]
Incompatible datatypes in expression: >
ERROR CODE: 3924
```

DescriptorException

A development exception that is raised when insufficient information has been provided to the descriptor. The message returned includes the name of the descriptor or mapping that caused the exception. If a mapping within the descriptor

caused the error, the name and parameters of the mapping are part of the returned message.

Internal exception, mapping and descriptor are optional. They are displayed only when TopLink has enough information about the source of the problem.

Format

```
EXCEPTION [TOPLINK - error code]: Exception name
EXCEPTION DESCRIPTION: Message
INTERNAL EXCEPTION: Message
MAPPING: Database mapping
DESCRIPTOR: Descriptor
```

Example 1-5 DescriptorException

```
EXCEPTION [TOPLINK - 75]: oracle.toplink.exceptions.DescriptorException
EXCEPTION DESCRIPTION: The reference class is not specified.
```

OptimisticLockException

A runtime exception that is raised when the row on the database matching the desired object is missing or when the value on the database does not match the registered number. It is used in conjunction with the optimistic locking feature. This applies only on update or delete operation. Refer to the *Oracle9iAS TopLink: Mapping Workbench Reference Guide* for more information on optimistic locking. These exceptions should be handled in a try-catch block.

Format

```
EXCEPTION [TOPLINK - error code]: Exception Name
EXCEPTION DESCRIPTION: Message
```

Example 1-6 OptimisticLockException

```
EXCEPTION [TOPLINK - 5003]: oracle.toplink.exceptions.OptimisticLockException
EXCEPTION DESCRIPTION: The object, object.toString() cannot be deleted because
it has changed or been deleted since it was last read.
```

QueryException

A development exception that is raised when insufficient information has been provided to the query. If possible, the message indicates the query that caused the

exception. Query is optional and is displayed if TopLink has a query that caused this exception.

Format

```
EXCEPTION [TOPLINK - error code]: Exception name
EXCEPTION DESCRIPTION: Message
QUERY:
```

Example 1-7 QueryException

```
EXCEPTION [TOPLINK - 6026]: oracle.toplink.exceptions.QueryException
EXCEPTION DESCRIPTION: Query is not defined.
```

TopLinkException

Superclass for all TopLink-generated exceptions. It is an abstract class.

ValidationException

A development exception that is raised when an incorrect state is detected or an API is used incorrectly.

Format

```
EXCEPTION [TOPLINK - error code]: Exception name
EXCEPTION DESCRIPTION: Message
```

Example 1-8 ValidationException

```
EXCEPTION [TOPLINK - 6026]: oracle.toplink.exceptions.ValidationException
EXCEPTION DESCRIPTION: Child descriptors do not have an identity map, they share
their parent's.
```

Error Codes and Descriptions

This section lists each TopLink error code. Each error entry contains a description of the **error**, the probable **cause**, and the recommended **action**. Each error code corresponds to an exception class. See "[Runtime and Development Exceptions](#)" on page 1-1 for more information.

Format:

A description shown in the actual exception thrown.

Cause: The most probable cause for the error.

Action: Suggestions for resolving the error.

Descriptor Exceptions (1 – 176)

Error code: 1**ATTRIBUTE_AND_MAPPING_WITH_INDIRECTION_MISMATCH**

Cause: *<attributeName>* is not declared as type `ValueHolderInterface` but the mapping uses indirection. Mapping is set to use indirection but the related attribute is not defined as type `ValueHolderInterface`. It is thrown on foreign reference mappings.

Action: If you want to use indirection on the mapping, change the attribute to type `ValueHolderInterface`. Otherwise, change the mapping associated with the attribute so that it does not use indirection.

Error code: 2**ATTRIBUTE_AND_MAPPING_WITHOUT_INDIRECTION_MISMATCH**

Cause: *<attributeName>* is declared as type `ValueHolderInterface` but the mapping is not using indirection. Attribute is defined to be of type `ValueHolderInterface` but the mapping is not set to use indirection. It is thrown on foreign reference mappings.

Action: If you do not want to use indirection on the mapping, change the attribute to not be of type `ValueHolderInterface`. Otherwise, change the mapping associated with the attribute to use indirection.

Error code: 6

ATTRIBUTE_NAME_NOT_SPECIFIED

Cause: Attribute name is missing or not specified in the mapping definition.

Action: Specify the attribute name in the mapping by calling method `setAttributeName(String attribute name)`.

Error code: 7

ATTRIBUTE_TYPE_NOT_VALID

Cause: *<attributeName>* should be defined as type `Vector`, or a type that implements `Map` or `Collection` if using Java 2. It happens in one to many mapping, many to many mapping and collection mapping when mapping is set not to use indirection and attribute type is not declared of type `java.util.Vector`.

Action: Declare the attribute to be of type `java.util.Vector`.

Error code: 8

CLASS_INDICATOR_FIELD_NOT_FOUND

Cause: The class indicator field has not been defined, however the descriptor has been set to use inheritance. When using inheritance, a class indicator field or class extraction method must be set. The class indicator field is used to create the right type of domain object

Action: Either a class indicator field or class extraction method must be set.

Error code: 9

DIRECT_FIELD_NAME_NOT_SET

Cause: The direct field name from the target table is not set in the direct collection mapping. The field values form the collection in this mapping.

Action: Specify the direct field name by calling method `setDirectFieldName(String fieldName)`.

Error code: 10

FIELD_NAME_NOT_SET_IN_MAPPING

Cause: The field name is not set in the mapping. It is thrown from direct to field mapping, array mapping and structure mapping.

Action: Specify the field name by calling method `setFieldName(String fieldName)`.

Error code: 11

FOREIGN_KEYS_DEFINED_INCORRECTLY

Cause: One to one mapping foreign key defined incorrectly. Multiple foreign key fields were set for one to one mapping by calling method `setForeignKeyFieldName(String fieldName)`.

Action: Use method `addForeignKeyFieldName(String sourceForeignKeyName, String targetPrimaryKeyFieldName)` to add multiple foreign key fields.

Error code: 12

IDENTITY_MAP_NOT_SPECIFIED

Cause: The descriptor must use an identity map to use the **Check cache does exist** option. Descriptor has been set to not use identity map but the existence checking is set to be done on identity map.

Action: Either use identity map or set the existence checking to some other option.

Error code: 13

ILLEGAL_ACCESS_WHILE_GETTING_VALUE_THRU_INSTANCE_VARIABLE_ACCESSOR

Cause: `<attributeName>` instance variable in object `<objectName>` is inaccessible. Instance variable in the domain object is not accessible. This exception is thrown when TopLink tries to access the instance variable using Java reflection. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 14

ILLEGAL_ACCESS_WHILE_CLONING

Cause: Problem in cloning the object *<domainObject>*. Clone method *<methodName>* is not accessible. The method name specified using `useCloneCopyPolicy(String cloneMethodName)` or the `clone()` method to create clone on the domain object is not accessible by TopLink using Java reflection. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 15

ILLEGAL_ACCESS_WHILE_CONSTRUCTOR_INSTANTIATION

Cause: The domain class does not define a public default constructor, which is needed by TopLink to create new instances of the domain class.

Action: Define one or use different instantiation policy.

Error code: 16

ILLEGAL_ACCESS_WHILE_EVENT_EXECUTION

Cause: The descriptor callback method *<eventMethodName>* with `DescriptorEvent` as argument, is not accessible. This exception is thrown when TopLink tries to access the event method using Java reflection. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 17

ILLEGAL_ACCESS_WHILE_GETTING_VALUE_THRU_METHOD_ACCESSOR

Cause: Trying to invoke inaccessible *<methodName>* on the object *<objectName>*. The underlying get accessor method to access an attribute in the domain object is not accessible. This exception is thrown when TopLink tries to access an attribute through method using Java reflection. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 18**ILLEGAL_ACCESS_WHILE_INSTANTIATING_METHOD_BASED_PROXY**

Cause: The method used by the Transformation mapping using a `ValueHolder` is illegal. This exception is thrown when TopLink tries to access the method using Java reflection. The problem is caused when method base value holder is getting instantiated.

Action: Inspect the internal exception and check the Java manuals.

Error code: 19**ILLEGAL_ACCESS_WHILE_INVOKING_ATTRIBUTE_METHOD**

Cause: On transformation mapping the underlying attribute method used to retrieve value from the database row while reading transformation mapped attribute is not accessible.

Action: Inspect the internal exception and check the Java manuals.

Error code: 20**ILLEGAL_ACCESS_WHILE_INVOKING_FIELD_TO_METHOD**

Cause: On transformation mapping the method `<methodName>` used to retrieve value from the object while writing transformation mapped attribute is not accessible. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 21**ILLEGAL_ACCESS_WHILE_INVOKING_ROW_EXTRACTION_METHOD**

Cause: Problem in extracting class from `<row>`; Static method `<method>` with `<databaseRow>` as argument, is not accessible. The method to extract class from row on the domain object is not accessible. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 22**ILLEGAL_ACCESS_WHILE_METHOD_INSTANTIATION**

Cause: Problem in creating new instance; the method *<methodName>* to create instances on the domain class is not accessible. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 23

ILLEGAL_ACCESS_WHILE_OBSOLETE_EVENT_EXECUTION

Cause: The descriptor callback method *<eventMethodName>* with Session as argument, is inaccessible. This exception is thrown when TopLink tries to access the event method using Java reflection. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 24

ILLEGAL_ACCESS_WHILE_SETTING_VALUE_THRU_INSTANCE_VARIABLE_ACCESSOR

Cause: The *<attributeName>* instance variable in the object *<objectName>* is not accessible through Java reflection. The error is thrown by Java and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 25

ILLEGAL_ACCESS_WHILE_SETTING_VALUE_THRU_METHOD_ACCESSOR

Cause: Trying to invoke inaccessible method *<setMethodName>* on the object with parameter *<parameter>*. The attribute's set accessor method is not accessible through Java reflection. The error is thrown by Java and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 26

ILLEGAL_ARGUMENT_WHILE_GETTING_VALUE_THRU_INSTANCE_VARIABLE_ACCESSOR

Cause: Trying to get a value for an instance variable *<attributeName>* of type *<typeName>* from the object. The specified object is not an instance of the class

or interface declaring the underlying field. An object is accessed to get the value of an instance variable that does not exist.

Action: Inspect the internal exception and check the Java manuals.

Error code: 27

ILLEGAL_ARGUMENT_WHILE_GETTING_VALUE_THRU_METHOD_ACCESSOR

Cause: Trying to invoke method *<methodName>* on the object *<objectName>*. The get accessor method declaration on the domain object differs from the one that is defined. The number of actual and formal parameters differ, or an unwrapping conversion has failed.

Action: Inspect the internal exception and check the Java manuals.

Error code: 28

ILLEGAL_ARGUMENT_WHILE_INSTANTIATING_METHOD_BASED_PROXY

Cause: The method used by the method-based proxy in a Transformation mapping is getting illegal arguments when the value holder is getting instantiated. This exception is thrown when TopLink tries to access the method using Java reflection.

Action: Inspect the internal exception and check the Java manuals.

Error code: 29

ILLEGAL_ARGUMENT_WHILE_INVOKING_ATTRIBUTE_METHOD

Cause: The number of actual and formal parameters differ, or an unwrapping conversion has failed. On transformation mapping the method used to retrieve value from the database row while reading transformation mapped attribute is getting illegal argument.

Action: Inspect the internal exception and check the Java manuals.

Error code: 30

ILLEGAL_ARGUMENT_WHILE_INVOKING_FIELD_TO_METHOD

Cause: The number of actual and formal parameters differ for method *<methodName>* or an unwrapping conversion has failed. On transformation mapping the method used to retrieve value from the object while writing

transformation mapped attribute is getting illegal argument. The error is purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 31

ILLEGAL_ARGUMENT_WHILE_OBSOLETE_EVENT_EXECUTION

Cause: The number of actual and formal parameters for the descriptor callback method *<eventName>* differs, or an unwrapping conversion has failed. The callback event method is invoked with illegal argument. This exception is thrown when TopLink tries to invoke the event method using Java reflection. The error is a purely Java exception and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 32

ILLEGAL_ARGUMENT_WHILE_SETTING_VALUE_THRU_INSTANCE_VARIABLE_ACCESSOR

Cause: Illegal value is being assigned to the attribute instance variable. Trying to set a value *<value>* for an instance variable *<attributeName>* of type *<typeName>* in the object. The specified object is not an instance of the class or interface declaring the underlying field, or an unwrapping conversion has failed.

TopLink does the assignment of value by using Java reflection. Java throws the error and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 33

ILLEGAL_ARGUMENT_WHILE_SETTING_VALUE_THRU_METHOD_ACCESSOR

Cause: Illegal argument is being passed to the attribute's set accessor method. Trying to invoke method *<setMethodName>* on the object. The number of actual and formal parameters differs, or an unwrapping conversion has failed. Java throws the error and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 34**INSTANTIATION_WHILE_CONSTRUCTOR_INSTANTIATION**

Cause: The class does not define a public default constructor, or the constructor raised an exception. The default constructor for the domain object is invoked to create new instance of the object while building new domain objects. If this Class represents an abstract class, an interface, an array class, a primitive type, or void; or if the instantiation fails for some other reason. Java throws the error and TopLink only wraps the reflection exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 35**INVALID_DATA_MODIFICATION_EVENT**

Cause: This is an exception that an application should never encounter. The exception can occur at the time of developing TopLink. In cases where one writes new mapping it is possible to get this exception. In direct collection mapping and many to many mapping the target table and relational table are populated at the end of the commit process and if data modification event is sent to any other mapping then this exception is thrown.

Action: Contact Technical Support.

Error code: 36**INVALID_DATA_MODIFICATION_EVENT_CODE**

Cause: This is an exception that an application should never encounter. The exception can occur at the time of developing TopLink. In cases where one writes new mapping it is possible to get this exception. In direct collection mapping and many to many mapping the target table and relational table are populated at the end of the commit process and if data modification event is sent to these two mappings with wrong event code then this exception is thrown.

Action: Contact Technical Support.

Error code: 37**INVALID_DESCRIPTOR_EVENT_CODE**

Cause: This is an exception that an application should never encounter. The exception can occur at the time of developing TopLink. The exception means

that descriptor event manager does not support the event code passed in the event.

Action: Contact Technical Support.

Error code: 38

INVALID_IDENTITY_MAP

Cause: Identity map constructor failed because an invalid identity map was specified. The identity map class given in the descriptor cannot be instantiated. The exception is Java exception thrown by Java reflection when TopLink is instantiating the identity map class. TopLink only wraps the Java exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 39

JAVA_CLASS_NOT_SPECIFIED

Cause: The descriptor does not define a Java class. The Java class is not specified in the descriptor.

Action: Specify the Java Class

Error code: 40

DESCRIPTOR_FOR_INTERFACE_IS_MISSING

Cause: A descriptor for the referenced interface is not added to the session.

Action: Add that descriptor to the session.

Error code: 41

MAPPING_FOR_SEQUENCE_NUMBER_FIELD

Cause: A non-read-only mapping is not defined for the sequence number field. A mapping is required so that TopLink can put and extract values for the primary key.

Action: Define a mapping.

Error code: 43

MISSING_CLASS_FOR_INDICATOR_FIELD_VALUE

Cause: Missing class for indicator field value *<classFieldValue>* of type *<type>*. There was no class entry found in the inheritance policy for the indicator field

value read from the database. Probably the method `addClassIndicator(Class class, Object typeValue)` was not called for the field value. The class and `typeValue` is stored in the hashtable and later on the class is extracted from the hashtable by passing `typeValue` as a key. Remember `Integer(1)` is not equal to `Float(1)`, this is another major reason for this problem when the type of `typeValue` is different.

Action: Check descriptor.

Error code: 44

MISSING_CLASS_INDICATOR_FIELD

Cause: The class indicator field is missing from the database row `<row>` that was read from the database. This is done in inheritance model where after reading rows from the database, child domain objects are to be constructed depending upon the type indicator values.

Action: Check the printed row to make sure the spelling is correct.

Error code: 45

MISSING_MAPPING_FOR_FIELD

Cause: Missing mapping for field `<field>`; a mapping for the field is not specified.

Action: Define a mapping for the field.

Error code: 46

NO_MAPPING_FOR_PRIMARY_KEY

Cause: A mapping for the primary key is not specified. There should be one non-read-only mapping defined for the primary key field.

Action: Define a mapping for the primary key.

Error code: 47

MULTIPLE_TABLE_PRIMARY_KEY_NOT_SPECIFIED

Cause: The multiple table primary key mapping must be specified when a custom multiple table join is used. If multiple tables are specified in the descriptor and the join expression is customized then the primary keys for all the tables must be specified. If the primary keys are not specified then the exception is caused.

Action: Call method `addMultipleTablePrimaryKeyName(String fieldNameInPrimaryTable, String fieldNameInSecondaryTable)` on the descriptor to set the primary keys.

Error code: 48

MULTIPLE_WRITE_MAPPINGS_FOR_FIELD

Cause: There are multiple writable mappings for the field *<fieldName>* in the descriptor. Exactly one must be defined writable. The others must be specified as read-only. When multiple write mappings are defined for the field, TopLink cannot decide on which mapping to pick up for writing the value of the field in the database row hence the exception is thrown during the validation process of descriptors. The most common site for this problem is that sometimes the field has direct-to-field mapping and also one-to-one mapping. In this case if direct-to-field mapping has to be kept then one-to-one mapping should either be read-only or it should be target foreign key reference.

Action: Make one of those mappings read only.

Error code: 49

NO_ATTRIBUTE_TRANSFORMATION_METHOD

Cause: The attribute transformation method name in the transformation mapping is not specified. This method is invoked internally by TopLink to retrieve value to store in the domain object.

Action: Define a method and set the method name on the mapping by calling method `setAttributeTransformation(String methodName)`.

Error code: 50

NO_FIELD_NAME_FOR_MAPPING

Cause: No field name is specified in direct-to-field mapping.

Action: Set the field by calling `setFieldName(String fieldName)`.

Error code: 51

NO_FOREIGN_KEYS_ARE_SPECIFIED

Cause: Neither the selection criteria nor the foreign keys were specified on one-to-one mapping. If the selection criterion is not specified then TopLink tries to build one from the foreign keys specified in the mapping.

Action: Specify the fields.

Error code: 52

NO_REFERENCE_KEY_IS_SPECIFIED

Cause: No query key named: *<queryKey>* found in: *<descriptor>*; no reference key from the target table is specified on direct collection mapping.

Action: Specify the fields by calling method `setReferenceKeyFieldName(String fieldName)`.

Error code: 53

NO_RELATION_TABLE

Cause: The relation table name is not set in this many-to-many mapping.

Action: Set relation table name by calling method `setRelationTableName(String tableName)`.

Error code: 54

NO_SOURCE_RELATION_KEYS_SPECIFIED

Cause: There are no source relation keys specified in this many-to-many mapping.

Action: Add source relation keys to the mapping.

Error code: 55

NO_SUCH_METHOD_ON_FIND_OBSOLETE_METHOD

Cause: The descriptor callback method *<selector>* on the domain class was not found. It must take a `Session` or a `DescriptorEvent` as its argument. `TopLink` tries to invoke the method using Java reflection. It is a Java exception and `TopLink` is only wrapping the main exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 56

NO_SUCH_METHOD_ON_INITIALIZING_ATTRIBUTE_METHOD

Cause: The method *<attributeMethodName>* with parameters *<databaseRow>* or *<databaseRow, session>* is not found. `TopLink` wraps the Java reflection exception

that is caused when the method is being created from the method name. This method is set by calling `setAttributeMethodName(String aMethodName)`.

Action: Inspect the internal exception and check the Java manuals.

Error code: 57

NO_SUCH_METHOD_WHILE_CONSTRUCTOR_INSTANTIATION

Cause: Inaccessible constructor. TopLink wraps the Java reflection exception that is caused when it is creating a new instance of the domain.

Action: Inspect the internal exception and check the Java manuals.

Error code: 58

NO_SUCH_METHOD_WHILE_CONVERTING_TO_METHOD

Cause: Method `<methodName>` not found with parameters `()` or `(Session)`. TopLink wraps the Java reflection exception that is caused when it is creating a Method type from the method names in transformation mapping.

Action: Inspect the internal exception and check the Java manuals.

Error code: 59

NO_SUCH_FIELD_WHILE_INITIALIZING_ATTRIBUTES_IN_INSTANCE_VARIABLE_ACCESSOR

Cause: The instance variable `<attributeName>` is not defined in the domain class or it is not accessible. TopLink wraps the Java reflection exception that is caused when it is creating a Field type from the attribute name.

Action: Inspect the internal exception and check the Java manuals.

Error code: 60

NO_SUCH_METHOD_WHILE_INITIALIZING_ATTRIBUTES_IN_METHOD_ACCESSOR

Cause: The accessor method `<setMethodName>` or `<getMethodName>` is not defined for the attribute in the domain class `<javaClassName>`, or it is not accessible. TopLink wraps the Java reflection exception that is caused when it is creating a Method type from the method name.

Action: Inspect the internal exception and check the Java manuals.

Error code: 61**NO_SUCH_METHOD_WHILE_INITIALIZING_CLASS_EXTRACTION_METHOD**

Cause: The static class extraction method *<methodName>* with *<databaseRow>* as argument does not exist, or is not accessible. Java reflection exception wrapped in TopLink exception is thrown when class extraction method is being created from the method name in inheritance policy.

Action: Inspect the internal exception and check the Java manuals.

Error code: 62**NO_SUCH_METHOD_WHILE_INITIALIZING_COPY_POLICY**

Cause: The clone method *<methodName>* with no arguments does not exist, or is not accessible. Java reflection exception wrapped in TopLink exception is thrown when a method to create clones is being created from the method name in copy policy.

Action: Inspect the internal exception and check the Java manuals.

Error code: 63**NO_SUCH_METHOD_WHILE_INITIALIZING_INSTANTIATION_POLICY**

Cause: The instance creation method *<methodName>* with no arguments does not exist, or is not accessible. Java reflection exception wrapped in TopLink exception is thrown when a method to create new instances is being created from the method name in instantiation policy.

Action: Inspect the internal exception and check the Java manuals.

Error code: 64**NO_TARGET_FOREIGN_KEYS_SPECIFIED**

Cause: The foreign keys in the target table are not specified in one-to-many mappings. These fields are not required if a selection criterion is given in the mapping but otherwise they must be specified.

Action: Set target foreign keys or selection criteria.

Error code: 65**NO_TARGET_RELATION_KEYS_SPECIFIED**

Cause: There are no target relation keys specified in many-to-many mappings.

Action: Call method `addTargetRelationKeyFieldName(String targetRelationKeyFieldName, String targetPrimaryKeyFieldName)` to set the fields.

Error code: 66

NOT_DESERIALIZABLE

Cause: The object cannot be de-serialized from the byte array read from the database. The exception is thrown when serialized object mapping is converting byte array into object.

Action: Inspect the internal exception and check the Java manuals.

Error code: 67

NOT_SERIALIZABLE

Cause: The object cannot be serialized into byte array. The exception is thrown when serialized object mapping is object into byte array.

Action: Inspect the internal exception and check the Java manuals.

Error code: 68

NULL_FOR_NON_NULL_AGGREGATE

Cause: Value of aggregate in the source object *<object>* is null. Null values not allowed for aggregate mappings unless **allow null** is specified in aggregate mapping.

Action: Call method `allowNull()` on the mapping.

Error code: 69

NULL_POINTER_WHILE_GETTING_VALUE_THRU_INSTANCE_VARIABLE_ACCESSOR

Cause: An object is accessed to get the value of an instance variable through Java reflection. This exception is thrown only on some VMs.

Action: Inspect the internal exception and check the Java manuals.

Error code: 70

NULL_POINTER_WHILE_GETTING_VALUE_THRU_METHOD_ACCESSOR

Cause: The get accessor method is invoked to get the value of attribute through Java reflection. This exception is thrown only on some VMs.

Action: Inspect the internal exception and check the Java manuals.

Error code: 71

NULL_POINTER_WHILE_SETTING_VALUE_THRU_INSTANCE_VARIABLE_ACCESSOR

Cause: Null Pointer Exception is thrown while setting value of *<attributeName>* instance variable in the object to value. An object is accessed to set the value of an instance variable through Java reflection. This exception is thrown only on some VMs.

Action: Inspect the internal exception and check the Java manuals.

Error code: 72

NULL_POINTER_WHILE_SETTING_VALUE_THRU_METHOD_ACCESSOR

Cause: Null Pointer Exception is thrown while setting value through *<setMethodName>* method in the object with argument *<argument>*. The set accessor method is invoked to set the value of attribute through Java reflection. This exception is thrown only on some VMs.

Action: Inspect the internal exception and check the Java manuals.

Error code: 73

PARENT_DESCRIPTOR_NOT_SPECIFIED

Cause: Cannot find descriptor for parent class. The descriptor of a subclass has no parent descriptor.

Action: The method `setParentClass(Class parentClass)` on the subclass descriptor must be called.

Error code: 74

PRIMARY_KEY_FIELDS_NOT_SPECIFIED

Cause: The primary key fields are not set for this descriptor.

Action: Add primary key field names using method `setPrimaryKeyFieldName(String fieldName)` or `setPrimaryKeyFieldName(String fieldName)`.

Error code: 75

REFERENCE_CLASS_NOT_SPECIFIED

Cause: The reference class is not specified in the foreign reference mapping.

Action: Set reference class by calling method `setReferenceClass(Class aClass)`

Error code: 77

REFERENCE_DESCRIPTOR_IS_NOT_AGGREGATE

Cause: The referenced descriptor for `<className>` should be set to aggregate descriptor. An aggregate mapping should always reference a descriptor that is aggregate.

Action: Call method `descriptorIsAggregate()` on the referenced descriptor.

Error code: 78

REFERENCE_KEY_FIELD_NOT_PROPERLY_SPECIFIED

Cause: The table for the reference field must be the reference table. If the reference field name specified in the direct collection mapping is qualified with the table name then the table name should match the reference table name.

Action: Qualify the field with the proper name or change the reference table name.

Error code: 79

REFERENCE_TABLE_NOT_SPECIFIED

Cause: The reference table name in the direct collection mapping is not specified.

Action: Use method `setReferenceTableName(String tableName)` on the mapping to set the table name.

Error code: 80

RELATION_KEY_FIELD_NOT_PROPERLY_SPECIFIED

Cause: The table for the relation key field must be the relation table. If the source and target relation fields name specified in the many to many mapping are qualified with the table name then the table name should match the relation table name.

Action: Qualify the field with the proper name or change the relation table name.

Error code: 81

RETURN_TYPE_IN_GET_ATTRIBUTE_ACCESSOR

Cause: The method *<attributeMethodName>* specified in the transformation mapping should have a return type set in the attribute because this method is used to extract value from the database row.

Action: Check the method and make appropriate changes.

Error code: 82

SECURITY_ON_FIND_METHOD

Cause: The descriptor callback method *<selector>* with DescriptorEvent as argument is not accessible. Java throws security exception when a Method type is created from the method name using Java reflection. The method is a descriptor event callback on the domain object that takes DescriptorEvent as its parameter.

Action: Inspect the internal exception and check the Java manuals.

Error code: 83

SECURITY_ON_FIND_OBSOLETE_METHOD

Cause: The descriptor callback method *<selector>* with *<session>* as argument is not accessible. Java throws security exception when a Method type is created from the method name using Java reflection. The method is a descriptor event callback on the domain object which takes class and session as its parameters.

Action: Inspect the internal exception and check the Java manuals.

Error code: 84

SECURITY_ON_INITIALIZING_ATTRIBUTE_METHOD

Cause: Access to the method *<attributeMethodName>* with parameters *<databaseRow>* or *<databaseRow, Session>* has been denied. Java throws security exception when a Method type is created from the attribute method name using Java reflection. The attribute method specified in the transformation mapping is used to extract value from the database row and set by calling `setAttributeTransformation(String methodName)`.

Action: Inspect the internal exception and check the Java manuals.

Error code: 85

SECURITY_WHILE_CONVERTING_TO_METHOD

Cause: Method: *<methodName>* ACCESS DENIED with *<>* or *<session>* parameters. Java throws security exception when a Method type is created from the method name using Java reflection. These are the methods that will extract the field value from the domain object in the transformation mapping.

Action: Inspect the internal exception and check the Java manuals.

Error code: 86

SECURITY_WHILE_INITIALIZING_ATTRIBUTES_IN_INSTANCE_VARIABLE_ACCESSOR

Cause: Access to the instance variable, *<attributeName>* in the class *<javaClassName>* is denied. Java throws security exception when creating Field type from the given attribute name using Java reflection.

Action: Inspect the internal exception and check the Java manuals.

Error code: 87

SECURITY_WHILE_INITIALIZING_ATTRIBUTES_IN_METHOD_ACCESSOR

Cause: The methods *<setMethodName>* and *<getMethodName>* in the object *<javaClassName>* are inaccessible. Java throws security exception when creating Method type from the given attribute accessor method name using Java reflection.

Action: Inspect the internal exception and check the Java manuals.

Error code: 88

SECURITY_WHILE_INITIALIZING_CLASS_EXTRACTION_METHOD

Cause: The static class extraction method *<methodName>* with DatabaseRow as argument is not accessible. Java throws security exception when creating Method type from the given class extraction method name using Java reflection. The method is used to extract class from the database row in inheritance policy.

Action: Inspect the internal exception and check the Java manuals.

Error code: 89**SECURITY_WHILE_INITIALIZING_COPY_POLICY**

Cause: The clone method *<methodName>* with no arguments is not accessible. Java throws security exception when creating Method type from the given method name using Java reflection. This method on copy policy is used to create clones of the domain object.

Action: Inspect the internal exception and check the Java manuals.

Error code: 90**SECURITY_WHILE_INITIALIZING_INSTANTIATION_POLICY**

Cause: The instance creation method *<methodName>* with no arguments is not accessible. Java throws security exception when creating Method type from the given method name using Java reflection. This method on instantiation policy is used to create new instances of the domain object.

Action: Inspect the internal exception and check the Java manuals.

Error code: 91**SEQUENCE_NUMBER_PROPERTY_NOT_SPECIFIED**

Cause: Either the sequence field name or the sequence number name is missing. To use sequence generated ids both the sequence number name and field name properties must be set.

Action: To use sequence-generated ids, both the sequence number name and field name properties must be set.

Error code: 92**SIZE_MISMATCH_OF_FOREIGN_KEYS**

Cause: The size of the primary keys on the target table does not match the size of the foreign keys on the source in one to one mapping.

Action: Check the mapping and the reference descriptor's primary keys.

Error code: 93**TABLE_NOT_PRESENT**

Cause: The table *<tableName>* is not present in the descriptor.

Action: Check the qualified field names specified in the mappings and descriptor, if these fields are qualified with the table name then those fields should have right table.

Error code: 94

TABLE_NOT_SPECIFIED

Cause: No table is specified in the descriptor. Descriptor must have a table name defined.

Action: Call method `addTableName(String tableName)` or `setTableName(String tableName)` to set the tables on the descriptor.

Error code: 96

TARGET_FOREIGN_KEYS_SIZE_MISMATCH

Cause: The size of the foreign keys on the target table does not match the size of the source keys on the source table in one to many mapping.

Action: Check the mapping.

Error code: 97

TARGET_INVOCATION_WHILE_CLONING

Cause: Problem in cloning the object `<domainObject>` clone method. `<methodName>` triggered an exception. Java is throwing exception when clone method is invoked using Java reflection while cloning object. The clone method is specified on the copy policy that is usually invoked to create clones in unit of work.

Action: Inspect the internal exception and check the Java manuals.

Error code: 98

TARGET_INVOCATION_WHILE_EVENT_EXECUTION

Cause: The underlying descriptor callback method `<eventMethodName>` with `DescriptorEvent` as argument, throws an exception. Java is throwing exception when descriptor event method is invoked using Java reflection.

Action: Inspect the internal exception and check the Java manuals.

Error code: 99**TARGET_INVOCATION_WHILE_GETTING_VALUE_THRU_METHOD_ACCESSOR**

Cause: The method *<methodName>* on the object *<objectName>* is throwing an exception. Java is throwing exception while getting an attribute value from the object through method accessor.

Action: Inspect the internal exception and check the Java manuals.

Error code: 100**TARGET_INVOCATION_WHILE_INSTANTIATING_METHOD_BASED_PROXY**

Cause: Method has thrown an exception. Java is throwing exception while instantiating method based proxy. This happens while instantiating transformation mapping.

Action: Inspect the internal exception and check the Java manuals.

Error code: 101**TARGET_INVOCATION_WHILE_INVOKING_ATTRIBUTE_METHOD**

Cause: The underlying method throws an exception. Java is throwing exception while invoking attribute transformation method on transformation mapping. The method is invoked to extract value from the database row to set into the domain object.

Action: Inspect the internal exception and check the Java manuals.

Error code: 102**TARGET_INVOCATION_WHILE_INVOKING_FIELD_TO_METHOD**

Cause: The method *<methodName>* is throwing an exception. Java is throwing exception while invoking field transformation method on transformation mapping. The method is invoked to extract value from the domain object to set into the database row.

Action: Inspect the internal exception and check the Java manuals.

Error code: 103**TARGET_INVOCATION_WHILE_INVOKING_ROW_EXTRACTION_METHOD**

Cause: Problem in extracting class from row *<row>*, static method, *<method>* with *<databaseRow as argument>*. An exception was triggered Java is throwing exception while invoking class extraction method. The method is used to extract class type so that right kind of object can be created in inheritance model.

Action: Inspect the internal exception and check the Java manuals.

Error code: 104

TARGET_INVOCATION_WHILE_METHOD_INSTANTIATION

Cause: Problem in creating new instance. Creation method *<methodName>* caused an exception. Java is throwing exception while invoking instantiation method. The method is used to create new instances of the domain objects.

Action: Inspect the internal exception and check the Java manuals.

Error code: 105

TARGET_INVOCATION_WHILE_OBSOLETE_EVENT_EXECUTION

Cause: The underlying descriptor callback method *<eventMethodName>* with *<session>* as argument, throws an exception. Java is throwing exception while invoking descriptor event method that takes session as its parameter.

Action: Inspect the internal exception and check the Java manuals.

Error code: 106

TARGET_INVOCATION_WHILE_SETTING_VALUE_THRU_METHOD_ACCESSOR

Cause: The method *<setMethodName>* on the object is throwing an exception. Java is throwing exception while invoking set accessor method on the domain object to set an attribute value into the domain object.

Action: Inspect the internal exception and check the Java manuals.

Error code: 108

VALUE_NOT_FOUND_IN_CLASS_INDICATOR_MAPPING

Cause: The indicator value is not found in the class indicator mapping in the parent descriptor for the class.

Action: Check `addClassIndicator(Class childClass, Object typeValue)` on the inheritance policy.

Error code: 109**WRITE_LOCK_FIELD_IN_CHILD_DESCRIPTOR**

Cause: The child descriptor should not have a write lock field defined because it gets it from the parent descriptor.

Action: Check your child descriptor and remove the field.

Error code: 110**DESCRIPTOR_IS_MISSING**

Cause: The descriptor for the reference class `<className>` is missing from the mapping.

Action: Check session to see if the descriptor for the reference class was added.

Error code: 111**MULTIPLE_TABLE_PRIMARY_KEY_MUST_BE_FULLY_QUALIFIED**

Cause: Multiple table primary key field names must be fully qualified. These fields names are given on the descriptor if it has more than one table.

Action: Specify the field names with table name.

Error code: 112**ONLY_ONE_TABLE_CAN_BE_ADDED_WITH_THIS_METHOD**

Cause: Only one table can be added through this method.

Action: Use `addTableName(String tableName)` to add multiple tables to descriptor.

Error code: 113**NULL_POINTER_WHILE_CONSTRUCTOR_INSTANTIATION**

Cause: Inaccessible constructor. Java is throwing this exception while invoking a default constructor to create new instances of the domain object.

Action: Inspect the internal exception and check the Java manuals.

Error code: 114**NULL_POINTER_WHILE_METHOD_INSTANTIATION**

Cause: Problem in creating new instance <methodName> creation method is not accessible. Java is throwing an exception while calling a method to build new instance of the domain object. This method is given by the user to override the default behavior of creating new instances through class constructor.

Action: Inspect the internal exception and check the Java manuals.

Error code: 115

NO_ATTRIBUTE_VALUE_CONVERSION_TO_FIELD_VALUE_PROVIDED

Cause: The field conversion value for the attribute value <attributeValue> was not given in the object type mapping.

Action: Check the attribute value and provide a corresponding field value in the mapping.

Error code: 116

NO_FIELD_VALUE_CONVERSION_TO_ATTRIBUTE_VALUE_PROVIDED

Cause: The attribute conversion value for the <fieldValue> was not given in the object type mapping.

Action: Check the field value and provide a corresponding attribute value in the mapping.

Error code: 118

LOCK_MAPPING_CANNOT_BE_READONLY

Cause: The domain object <className> cannot have a read only mapping for the write lock fields when the version value is stored in the object.

Action: Check the mappings on write lock fields.

Error code: 119

LOCK_MAPPING_MUST_BE_READONLY

Cause: The domain object <className> should have a read only mapping for the write lock fields when the version value is stored in the cache.

Action: Check the mappings on write lock fields.

Error code: 120

CHILD_DOES_NOT_DEFINE_ABSTRACT_QUERY_KEY

Cause: The queryKey <queryKeyName> is defined in the parent descriptor but not in the child descriptor. The descriptor has not defined abstract query key.

Action: Any implementors of interface descriptor must define the query key defined by abstract query key in the interface descriptor.

Error code: 122

SET_EXISTENCE_CHECKING_NOT_UNDERSTOOD

Cause: The interface descriptor <parent> must have at least one abstract query key defined. The string given to the method `setExistenceChecking(String token)` is not understood.

Action: The string passed should be one of the following:

- Check cache
- Check database
- Assume existence
- Assume non-existence

Error code: 125

VALUE HOLDER INSTANTIATION MISMATCH

Cause: The mapping for the attribute <mapping.getAttributeName()> uses indirection and must be initialized to a new ValueHolder.

Action: Ensure the mapping uses indirection and the attribute is initialized to a new valueHolder.

Error code: 126

NO_SUB_CLASS_MATCH

Cause: No sub-class matches this class <theClass> when inheritance is in aggregate relationship mapping.

Action: Verify the sub-class and the relationship mapping.

Error code: 127

RETURN_AND_MAPPING_WITH_INDIRECTION_MISMATCH

Cause: The get method return type for the attribute `mapping.getAttributeName()` is not declared as type `ValueHolderInterface`, but the mapping is using indirection.

Action: Ensure the get method returns a `ValueHolder` or change the mapping to not use indirection.

Error code: 128

RETURN_AND_MAPPING_WITHOUT_INDIRECTION_MISMATCH

Cause: The get method return type for the attribute `mapping.getAttributeName()` is declared as type `ValueHolderInterface`, but the mapping is not using indirection.

Action: Ensure the mapping is using indirection or change the return type from `ValueHolder`

Error code: 129

PARAMETER_AND_MAPPING_WITH_INDIRECTION_MISMATCH

Cause: The set method parameter type for the attribute `mapping.getAttributeName()` is not declared as type `ValueHolderInterface`, but the mapping is using indirection.

Action: Ensure the set method parameter is declared as a `ValueHolder` or the mapping is changed to not use indirection.

Error code: 130

PARAMETER_AND_MAPPING_WITHOUT_INDIRECTION_MISMATCH

Cause: The set method parameter type for the attribute `mapping.getAttributeName()` is declared as type `ValueHolderInterface`, but the mapping is not using indirection.

Action: Ensure the mapping is changed to use indirection or the set method parameter is not declared as a `ValueHolder`.

Error code: 131

GET_METHOD_RETURN_TYPE_NOT_VALID

Cause: The get method return type for the attribute `mapping.getAttributeName()` should be declared as type `Vector` (or a type that implements `Map` or `Collection`, if using Java2).

Action: The get method return type for the attribute should be declared as type Vector (or a type that implements Map or Collection, if using Java2).

Error code: 133

SET_METHOD_PARAMETER_TYPE_NOT_VALID

Cause: The set method parameter type for the attribute `mapping.getAttributeName()` should be declared as type Vector (or a type that implements Map or Collection, if using Java2).

Action: The set method parameter type for the attribute should be declared as type Vector (or a type that implements Map or Collection, if using Java2).

Error code: 135

ILLEGAL_TABLE_NAME_IN_MULTIPLE_TABLE_FOREIGN_KEY

Cause: The table in the multiple table foreign key relationship refers to an unknown table.

Action: Verify the table name.

Error code: 138

ATTRIBUTE_AND_MAPPING_WITH_TRANSPARENT_INDIRECTION_MISMATCH

Cause: The attribute `mapping.getAttributeName()` is not declared as a super-type of `validTypeName`, but the mapping is using transparent indirection.

Action: Verify the attribute's type and the mapping setup.

Error code: 139

RETURN_AND_MAPPING_WITH_TRANSPARENT_INDIRECTION_MISMATCH

Cause: The get method return type for the attribute `mapping.getAttributeName()` is not declared as a super-type of `validTypeName`, but the mapping is using transparent indirection.

Action: Verify the attribute's type and the mapping setup.

Error code: 140

PARAMETER_AND_MAPPING_WITH_TRANSPARENT_INDIRECTION_MISMATCH

Cause: The set method parameter type for the attribute `mapping.getAttributeName()` is not declared as a super-type of `validTypeName`, but the mapping is using transparent indirection.

Action: Verify the attribute's type and the mapping setup.

Error code: 141

FIELD_IS_NOT_PRESENT_IN_DATABASE

Cause: Field `<fieldname>` is not present in the table `<tableName>` in the database.

Action: Verify the field name for the attribute.

Error code: 142

TABLE_IS_NOT_PRESENT_IN_DATABASE

Cause: `descriptor.getTable_name()` is not present in the database.

Action: Verify the table name for the descriptor.

Error code: 143

MULTIPLE_TABLE_INSERT_ORDER_MISMATCH

Cause: The multiple table insert order Vector specified `aDescriptor.getMultipleTableInsertOrder()`, has more/fewer tables than are specified in the descriptor `aDescriptor.getTables()`. All the tables must be included in the insert order Vector.

Action: Verify that all table names for the descriptor are present and that there are no extras.

Error code: 144

INVALID_USE_OF_TRANSPARENT_INDIRECTION

Cause: Transparent Indirection can only be used with CollectionMappings.

Action: Verify the mapping. It must be a collection mapping.

Error code: 145**MISSING_INDIRECT_CONTAINER_CONSTRUCTOR**

Cause: The indirect container class must implement the constructor.

Action: Implement the constructor for the container.

Error code: 146**COULD_NOT_INSTANTIATE_INDIRECT_CONTAINER_CLASS**

Cause: The indirect container class could not be instantiated using the constructor.

Action: Validate the constructor for the indirect container class.

Error code: 147**INVALID_CONTAINER_POLICY**

Cause: This container policy should only be used in JDK1.1: containerPolicy. It was instantiated for javaClass.

Action: Validate the container policy being used.

Error code: 148**INVALID_CONTAINER_POLICY_WITH_TRANSPARENT_INDIRECTION**

Cause: The container policy is incompatible with transparent indirection.

Action: Change the container policy to be compatible with transparent indirection or do not use transparent indirection.

Error code: 149**INVALID_USE_OF_NO_INDIRECTION**

Cause: No Indirection should not receive this message.

Action: Change to use no indirection.

Error code: 150**INDIRECT_CONTAINER_INSTANTIATION_MISMATCH**

Cause: The mapping for the attribute `mapping.getAttributeName()` uses transparent indirection and must be initialized to an appropriate container.

Action: Initialize the mapping to an appropriate container.

Error code: 151

INVALID_MAPPING_OPERATION

Cause: Invalid mapping operation.

Action: Check the documentation for valid mapping operations.

Error code: 152

INVALID_INDIRECTION_POLICY_OPERATION

Cause: Invalid indirection policy operation.

Action: Check the documentation for valid indirection policy operations.

Error code: 153

REFERENCE_DESCRIPTOR_IS_NOT_AGGREGATECOLLECTION

Cause: The reference descriptor for <className> should be set to aggregate collection descriptor.

Action: Set the reference descriptor to an aggregate collection descriptor.

Error code: 154

INVALID_INDIRECTION_CONTAINER_CLASS

Cause: Invalid indirection container class.

Action: Verify the container class.

Error code: 155

MISSING_FOREIGN_KEY_TRANSLATION

Cause: The mapping does not include a foreign key field linked to the primary key field.

Action: Link the foreign key to the appropriate primary key.

Error code: 156

STRUCTURE_NAME_NOT_SET_IN_MAPPING

Cause: The structure name is not set.

Action: Set the structure name appropriately.

Error code: 157

NORMAL_DESCRIPTOR_DO_NOT_SUPPORT_NON_RELATIONAL_EXTENSIONS

Cause: Normal descriptors do not support non-relational extensions.

Action: Contact Technical Support.

Error code: 158

PARENT_CLASS_IS_SELF

Cause: The descriptor's parent class has been set to itself.

Action: Contact Technical Support.

Error code: 159

PROXY_INDIRECTION_NOT_AVAILABLE

Cause: An attempt to use proxy indirection has been made but JDK 1.3 is not being used.

Action: Proxy indirection is only supported in JDK 1.3

Error code: 160

INVALID_ATTRIBUTE_TYPE_FOR_PROXY_INDIRECTION

Cause: The attribute wasn't specified in the list of interfaces given to use Proxy Indirection.

Action: Verify the attribute

Error code: 161

INVALID_GET_RETURN_TYPE_FOR_PROXY_INDIRECTION

Cause: The return type for the indirection policy is invalid for the indirection policy.

Action: Verify that the parameter type of the attribute's get method is correct for the indirection policy.

Error code: 162

INVALID_SET_PARAMETER_TYPE_FOR_PROXY_INDIRECTION

Cause: The parameter for the set method is incorrect for the indirection type.

Action: Verify that the parameter type of the attribute's set method is correct for the indirection policy.

Error code: 163

INCORRECT_COLLECTION_POLICY

Cause: The container policy is invalid for the collection type.

Action: Check that the container policy is correct for the collection type.

Error code: 164

INVALID_AMENDMENT_METHOD

Cause: The amendment method provided is invalid, not public, or cannot be found.

Action: Ensure the amendment method is public, static, returns void and has a single argument: Descriptor.

Error code: 165

ERROR_OCCURRED_IN_AMENDMENT_METHOD

Cause: The specified amendment method threw an exception.

Action: Examine the returned exception for further details.

Error code: 166

VARIABLE_ONE_TO_ONE_MAPPING_IS_NOT_DEFINED

Cause: There is no mapping for attribute.

Action: Validate the mapping and attribute.

Error code: 168

TARGET_INVOCATION_WHILE_CONSTRUCTOR_INSTANTIATION

Cause: Constructor is missing.

Action: Make the required constructor.

Error code: 169**TARGET_INVOCATION_WHILE_CONSTRUCTOR_INSTANTIATION_OF_FACTORY**

Cause: Constructor is missing.

Action: Make the required constructor.

Error code: 170**ILLEGAL_ACCESS_WHILE_CONSTRUCTOR_INSTANTIATION_OF_FACTORY**

Cause: Permissions do not permit access to the constructor.

Action: Adjust the Java security permissions to permit access to the constructor.

Error code: 171**INSTANTIATION_WHILE_CONSTRUCTOR_INSTANTIATION_OF_FACTORY**

Cause: An instantiation failed inside the associated constructor.

Action: Determine which objects are being instantiated, and ensure all are being done properly.

Error code: 172**NO_SUCH_METHOD_WHILE_CONSTRUCTOR_INSTANTIATION_OF_FACTORY**

Cause: A message send invoked from inside the constructor is invalid because the method does not exist.

Action: Correct the message send ensuring that the message exists.

Error code: 173**NULL_POINTER_WHILE_CONSTRUCTOR_INSTANTIATION_OF_FACTORY**

Cause: A message is being sent from inside a constructor to a null object.

Action: Avoid sending a message to an object that is null.

Error code: 174

ILLEGAL_ACCESS_WHILE_METHOD_INSTANTIATION_OF_FACTORY

Cause: A message is being sent to an object from inside a factory instantiation. Java has determined this message to be illegal.

Action: Determine why the message send is illegal, and replace the message with the proper legal one.

Error code: 175

TARGET_INVOCATION_WHILE_METHOD_INSTANTIATION_OF_FACTORY

Cause: Error inside the factory associated with the invocation of a target.

Action: Determine the faulty target, and replace with the correct target, or proper message send.

Error code: 176

NULL_POINTER_WHILE_METHOD_INSTANTIATION_OF_FACTORY

Cause: A message is being sent to null inside a factory instantiation.

Action: Avoid sending a message to null.

Builder Exceptions (1001 – 1042)

Error code: 1001

No such method

Cause: Tokens in the builder generated files are the subsets of all the tokens a Project Reader can understand. Each token has a related public method on TopLink. The exception would mean that the method name is incorrect.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1002

Could not find post load method *<methodName>* on class *<aClass>*

Cause: The post load method defined in descriptor properties is not defined on the related domain class.

Action: Must define the method on the specified class.

Error code: 1003

Cannot write parameter *<object>* of class *<type>*

Cause: While creating project class the parameter tokens are read from the file and are converted to actual types before sending them to the methods. An unknown type will cause this exception.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1004

Could not access method *<method>*

Cause: Java is throwing an illegal access reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals. If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1005

Invoking *<applyResultMethod>* raised exception *<exception>*

Cause: Java is throwing an invocation reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals. If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1006

Invalid arguments invoking: <applyResultMethod> with <receiver>

Cause: Java is throwing an invalid argument reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals. If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1007

Could not access <applyResultMethod> with <receiver>

Cause: Java is throwing reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals. If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1008

Parameter mismatch <method>; received <size> parameters

Cause: The number of parameters for the token read from the project or descriptor file do not match the number of parameters a related method can take.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1009

Accessing <methodName> on <className> with <parameters>

Cause: Java is throwing reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals. If the project files are not manually edited and corrupted then this is usually an

internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1010

Could not find section definition *<section>* when building section definitions for *<target>*

Cause: Invalid section name was found in the project or descriptor file.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1011

Could not convert *<object>* into an accessible Java class.

Cause: The parameter read from the file cannot be converted to a appropriate type.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1012

File not found

Cause: The project or descriptor file was not found.

Action: Check that the path was given correctly in a project reader and also the path is correct in project file.

Error code: 1013

Invalid class/method name format.

Cause: No one should actually be using the URL way of reading INI files. This is untested and undocumented feature.

Action: Use other ways of reading files.

Error code: 1015

Open failed for URL <url>

Cause: Open failed for URL.

Action: Inspect the internal exception and check the Java manuals.

Error code: 1016

Could not resolve INIFile location: <sourceString> using search paths <searchPaths>

Cause: The file was not found on the given search paths.

Action: Check your search paths.

Error code: 1017

Invoking <method> on <receiver>

Cause: Java is throwing reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals. If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1018

Invoking <method> on <receiver>

Cause: Java is throwing reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals. If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1019

Invalid character value; expecting \$* format

Cause: An invalid character format was written to the file.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1020

Unexpected character: {

Cause: Unexpected character { found while reading vector values from the file.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1021

Unexpected character: }

Cause: Unexpected character } found while reading vector values from the file.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1022

Expecting object, found token <nextToken>

Cause: Unexpected token found while reading from the file.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1023

Unexpected word

Cause: Unexpected token found while reading from the file.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical

Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1024

setExistenceChecking <token>; not understood

Cause: Existence checking string specified on the descriptor is not understood.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1025

Class <className> not found

Cause: Java is throwing reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals. If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1026

Not enough INI elements. Found <count>.

Cause: If the line in an INI file is incomplete, i.e., it does not have enough tokens.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1027

Too many INI elements. Found <count>.

Cause: If the line in an INI file has more tokens than needed.

Action: If the project files are not manually edited and corrupted then this is usually an internal exception to TopLink and must be reported to Technical

Support. But if the file was manually edited or corrupted then the files must be generated again.

Error code: 1028

Error writing <writeString>

Cause: Could not write into the file. Perhaps / are used and file or directory structure does not exist.

Action: Inspect the internal exception and check the Java manuals. Try using \\.

Error code: 1029

Illegal access exception

Cause: Java is throwing reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 1030

Invocation target exception

Cause: Java is throwing reflection exception while invoking the method on the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 1031

Attempting to instantiate <className> with default constructor.

Cause: Java is throwing reflection exception while instantiating the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 1032

Attempting to instantiate <className> with default constructor.

Cause: Java is throwing reflection exception while instantiating the object. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 1033

IO Exception in next token

Cause: Java is throwing reflection. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 1034

IOException on close.

Cause: Java is throwing reflection. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 1035

Invalid INI(URL) Method: <method>. Should return a string.

Cause: No one should actually be using the URL way of reading INI files. This is untested and undocumented feature.

Action: Use other ways of reading files.

Error code: 1036

Could not cast using <castString>.

Cause: An error occurred during an attempt to cast using the `castString`

Action: Validate the `castString`

Error code: 1037

A writer or a target file name must be specified

Cause: A writer or a target file name is not specified.

Action: A writer or a target file name must be specified.

Error code: 1039

IOException on open.

Cause: Java is throwing reflection. TopLink only wraps that exception.

Action: Inspect the internal exception and check the Java manuals.

Error code: 1040

Post Load Method Not Static

Cause: The method specified is not static.

Action: Modify the method to be static.

Error code: 1041

Project Not Found.

Cause: No projects were found in the specified directory.

Action: Verify the directory.

Error code: 1042

Multiple Projects With Name.

Cause: More than one project with the same name was found.

Action: Verify the project name.

Concurrency Exceptions (2001 – 2004)

Error code: 2001

WAIT_WAS_INTERRUPTED

Cause: In a multi threaded environment one of the waiting thread was interrupted.

Action: Usually such exceptions would mean restarting the application but it is totally dependent on the application.

Error code: 2002

WAIT_FAILURE_SERVER

Cause: Wait failure on ServerSession. When the number of non-pooled connections reach the threshold any more requests for such connection results in wait until some one releases the connection resource. If this wait was interrupted then an exception is thrown.

Action: Usually such exceptions would mean restarting the application but it is totally dependent on the application.

Error code: 2003

WAIT_FAILURE_CLIENT

Cause: Wait failure on ClientSession. When the number of pooled connections reach the threshold any more requests for such connection results in wait until some one releases the connection resource. If this wait was interrupted then an exception is thrown.

Action: Usually such exceptions would mean restarting the application but it is totally dependent on the application.

Error code: 2004

SIGNAL_ATTEMPTED_BEFORE_WAIT

Cause: A signal was attempted before wait on concurrency manager. This normally means that an attempt was made to commit or rollback a transaction before it was started, or to rollback a transaction twice.

Action: Check transactions in the application.

Conversion Exceptions (3001 – 3007)

Error code: 3001

COULD_NOT_BE_CONVERTED

Cause: The object *<object>* of class *<objectClass>* could not be converted to *<javaClass>*. The object cannot be converted to given type.

Action: Check that the object being converted is of right type or not.

Error code: 3003

INCORRECT_DATE_FORMAT

Cause: The date in *<dateString>* is in an incorrect format. Expected format is YYYY-MM-DD.

Action: Check the date format.

Error code: 3004

INCORRECT_TIME_FORMAT

Cause: The time in *<timeString>* is in an incorrect format. Expected format is HH:MM:SS.

Action: Check the time format.

Error code: 3005

INCORRECT_TIMESTAMP_FORMAT

Cause: The timestamp *<timestampString>* is in an incorrect format. Expected format is **YYYY-MM-DD HH:MM:SS.NNNNNNNNNN**.

Action: Check the timestamp format.

Error code: 3006

COULD_NOT_CONVERT_TO_BYTE_ARRAY

Cause: The String object must be of even length to be converted to a ByteArray. This object could not be converted to a ByteArray

Action: Check the object being converted.

Error code: 3007

COULD_NOT_BE_CONVERTED_TO_CLASS

Cause: The object *<object>* of class *<objectClass>* could not be converted to *<javaClass>*. The class *<javaClass>* is not on the CLASSPATH.

Action: Check that the class *<javaClass>* is on the CLASSPATH.

Database Exceptions (4001 – 4018)

Error code: 4002

SQL_EXCEPTION

Cause: An SQL exception was encountered, thrown by the underlying JDBC bridge. TopLink only wraps that exception.

Action: One must inspect the internal exception thrown.

Error code: 4003

CONFIGURATION_ERROR_CLASS_NOT_FOUND

Cause: The driver class name was not found.

Action: Check the class name given in JDBCLogin.

Error code: 4005

DATABASE_ACCESSOR_NOT_CONNECTED

Cause: Session is not connected to the database while doing reading or writing on the database.

Action: An application may have to login again because the connection to the database might have been lost.

Error code: 4006

ERROR_READING_BLOB_DATA

Cause: Error reading blob data from the database. There are two possibilities for this exception, first is that the blob data was not read properly from the result set or the TopLink could not process the blob data using `ByteArrayOutputStream`.

Action: Check if the underlying driver support blobs properly or not and if it does then report this problem to Technical Support.

Error code: 4007

OULD_NOT_CONVERT_OBJECT_TYPE

Cause: Could not convert object type on internal error. `java.sql.TYPES = <type>`. The object from the result set cannot be converted to the type returned from the metadata information.

Action: Check if the underlying driver support the conversion type properly or not and if it does then report this problem to Technical Support.

Error code: 4008

LOGOUT_WHILE_TRANSACTION_IN_PROGRESS

Cause: An attempt has been made to logout while the transaction is still in progress. You cannot logout while a transaction is in progress.

Action: Wait until the transaction is over.

Error code: 4009

SEQUENCE_TABLE_INFORMATION_NOT_COMPLETE

Cause: The sequence information given to TopLink is not sufficiently complete to get the set of sequence numbers from the database. This usually happens on native sequencing on oracle database.

Action: Check the data given specially the sequence name given in TopLink.

Error code: 4011

ERROR_PREALLOCATING_SEQUENCE_NUMBERS

Cause: Error preallocating sequence numbers on the database; the sequence table information is not complete.

Action: Check if the sequence table was created on the database properly or not.

Error code: 4014

CANNOT_REGISTER_SYNCHRONIZATIONLISTENER_FOR_UNITOFWORK

Cause: Cannot register Synchronization Listener: *underlying_exception_string*. When the TopLink Session is configured with an `ExternalTransactionController`, any unit of work requested by a client must operate within the context of a JTS external global transaction. When a unit of work is created and the external global transaction is not in existence or if the system cannot acquire a reference to it, this error is reported.

Action: Ensure that a JTS transaction is in progress before acquiring the unit of work.

Error code: 4015

SYNCHRONIZED_UNITOFWORK_DOES_NOT_SUPPORT_COMMITANDRESUME

Cause: Synchronized `UnitOfWork` does not support the `commitAndResume` operation. When the TopLink Session is configured with an `ExternalTransactionController`, any unit of work requested by a client must operate within the context of a JTS external global transaction (see [Error code: 4014](#)). The JTS specification does not support the concept of "checkpointing" a transaction, that is, committing the work done and then continuing to work within the same transaction context (JTS does not support nested transactions, either). Thus, if client code invokes `commitAndResume()` on a "synchronized" unit of work, this error will be reported.

Action: None.

Error code: 4016

CONFIGURATION_ERROR_NEW_INSTANCE_INSTANTIATION_EXCEPTION

Cause: Configuration error. Attempting to instantiate Driver: <javaClass>. Could not instantiate driver.

Action: Check the driver.

Error code: 4017

CONFIGURATION_ERROR_NEW_INSTANCE_ILLEGAL_ACCESS_EXCEPTION

Cause: A configuration error occurred while attempting to instantiate Driver: <javaClass>. Could not instantiate driver.

Action: Check the driver.

Error code: 4018

TRANSACTION_MANAGER_NOT_SET_FOR_JTS_DRIVER

Cause: The transaction manager has not been set for the JTSSynchronizationListener.

Action: Set a transaction manager for the JTSSynchronizationListener.

Optimistic Lock Exceptions (5001 – 5007)

Error code: 5001

NO_VERSION_NUMBER_WHEN_DELETING

Cause: An attempt was made to delete the object <object> but it has no version number in the identity map. This object either was never read or has already been deleted.

Action: Logging SQL is very helpful in understanding the reason why the exception is thrown. The last delete will show the object we are deleting that throws an exception.

Error code: 5003

OBJECT_CHANGED_SINCE_LAST_READ_WHEN_DELETING

Cause: The object state has changed in the database. The object *<object>* cannot be deleted because it has changed or been deleted since it was last read. This usually means that the row in the table was changed by some other application.

Action: Refreshing an object will refresh the object with the new data from the database.

Error code: 5004

NO_VERSION_NUMBER_WHEN_UPDATING

Cause: An attempt has been made to update the object *<object>* but it has no version number in the identity map. It may not have been read before being updated or has been deleted.

Action: Logging SQL is very helpful in understanding the reason why the exception is thrown. The last update will show the object we are updating that throws an exception.

Error code: 5006

OBJECT_CHANGED_SINCE_LAST_READ_WHEN_UPDATING

Cause: The object state has changed in the database. The object *<object>* cannot be updated because it has changed or been deleted since it was last read. This usually means that the row in the table was changed by some other application.

Action: Refreshing an object will refresh the object with the new data from the database.

Error code: 5007

MUST_HAVE_MAPPING_WHEN_IN_OBJECT

Cause: The object *<aClass>* must have a non-read-only mapping corresponding to the version lock field. The mapping was not defined for the locking field. It is needed when the lock value is stored in the domain object than in a cache.

Action: Define a mapping for the field.

Error code: 5008

NEED_TO_MAP_JAVA_SQL_TIMESTAMP

Cause: A write lock value which is stored in a domain object is not an instance of `java.sql.Timestamp`.

Action: Change the value of the attribute to be an instance of `java.sql.Timestamp`.

Query Exceptions (6001 – 6092)

Error code: 6001

ADDITIONAL_SIZE_QUERY_NOT_SPECIFIED

Cause: Cursored SQL queries must provide an additional query to retrieve the size of the result set. Custom queries using cursor streams and not specifying the size query will cause this exception. The size query is used to retrieve the size of the number of object read from the database. This is done on JDBC 1.0 compatible drivers because it provides no way to extract the size of the result set read. Non custom queries using cursor streams will have default queries generated by TopLink.

Action: Specify a size query.

Error code: 6002

AGGREGATE_OBJECT_CANNOT_BE_DELETED

Cause: Aggregated objects cannot be written or deleted independent of their owners. There is no identity maintained on such objects.

Action: Application's discretion.

Error code: 6003

ARGUMENT_SIZE_MISMATCH_IN_QUERY_AND_QUERY_DEFINITION

Cause: The number of arguments provided to the query for execution does not match the number of arguments provided with the query definition.

Action: Check the query and query execution.

Error code: 6004

BACKUP_CLONE_IS_ORIGINAL_FROM_PARENT

Cause: The object `<clone>` of class `<clone.getClass()>` with identity hashcode `(System.identityHashCode())` `<System.identityHashCode(clone)>` is not from this unit of work space but from the parent session. The object was never registered in this unit of work but

read from the parent session and related to an object registered in the unit of work.

Action: Ensure that you are correctly registering your objects. If you are still having problems, you can use the `UnitOfWork.validateObjectSpace()` method to help debug where the error occurred. Please see the manual and FAQ for more information.

Error code: 6005

BACKUP_CLONE_IS_ORIGINAL_FROM_SELF

Cause: The object clone of class `<clone.getClass()>` with identity hashcode `(System.identityHashCode()) <System.identityHashCode(clone)>` is the original to a registered new object. The unit of work clones registered new objects, so you must ensure that it is registered before it is reference by another object. If you do not want the new object to be cloned, used the `UnitOfWork.registerNewObject(Object)` API.

Action: Ensure that you are correctly registering your objects. If you are still having problems, you can use the `UnitOfWork.validateObjectSpace()` method to help debug where the error occurred. Please see the manual and FAQ for more information.

Error code: 6006

BATCH_READING_NOT_SUPPORTED

Cause: This mapping does not support batch reading. The optimization of batching the read of all the target rows is not supported for the mapping.

Action: The problem is a TopLink development problem and user should never encounter this unless the mapping is a new custom mapping. Contact Technical Support.

Error code: 6007

DESCRIPTOR_IS_MISSING

Cause: Missing descriptor for `<reference Class>`. The descriptor related to the class or the object is not found in the session.

Action: Check if the related descriptor was added to the session or the query is done on the right object or class.

Error code: 6008

DESCRIPTOR_IS_MISSING_FOR_NAMED_QUERY

Cause: Missing descriptor for <domain Class Name> for query named <queryName>. The descriptor where named query is defined is not added to the session.

Action: Check if the related descriptor was added to the session or the query is done on the right class.

Error code: 6013

INCORRECT_SIZE_QUERY_FOR_CURSOR_STREAM

Cause: The size query given on the queries returning cursor streams is not correct. The execution of the size query did not return any size.

Action: If the cursor stream query was custom query than check the size query specified else report this problem to Technical Support.

Error code: 6014

INVALID_QUERY

Cause: Objects cannot be written in unit of work using modify queries, they must be registered.

Action: Objects are registered in the unit of work and during commit unit of work will do the required changes to the database.

Error code: 6015

INVALID_QUERY_KEY_IN_EXPRESSION

Cause: The query key <key> does not exist. Usually this happens because of mis-spelled query key.

Action: Check the query key specified in the expression and make sure that a query key was added to the descriptor.

Error code: 6016

INVALID_QUERY_ON_SERVER_SESSION

Cause: Objects and the database cannot be changed through the server session; all changes must be done through a client session's unit of work. The objects cannot be changed on the server session by modifying queries. Objects are changed in the client sessions acquired from this server session.

Action: Use client session's unit of work to change the object.

Error code: 6020

NO_CONCRETE_CLASS_INDICATED

Cause: No concrete class indicated for the type in this row. The type indicator read from the database row has no entry in the type indicator hashtable or if class extraction method was used, it did not return any concrete class type. The exception is thrown when sub classes are being read.

Action: Check class extraction method if specified or check the descriptor to see if all the type indicator values were specified or not.

Error code: 6021

NO_CURSOR_SUPPORT

Cause: No cursor support provided for abstract class multiple table descriptors using expressions.

Action: Consider using custom SQL or multiple queries. <query>

Error code: 6023

OBJECT_TO_INSERT_IS_EMPTY

Cause: There are no fields to be insert into the table. The fields to insert into the table, <table> is empty. You must define at least one mapping for this table

Action: Must define at least one mapping for this table.

Error code: 6024

OBJECT_TO_MODIFY_NOT_SPECIFIED

Cause: An object to modify is required for a modify query.

Action: Ensure the query contains an object before executing.

Error code: 6026

QUERY_NOT_DEFINED

Cause: Query is not defined. When executing query on the session the parameter which takes query is null.

Action: Check if the query is passed properly.

Error code: 6027

QUERY_SENT_TO_INACTIVE_UNIT_OF_WORK

Cause: The unit of work has been released and is now inactive.

Action: The unit of work, once released, cannot be reused unless `commitAndResume` is called.

Error code: 6028

READ_BEYOND_QUERY

Cause: An attempt has been made to read from the cursor streams beyond its limits (beyond the end of the stream).

Action: Ensure the stream is being checked for an "end of stream" condition before attempting to retrieve more objects.

Error code: 6029

REFERENCE_CLASS_MISSING

Cause: The reference class in the query is not specified. A reference class must be provided.

Action: Check the query.

Error code: 6030

REFRESH_NOT_POSSIBLE_WITHOUT_CACHE

Cause: Refresh is not possible if caching is not set. The read queries which skip cache to read objects cannot be used to do refresh the object. Refreshing is not possible without identity.

Action: Check query.

Error code: 6031

SIZE_ONLY_SUPPORTED_ON_EXPRESSION_QUERIES

Cause: Did not find size query. Size only supported on expression queries, unless a size query is given.

Action: Cursor streams on custom query should also define size query.

Error code: 6032**SQL_STATEMENT_NOT_SET_PROPERLY**

Cause: The SQL statement has not been properly set. The user should never encounter this unless queries have been customized.

Action: Contact Technical Support.

Error code: 6034**INVALID_QUERY_ITEM**

Cause: Invalid query item expression.

Action: Validate the Expression being used.

Error code: 6041**SELECTION_OBJECT_CANNOT_BE_NULL**

Cause: The selection object passed to a read object or refresh was null.

Action: Check `setSelectionObject()` on the read query.

Error code: 6042**UNNAMED_QUERY_ON_SESSION_BROKER**

Cause: Data read and data modify queries is being executed without the session name. Only object-level queries can be directly executed by the session broker, unless the query is named.

Action: Session broker has no idea about such queries about which session to execute the query on. Specify the session name.

Error code: 6043**REPORT_RESULT_WITHOUT_PKs**

Cause: ReportQuery without PKs cannot readObject. The report query result returned is without primary key values. An object from the result can only be created only if primary keys were also read.

Action: Check documentation on `retrievePriamryKeys()` on report query.

Error code: 6044**NULL_PRIMARY_KEY_IN_BUILDING_OBJECT**

Cause: The primary key read from the row, *<databaseRow>* during the execution of the query was detected to be null; primary keys must not contain null.

Action: Check query and also the table on the database.

Error code: 6045

NO_DESCRIPTOR_FOR_SUBCLASS

Cause: The subclass has no descriptor defined for it.

Action: Check session if the descriptor was added or not or else check class extraction method.

Error code: 6046

CANNOT_DELETE_READ_ONLY_OBJECT

Cause: Cannot delete an object of read-only class. The class *<className>* is declared read-only in this unit of work. Thrown in unit of work when trying to delete an object that is marked as read only.

Action: Contact Technical Support.

Error code: 6047

INVALID_OPERATOR

Cause: The operator *<data>* used in the expression is not valid.

Action: Check `ExpressionOperator` class to see a list of all of the operators supported.

Error code: 6048

ILLEGAL_USE_OF_GETFIELD

Cause: Illegal use of `getField` *<data>* in expression. This is a TopLink development exception that users should not encounter.

Action: Report this problem to Technical Support.

Error code: 6049

ILLEGAL_USE_OF_GETTABLE

Cause: Illegal use of `getTable` *<data>* in expression. This is a TopLink development exception that users should not encounter.

Action: Report this problem to Technical Support.

Error code: 6050

REPORT_QUERY_RESULT_SIZE_MISMATCH

Cause: The number of attributes requested does not match the attributes returned from the database in report query. This usually happens if one gives custom query on the report query.

Action: Check the custom query if it is specified or report this problem to Technical Support.

Error code: 6051

CANNOT_CACHE_PARTIAL_OBJECT

Cause: Partial Objects are never put in the cache. Partial object queries are not allowed to maintain the cache or be edited. You must set `dontMaintainCache()`.

Action: Call the `dontMaintainCache()` method before executing the query.

Error code: 6052

OUTER_JOIN_ONLY_VALID_FOR_ONE_TO_ONE

Cause: An outer join (`getAllowingNull`) is only valid for one to one mappings, and cannot be used for the mapping.

Action: Do not attempt to use `getAllowingNull` for mappings other than one to one.

Error code: 6054

CANNOT_ADD_TO_CONTAINER

Cause: Cannot add `<anObject>` to a `<containerClass>` using `<policy>`. This is TopLink development exception and user should never encounter this problem unless a custom container policy has been written.

Action: Contact Technical Support.

Error code: 6055

METHOD_INVOCATION_FAILED

Cause: The method invocation of *<aMethod>* on the object *<anObject>* threw a Java reflection exception while accessing method.

Action: Check Java documentation on the internal exception.

Error code: 6056

CANNOT_CREATE_CLONE

Cause: Cannot create a clone of *<anObject>* using *<policy>*. This is a TopLink development exception and user should never encounter this problem unless a custom container policy has been written.

Action: Report this problem to Technical Support.

Error code: 6057

METHOD_NOT_VALID

Cause: The method *<methodName>* is not valid to call on object *<aReceiver>*. This is a TopLink development exception and user should never encounter this problem unless a custom container policy has been written.

Action: Contact Technical Support.

Error code: 6058

METHOD_DOES_NOT_EXIST_IN_CONTAINER_CLASS

Cause: The method named *<methodName>* was not found in class *<aClass>*. Thrown when looking for clone method on the container class. The clone is needed to create clones of the container in unit of work.

Action: Define clone method on the container class.

Error code: 6059

COULD_NOT_INSTANTIATE_CONTAINER_CLASS

Cause: The class *<aClass>* cannot be used as the container for the results of a query since it cannot be instantiated. Thrown when new interface container policy is being created using Java reflection.

Action: Check Java documentation on the internal exception.

Error code: 6060

MAP_KEY_NOT_COMPARABLE

Cause: Could not use object *<anObject>* of type *<objectClass>* as a key into *<aContainer>* which is of type *<containerClass>*. Key cannot be compared with the keys currently in the map. Throws Java reflection exception while accessing method.

Action: Check Java documentation on the internal exception.

Error code: 6061

CANNOT_ACCESS_METHOD_ON_OBJECT

Cause: Cannot reflectively access the method *<aMethod>* for object: *<anObject>* of type *<anObjectClass>*. Throws Java reflection exception while accessing method.

Action: Check Java documentation on the internal exception.

Error code: 6062

CALLED_METHOD_THREW_EXCEPTION

Cause: The method *<aMethod>* was called reflectively on object *<object>* of class *<objectClass>* and threw an exception. Throws Java reflection exception while accessing method.

Action: Check Java documentation on the internal exception.

Error code: 6063

INVALID_OPERATION

Cause: Invalid operation *<operation>* on cursor. The operation is not supported.

Action: Check the class documentation and look for the corresponding method that should be used.

Error code: 6064

CANNOT_REMOVE_FROM_CONTAINER

Cause: Cannot remove *<anObject>* of type *<anObjectClass>* from *<aContainerClass>* using *<policy>*. This is TopLink development exception and user should never encounter this problem unless a custom container policy has been written.

Action: Contact Technical Support.

Error code: 6065

CANNOT_ADD_ELEMENT

Cause: Cannot add element to the collection container policy (cannot add *<anObject>* of type *<anObjectClass>* to a *<aContainerClass>*).

Action: Check Java documentation on the internal exception.

Error code: 6066

BACKUP_CLONE_DELETED

Cause: Deleted objects cannot have reference after being deleted. The object clone of class *<clone.getClass()>* with identity hashcode (*System.identityHashCode()*) *<System.identityHashCode(clone)>* has been deleted, but still has references.

Action: Ensure that you are correctly registering your objects. If you are still having problems, you can use the *UnitOfWork.validateObjectSpace()* method to help debug where the error occurred. Please see the manual and FAQ for more information.

Error code: 6068

CANNOT_COMPARE_TABLES_IN_EXPRESSION

Cause: Cannot compare table reference to *<data>* in expression

Action: Check the expression

Error code: 6069

INVALID_TABLE_FOR_FIELD_IN_EXPRESSION

Cause: Field has invalid table in this context for field *<data>* in expression.

Action: Check the expression

Error code: 6070

INVALID_USE_OF_TO_MANY_QUERY_KEY_IN_EXPRESSION

Cause: Invalid use of a query key representing a to-many relationship *<data>* in expression. Use *anyOf* rather than *get*.

Action: Use the *anyOf* operator instead of *get*.

Error code: 6071**INVALID_USE_OF_ANY_OF_IN_EXPRESSION**

Cause: Invalid use of `anyOf` for a query key not representing a to-many relationship *<data>* in expression. Use `get` rather than `anyOf`.

Action: Use the `get` operator instead of `anyOf`.

Error code: 6072**CANNOT_QUERY_ACROSS_VARIABLE_ONE_TO_ONE_MAPPING**

Cause: Querying across a variable one-to-one mapping is not supported.

Action: Change the expression such that the query is not done across a variable one to one mapping.

Error code: 6073**ILL_FORMED_EXPRESSION**

Cause: Ill-formed expression in query, attempting to print an object reference into a SQL statement for *<queryKey>*.

Action: Contact Technical Support.

Error code: 6074**CANNOT_CONFORM_EXPRESSION**

Cause: This expression cannot determine if the object conforms in memory, you must set the query to check the database.

Action: Change the query such that it does not attempt to conform to the results of the query.

Error code: 6075**INVALID_OPERATOR_FOR_OBJECT_EXPRESSION**

Cause: Object comparisons can only use the `equal` or `notEqual` operators, other comparisons must be done through query keys or direct attribute level comparisons.

Action: Ensure the query uses only `equal` and `notEqual` if object comparisons are being used.

Error code: 6076

UNSUPPORTED_MAPPING_FOR_OBJECT_COMPARISON

Cause: Object comparisons can only be used with 1:1 mappings; other mapping comparisons must be done through query keys or direct attribute level comparisons.

Action: Use a query key instead of attempting to compare objects across the mapping.

Error code: 6077

OBJECT_COMPARISON_CANNOT_BE_PARAMETERIZED

Cause: Object comparisons cannot be used in parameter queries.

Action: Change the query such that it does not attempt to use object when using parameterized queries.

Error code: 6078

INCORRECT_CLASS_FOR_OBJECT_COMPARISON

Cause: The class of the argument for the object comparison is incorrect.

Action: Ensure the class for the query is correct.

Error code: 6079

CANNOT_COMPARE_TARGET_FOREIGN_KEYS_TO_NULL

Cause: Object comparison cannot be used for target foreign key relationships. Query on the source primary key instead.

Action: Query on source primary key.

Error code: 6080

INVALID_DATABASE_CALL

Cause: Invalid database call – the call must be an instance of DatabaseCall: *<call>*.

Action: Ensure the call being used is a DatabaseCall.

Error code: 6081

INVALID_DATABASE_ACCESSOR

Cause: Invalid database accessor – the accessor must be an instance of DatabaseAccessor: <accessor>.

Action: Ensure the accessor being used is a DatabaseAccessor.

Error code: 6082

METHOD_DOES_NOT_EXIST_ON_EXPRESSION

Cause: Method: <methodName> with argument types: <argTypes> cannot be invoked on Expression.

Action: Ensure the method being used is a supported method.

Error code: 6083

IN_CANNOT_BE_PARAMETERIZED

Cause: Queries using IN cannot be parameterized.

Action: Either disable the query prepare or binding.

Error code: 6084

REDIRECTION_CLASS_OR_METHOD_NOT_SET

Cause: The redirection query was not configured properly, the class or method name was not set.

Action: Check the configuration for the redirection class.

Error code: 6085

REDIRECTION_METHOD_NOT_DEFINED_CORRECTLY

Cause: The redirection query's method is not defined or define with the wrong arguments. It must be public static and have arguments DatabaseQuery, DatabaseRow, Session (the interface).

Action: Check the redirection query's method as above.

Error code: 6086

REDIRECTION_METHOD_ERROR

Cause: The redirection query's method invocation threw an exception.

Action: Check the redirection method for problems.

Error code: 6087

EXAMPLE_AND_REFERENCE_OBJECT_CLASS_MISMATCH

Cause: There is a class mismatch between the example object and the reference class specified for this query.

Action: Ensure the example and reference classes are compatible.

Error code: 6088

NO_ATTRIBUTES_FOR_REPORT_QUERY

Cause: A ReportQuery has been built with no attributes specified.

Action: Specify attribute for the query.

Error code: 6089

NO_EXPRESSION_BUILDER_CLASS_FOUND

Cause: The expression has not been initialized correctly. Only a single ExpressionBuilder should be used for a query. For parallel expressions, the query class must be provided to the ExpressionBuilder constructor, and the query's ExpressionBuilder must always be on the left side of the expression.

Action: Contact Technical Support.

Error code: 6090

CANNOT_SET_REPORT_QUERY_TO_CHECK_CACHE_ONLY

Cause: Cannot set ReportQuery to **Check Cache Only**.

Action: Contact Technical Support.

Error code: 6091

TYPE_MISMATCH_BETWEEN_ATTRIBUTE_AND_CONSTANT_ON_EXPRESSION

Cause: The type of the constant, used for comparison in the expression, does not match the type of the attribute.

Action: Contact Technical Support.

Error code: 6092

MUST_INSTANTIATE_VALUEHOLDERS

Cause: Uninstantiated ValueHolder detected. You must instantiate the relevant Valueholders to perform this in-memory query.

Action: Instantiate the valueholders for the collection you want to query on.

Validation Exceptions (7001 – 7097)

Error code: 7001

LOGIN_BEFORE_ALLOCATING_CLIENT_SESSIONS

Cause: You attempted to allocate client sessions before logging into the server.

Action: Ensure you have called `login()` on your `ServerSession` or `DatabaseSession`.

Note: This error also appears in multithreaded environments as a result of concurrency issues. Check that all of your threads are synchronized.

Error code: 7002

POOL_NAME_DOES_NOT_EXIST

Cause: The pool name used while acquiring client session from the server session does not exist.

Action: Check the pool name given while acquiring client session and all the existing pools on the server session.

Error code: 7003

MAX_SIZE_LESS_THAN_MIN_SIZE

Cause: The maximum number of connections in a connection pool should be more than the minimum number of connections.

Action: Check `addConnectionPool(String poolName, JDBCLogin login, int minNumberOfConnections, int maxNumberOfConnections)` on server session.

Error code: 7004

POOLS_MUST_BE_CONFIGURED_BEFORE_LOGIN

Cause: Pools must all be added before login on the server session has been done. Once logged in pools cannot be added.

Action: Check `addConnectionPool(String poolName, JDBCLogin login, int minNumberOfConnections, int maxNumberOfConnections)` on server session. This method should be called before login on server session.

Error code: 7008

JAVA_TYPE_IS_NOT_A_VALID_DATABASE_TYPE

Cause: The Java type `<javaClass>` is not a valid database type. The Java type of the field to be written to the database has no corresponding type on the database.

Action: Check table or stored procedure definition.

Error code: 7009

MISSING_DESCRIPTOR

Cause: The descriptor `<className>` is not found in the session. Verify that the descriptor has been properly registered with the session.

Action: Check if the related descriptor to the class was added to the session.

Error code: 7010

START_INDEX_OUT_OF_RANGE

Cause: This is a TopLink development exception and user should never encounter this problem. It happens when a copy of vector is created with start and end index.

Action: Report this problem to Technical Support.

Error code: 7011

STOP_INDEX_OUT_OF_RANGE

Cause: This is a TopLink development exception and user should never encounter this problem. It happens when a copy of vector is created with start and end index.

Action: Report this problem to Technical Support.

Error code: 7012**FATAL_ERROR_OCCURRED**

Cause: This is a TopLink development exception and user should never encounter this problem. It happens when test cases are executed.

Action: Report this problem to Technical Support. This error commonly occurs if you attempt to `commit()` an invalid (or previously committed) `UnitOfWork`.

If `ValidationException.cannotCommitUOWAgain()` appears in the stack trace, ensure that call `commit()` on valid `UnitOfWork` instances.

Error code: 7013**NO_PROPERTIES_FILE_FOUND**

Cause: `TopLink.properties` file cannot be found on the system CLASSPATH.

Action: Ensure that there is a `TopLink.properties` file located on the system CLASSPATH.

Error code: 7017**CHILD_DESCRIPTOR_DO_NOT_HAVE_IDENTITY_MAP**

Cause: Child descriptors do not have an identity map; they share their parent's. An Identity map is added to the child descriptor. A child descriptor shares its parent's identity map.

Action: Check child descriptor and remove identity map from it.

Error code: 7018**FILE_ERROR**

Cause: The user should never encounter this problem. It happens when test cases are executed.

Action: Contact Technical Support.

Error code: 7023**INCORRECT_LOGIN_INSTANCE_PROVIDED**

Cause: The login instance provided to the `login()` method is incorrect. A `JDBCLogin` must be provided.

Action: It should be of type `JDBCLogin`.

Error code: 7024

INVALID_MERGE_POLICY

Cause: This is a TopLink development exception and users should never encounter it.

Action: Contact Technical Support.

Error code: 7025

ONLY_FIELDS_ARE_VALID_KEYS_FOR_DATABASE_ROWS

Cause: The key on the database row is not either of type String or of type DatabaseField.

Action: Contact Technical Support.

Error code: 7027

SEQUENCE_SETUP_INCORRECTLY

Cause: Sequence <sequenceName> is setup incorrectly, increment does not match pre-allocation size.

Action: Contact Technical Support.

Error code: 7028

WRITE_OBJECT_NOT_ALLOWED_IN_UNIT_OF_WORK

Cause:

Action: The objects in the unit of work are written to database by registering and committing the unit of work.

Error code: 7030

CANNOT_SET_READ_POOL_SIZE_AFTER_LOGIN

Cause: You cannot set read pool size after the server session has already been logged in.

Action: The size should be set before login.

Error code: 7031

CANNOT_ADD_DESCRIPTOR_TO_SESSION_BROKER

Cause: You cannot add descriptors to session broker.

Action: Descriptors are added to the sessions contained in the session broker.

Error code: 7032

NO_SESSION_REGISTERED_FOR_CLASS

Cause: The descriptor related to the domain class *<domainClass>* was not found in any of the sessions registered in the session broker.

Action: Check sessions.

Error code: 7033

NO_SESSION_REGISTERED_FOR_NAME

Cause: The session with the given name *<sessionName>* is not registered in the session broker.

Action: Check session broker.

Error code: 7038

LOG_IO_ERROR

Cause: Error while logging message to session's log.

Action: Check the internal exception.

Error code: 7039

CANNOT_REMOVE_FROM_READ_ONLY_CLASSES_IN_NESTED_UNIT_OF_WORK

Cause: Cannot remove from the set of read-only classes in a nested unit of work. A nested unit of work's set of read-only classes must be equal to or a superset of its parent's set of read-only classes.

Action: Contact Technical Support.

Error code: 7040

CANNOT_MODIFY_READ_ONLY_CLASSES_SET_AFTER_USING_UNIT_OF_WORK

Cause: Cannot change the set of read-only classes in a unit of work after that unit of work has been used. Changes to the read-only set must be made when acquiring the unit of work or immediately after.

Action: Contact Technical Support.

Error code: 7042

LATFORM_CLASS_NOT_FOUND

Cause: The platform class *<className>* was not found and a reflection exception was thrown.

Action: Check internal exception.

Error code: 7043

NO_TABLES_TO_CREATE

Cause: *<project>* does not have any tables to create on the database.

Action: Validate the project and tables to be created.

Error code: 7044

LLEGAL_CONTAINER_CLASS

Cause: The container class specified *<className>* cannot be used as the container because it does not implement Collection or Map.

Action: The container class must implement either of these two interfaces.

Error code: 7047

ONTAINER_POLICY_DOES_NOT_USE_KEYS

Cause: Invalid Map class was specified for the container policy. The container specified (of class *<aPolicyContainerClass>*) does not require keys. You tried to use *<methodName>*.

Action: Use map class that implements Map interface.

Error code: 7048

METHOD_NOT_DECLARED_IN_ITEM_CLASS

Cause: The key method on the map container policy is not defined. The instance method *<methodName>* does not exist in the reference class *<className>* and therefore cannot be used to create a key in a Map.

Action:

Error code: 7051

MISSING_MAPPING

Cause: Missing attribute *<attributeName>* for descriptor *<descriptor>* called from *<source>*. This is a TopLink development exception and a user should never encounter it.

Action: Contact Technical Support.

Error code: 7052

ILLEGAL_USE_OF_MAP_IN_DIRECTCOLLECTION

Cause: The method `useMapClass` was called on a `DirectCollectionMapping`. It is illegal to call `useMapClass()` on a `DirectCollectionMapping`. TopLink cannot instantiate Java attributes mapped using a `DirectCollectionMapping` with a `Map`. The `useMapClass()` API is supported for `OneToManyMappings` and `ManyToManyMappings`. The Java2 Collection interface is supported using the `useCollectionClass()` method.

Action: Do not call `useMapClass()` on `DirectCollectionMappings`. Instead, use the `useCollectionClass()` API.

Error code: 7053

CANNOT_RELEASE_NON_CLIENTSESSION

Cause: Release attempted on a session that is not a client session. Only client sessions may be released.

Action: Modify code so that the client session is not released.

Error code: 7054

CANNOT_ACQUIRE_CLIENTSESSION_FROM_SESSION

Cause: Acquire attempted on a session that is not a client session. Client sessions may only be acquired from server sessions.

Action: Modify code so that the acquire in not attempted.

Error code: 7055

OPTIMISTIC_LOCKING_NOT_SUPPORTED

Cause: Optimistic Locking is not supported with stored procedure generation.

Action: Do not use `OptimisticLocking` with stored procedure generation.

Error code: 7056

WRONG_OBJECT_REGISTERED

Cause: The wrong object was registered into the unit of work. It should be the object from the parent cache.

Action: Ensure the object is from the parent cache

Error code: 7058

INVALID_CONNECTOR

Cause: The connector must be of type DefaultConnector.

Action: Ensure the connector is of type DefaultConnector.

Error code: 7059

INVALID_DATA_SOURCE_NAME

Cause: Invalid DataSource name: *<name>*.

Action: Verify the data source name.

Error code: 7060

CANNOT_ACQUIRE_DATA_SOURCE

Cause: Cannot acquire DataSource: *<name>* or an error has occurred in setting up the datasource.

Action: Verify the data source name. Check the nested SQL exception to determine the cause of the error. Typical problems include:

- The connection pool was not configured in your `config.xml`.
- The driver is not on the classpath.
- The user or password is incorrect.
- The database server url or driver name is not properly specified.

Error code: 7061

JTS_EXCEPTION_RAISED

Cause: Exception occurred within JTS.

Action: Examine the JTS exception returned and consult the JTS documentation.

Error code: 7062**FIELD_LEVEL_LOCKING_NOTSUPPORTED_OUTSIDE_A_UNIT_OF_WORK**

Cause: `FieldLevelLocking` not supported outside a `unitOfWork`. In order to use field level locking, a unit of work must be used for ALL writes.

Action: Use a unit of work for writing.

Error code: 7063**EJB_CONTAINER_EXCEPTION_RAISED**

Cause: Exception occurred within EJB container.

Action: Examine the EJB exception returned and consult the JTS documentation.

Error code: 7064**EJB_PRIMARY_KEY_REFLECTION_EXCEPTION**

Cause: Exception occurred in reflective EJB bean primary key extraction.

Action: Ensure your primary key object is defined correctly.

Error code: 7065**EJB_CANNOT_LOAD_REMOTE_CLASS**

Cause: The remote class for the bean cannot be loaded/found for the bean.

Action: Ensure the correct class loader is set correctly.

Error code: 7066**EJB_MUST_BE_IN_TRANSACTION**

Cause: Cannot create or remove beans unless a JTS transaction is present, `bean=<bean>`.

Action: Verify that the JTS transaction is present.

Error code: 7068**EJB_INVALID_PROJECT_CLASS**

Cause: The platform class `<platformName>` was not found for the `<projectName>` using default class loader.

Action: Validate the project and platform.

Error code: 7069

SESSION_AMENDMENT_EXCEPTION_OCCURRED

Cause: An exception occurred looking up or invoking the session amendment method, *<amendmentMethod>* on the class *<amendmentClass>*.

Action: Validate the amendment method and class.

Error code: 7070

EJB_TOPLINK_PROPERTIES_NOT_FOUND

Cause: A *toplink.properties* resource bundle must be located on the classpath in a TopLink directory.

Action: Validate the classpath and the location of the TopLink resource bundle.

Error code: 7071

CANT_HAVE_UNBOUND_IN_OUTPUT_ARGUMENTS

Cause: You cannot use input output parameters without using binding.

Action: Use binding on the *StoredProcedureCall*.

Error code: 7073

ORACLE_OBJECT_TYPE_NOT_DEFINED

Cause: Oracle object type with type name *<typeName>* is not defined.

Action: Verify that the Oracle object type is defined.

Error code: 7074

ORACLE_OBJECT_TYPE_NAME_NOT_DEFINED

Cause: Oracle object type *<typeName>* is not defined.

Action: Verify that the Oracle object type is defined.

Error code: 7075

ORACLE_VARRAY_MAXIMUM_SIZE_NOT_DEFINED

Cause: Oracle VARRAY type *<typeName>* maximum size is not defined

Action: Verify the maximum size for the Oracle VARRAY.

Error code: 7076**DESCRIPTOR_MUST_NOT_BE_INITIALIZED**

Cause: When generating the project class the descriptors must not be initialized.

Action: Ensure the descriptors are not initialized before generating the project class.

Error code: 7077**EJB_INVALID_FINDER_ON_HOME**

Cause: The Home Interface `<homeClassName.toString()>` specified during creation of `BMPWrapperPolicy` does not contain a correct `findByPrimaryKey` method. A `findByPrimaryKey` method must exist that takes the `PrimaryKey` class for this bean.

Action: Verify that a `FindByPrimaryKey` method exists and is correct.

Error code: 7078**EJB_NO_SUCH_SESSION_SPECIFIED_IN_PROPERTIES**

Cause: The `sessionName` specified on the deployment descriptor does not match any session specified in the `TopLink` properties file.

Action: Contact Technical Support.

Error code: 7079**EJB_DESCRIPTOR_NOT_FOUND_IN_SESSION**

Cause: The descriptor was not found in the session.

Action: Check the project being used for this session.

Error code: 7080**EJB_FINDER_EXCEPTION**

Cause: A `FinderException` was thrown when trying to load an object from the class with the primary key.

Action: Contact Technical Support.

Error code: 7081

CANNOT_REGISTER_AGGREGATE_OBJECT_IN_UNIT_OF_WORK

Cause: The aggregate object cannot be directly registered in the unit of work. It must be associated with the source (owner) object.

Action: Contact Technical Support.

Error code: 7082

MULTIPLE_PROJECTS_SPECIFIED_IN_PROPERTIES

Cause: The TopLink properties file specified multiple project files for the server. Only one project file can be specified.

Action: Specify one of `projectClass`, `projectFile`, or `xmlProjectFile`.

Error code: 7083

O_PROJECT_SPECIFIED_IN_PROPERTIES

Cause: The TopLink properties file does not include any information on the TopLink project to use for the server. One project file must be specified.

Action: Specify one of `projectClass`, `projectFile`, or `xmlProjectFile`.

Error code: 7084

INVALID_FILE_TYPE

Cause: The specified file is not a valid type for reading. `ProjectReader` must be given the deployed XML Project file.

Action: Contact Technical Support.

Error code: 7085

CANNOT_CREATE_EXTERNAL_TRANSACTION_CONTROLLER

Cause: Cannot create an instance of the external transaction controller specified in the properties file.

Action: Contact Technical Support.

Error code 7087:

EJB_SESSION_TYPE_CLASS_NOT_FOUND

Cause: SessionManager cannot load the class corresponding to session's type class name.

Action: Verify the class name of the session's type in the `Session.xml` or `toplink.properties` that it is fully qualified.

Error code 7088:

CANNOT_CREATE_EXTERNAL_TRANSACTION_CONTROLLER

Cause: SessionManager cannot load the class corresponding to external transaction controller's class name.

Action: Verify the class name of the external transaction controller in `Session.xml` or `toplink.properties` that it is valid and fully qualified.

Error code 7089:

SESSION_AMENDMENT_EXCEPTION_OCCURRED

Cause: SessionManager cannot load the class corresponding to the amendment class name or it cannot load the method on the amendment class corresponding to the amendment method name.

Action: Verify the class name of the amendment class in `Session.xml` or `toplink.properties` that it is fully qualified and the amendment method exists in the amendment class.

Error code 7091

SET_LISTENER_CLASSES_EXCEPTION

Cause: Cannot create the listener class that implements `SessionEventListener` for the internal use of `SessionXMLProject`.

Action: Contact Technical Support.

Error code 7092

EXISTING_QUERY_TYPE_CONFLICT

Cause: Add custom query with the same name and arguments to a session.

Action: Verify that no query is added to the session more than once or change the query name so that the query can be distinguished.

Error code 7093

QUERY_ARGUMENT_TYPE_NOT_FOUND

Cause: Cannot create an instance of the query argument type.

Action: Verify the argument type is a fully qualified class name and the argument class is included in the classpath environment.

Error code 7095

NO_SESSIONS_XML_FOUND

Cause: `Session.XML` or `toplink.properties` cannot be loaded.

Action: Add the path to the above file to the classpath environment.

Error code 7096

CANNOT_COMMIT_UOW_AGAIN

Cause: Invoke `commit()` on an inactive unit of work that was committed or released.

Action: Invoke `commit()` on a new unit of work or invoke `commitAndResume()` so that the unit of work can be reused. See *Oracle9iAS TopLink Foundation Library Guide* for more details on `commitAndResume()`.

Error code 7097:

OPERATION_NOT_SUPPORTED

Cause: Invoke a nonsupport operation on an object.

Action: Do not use the operation indicated in the stack trace.

Error Code: 7099

PROJECT_XML_NOT_FOUND

Cause: The filename specified for the XML-based project is incorrect.

Action: Verify the name and location of the file.

Error Code: 7101

NO_TOPLINK_EJB_JAR_XML_FOUND

Cause: The `toplink-ejb-jar.xml` file was not found.

Action: Please refer to the TopLink examples located in `<INSTALL DIR>\Examples`.

EJBQL Exceptions (8001 – 8009)

Error Code: 8001

`recognitionException`

Cause: The TopLink EJBQL parser doesn't recognize a clause in the EJBQL string

Action: Validate the EJBQL string

Error Code: 8002

`generalParsingException`

Cause: TopLink has encountered a problem while parsing the EJBQL string.

Action: Check the internal exception for details on the root cause of this exception

Error Code: 8003

`classNotFoundException`

Cause: The class specified in the EJBQL string was not found

Action: Ensure that the class is on the appropriate classpath

Error Code: 8004

`aliasResolutionException`

Cause: TopLink was unable to resolve the alias used in the EJBQL string

Action: Validate the identifiers used in the EJBQL string

Error Code: 8005

`resolutionClassNotFoundException`

Cause: TopLink was unable to resolve the class for an alias. This means that the class specified can not be found

Action: Ensure the class is specified properly and is on the classpath

Error Code: 8006

missingDescriptorException

Cause: The class specified in the query has no TopLink descriptor

Action: Ensure the class has been mapped and is specified correctly in the EJBQL string

Error Code: 8009

expressionNotSupported

Cause: An unsupported expression was used in the EJBQL

Action: Change the query to use only supported expressions

Synchronization Exceptions (8050 – 8070)

Error Code: 8050

DROPPING_REMOTE_CONNECTION

Cause: A communication error over the communication framework

Action: Reconnect the server to this session. If using the default discovery system restart the session on the disconnected server

Error Code: 8051

ERROR_DOING_REMOTE_MERGE

Cause: The remote server's cache has become corrupt.

Action: Restart the session on the remote server or initialize identity maps

Error Code: 8052

ERROR_DOING_LOCAL_MERGE

Cause: The local shared cache has become corrupt.

Action: Restart the session on this server or initialize identity maps.

Error Code: 8053

ERROR_LOOKING_UP_LOCAL_HOST

Cause: Failed to get the IP address of the local machine for generating the `SessionID`.

Action: Contact network administrator.

Error Code: 8054

ERROR_BINDING_CONTROLLER

Cause: A Global Naming Service, such as JNDI, was unavailable at supplied URL.

Action: Check that the `LocalHostURL` points to the Service and verify that the Service is available.

Error Code: 8055

ERROR_LOOKING_UP_CONTROLLER

Cause: Failed to find the remote connection for a remote session.

Action: Verify that the Global Naming Service, such as JNDI, is running properly.

Error Code: 8056

ERROR_UNMARSHALLING_MSG

Cause: Message from remote system could not be read by this session.

Action: None, however some servers may not be connected.

Error Code: 8057

ERROR_GETTING_SYNC_SERVICE

Cause: Failed to create the specified Synchronization Service.

Action: Verify that you have specified an available Synchronization service in the current configuration file, or in your application code.

Error Code: 8058

ERROR_NOTIFYING_CLUSTER

Cause: Failed to send existence announcement to remote servers.

Action: Check that this machine has access to the network and restart Synchronization service.

Error Code: 8059

ERROR_JOINING_MULTICAST_GROUP

Cause: Failed to set socket to listen to the multicast group.

Action: Contact network administrator to verify that this machine has network access and can use Multicast protocols.

Error Code: 8070

ERROR_RECEIVING_ANNOUNCEMENT

Cause: Error occurred while attempting to receive a Multicast announcement.

Action: Verify that this machine is connected to the network and is capable of sending and receiving multicast packets.

Mapping Workbench

This section contains information on troubleshooting the Mapping Workbench. In addition to the information in this chapter, refer the *Oracle9iAS TopLink: Mapping Workbench Reference Guide* and online help for detailed information.

Error Messages

The following Mapping Workbench errors messages (listed in alphabetical order) may appear in the status bar. Each message includes a description of the action that caused the error and the recommended resolution.

1-1 mappings for EJB 2.0 CMP descriptors must use ValueHolder indirection.

When creating a one-to-one mapping for EJB descriptors in projects with 2.0 CMP persistence, you must select the **ValueHolder** indirection option on the mapping's **General** tab.

1-M and M-M mappings for EJB 2.0 CMP descriptors must use transparent indirection.

When creating a one-to-many or a many-to-many mapping for projects with 2.0 CMP persistence, you must select the **Transparent** indirection option on the mapping's **General** tab.

A field/method pair is incomplete in the [mapping name] mapping.

You created a transformation mapping, but did not specify a complete **Object > Field Method** for each database field on the mapping's **General** tab.

A key pair has not been completely specified for a reference.

You must specify a foreign key reference for the database table. Use the database table's **Reference** tab to add a key pair.

A locking policy is specified, but the lock field is not specified.

You selected the **Optimistic Locking** field on the **Locking** tab, but did not specify the **Field**.

Aggregate fields are not specified.

For aggregate mappings, each **Field Description** on the **Fields** tab must contain a unique **Field**.

Aggregate mapping fields must be unique.

For aggregate mappings, each **Field Description** on the **Fields** tab must contain a unique **Field**.

An aggregate shared by multiple source descriptors cannot have one-to-many or many-to-many mappings.

Aggregate descriptors that are shared by multiple source descriptors cannot have mappings that contain a target object that references the descriptor.

For example, you cannot have multiple one-to-many and many-to-many or one-to-one mappings in which the target is the aggregate.

An ejb-jar.xml file needs to be specified.

If you are using a 2.0 CMP persistence project, you must specify the location of the `ejb-jar.xml` file on the project's **General Properties** tab.

Attribute is typed as a ValueHolderInterface but the mapping does not use Value Holder Indirection.

You did not specify indirection or transparent indirection for the mapping. If the class attribute is of type `ValueHolderInterface`, you must use value holder indirection for the mapping.

Attribute is typed as a ValueHolderInterface but the mapping does not use Indirection.

For one-to-one and transformation mappings, if the class attribute is of type `ValueHolderInterface`, you must use indirection for the mapping.

Classes cannot reference an aggregate target with one-to-one, one-to-many, or many-to-many mappings.

You cannot select an aggregate descriptor as the **Reference Descriptor** for a one-to-one, one-to-many, or many-to-many mapping.

EJB Class information is not compatible with project persistence type.

The information you entered on the EJB descriptor's **EJB Info** tab is incorrect, based on the project's persistence type (as specified on the project's **General Properties** tab).

If the mapping uses indirection, then the attribute must be a ValueHolderInterface.

Persistent classes that use indirection must replace the relationship attributes with a value holder (an instance of a class that implements the `ValueHolderInterface`, such as `ValueHolder`).

On the attribute's **General** mapping tab, select the **Use Indirection** option and specify **ValueHolder** type.

Mapping references write lock field, but is not read-only.

You specified a locking policy for a descriptor, but one of the attribute mappings is not read-only. Select the **Read Only** option on the mapping's **General** tab.

Mapping uses Indirection but its associated attribute is not a ValueHolderInterface.

If you select the **Use Indirection** option for a one-to-one mapping with transparent indirection, the associated class attribute must be **ValueHolderInterface**.

Mapping uses Value Holder Indirection but its associated attribute is not a ValueHolderInterface.

If you select the **Use Indirection (ValueHolder)** option for a one-to-many, many-to-many, or direct collection mapping, the associated class attribute must be **ValueHolderInterface**.

Mappings for EJB 2.0 CMP descriptors which use Value Holder Indirection must not use method accessing.

You cannot use method accessing on mappings for EJB 2.0 CMP descriptors which use **ValueHolder Indirection**. Because EJB attributes are code-generated, reference

mappings *should not* be set to use method access. The attributes are code-generated to be of type **ValueHolder** but the abstract methods are defined to return the local interface type of the related bean.

Method accessors have not been selected.

You selected **Use Method Accessing** for a mapping, but did not select a **Get** and **Set** method on the mapping's **General** tab.

More than one writable many-to-many mapping cannot use the same relation table.

A project cannot have more than one writable many-to-many mapping using the same relation table. You must either make sure only one of the mappings is writable or choose a different table for each many-to-many mapping.

Multi-table reference should be defined from the base table [table name] to the derived table.

This descriptor has Inheritance and Multi-Table Info advanced properties defined on it. The multi-table relationship that is defined between the base class's table and this derived class's table must be defined from base to derived.

No attribute is associated with the mapping named [mapping name].

The specified mapping does not have an associated class attribute. Either refresh the class or remove the mapping.

No class in inheritance tree is marked as root.

The inheritance hierarchy must contain one root descriptor. Select the **Is Root Descriptor** option on the **Inheritance** tab.

No class indicator field is selected for this root class.

You selected the **Use Class Indicator Dictionary** option for the root descriptor in the inheritance hierarchy, but did not specify an indicator value for the root and its children. Use the **Indicator Type** drop-down list on the **Inheritance** tab for the root class.

No collection type is selected.

You must select a collection type on the mapping's **Collection** tab.

No database field is selected.

For attributes with direct-to-field mappings, you must specify a **Database Field** on the mapping's **General** tab.

For attributes with type conversion mappings, you must specify a **Database Field** on the mapping's **General** tab.

No database type is selected.

For type conversion and object type mappings, you must select a **Database Type** on the mapping's **General** properties tab.

No direct field is specified.

For direct collection mappings, you must select a **Target Table** and **Direct Field** that the direct collection specifies.

No field/method pairs defined for the [mapping name] mapping.

You created a transformation mapping, but did not specify an **Object > Field Method**. You must specify at least one field/method pair, unless the mapping is **Read Only**.

No indicator field is selected.

You created a variable one-to-one mapping, but did not specify a database field in which to store indicator values.

No indicator values are specified.

You created a variable one-to-one mapping, but did not specify indicator values for each object type.

No method specified for transforming a database row into this attribute.

For transformation mappings, you must select a **Database Row -> Object Method** on the mapping's **General** tab.

If you are creating a write-only mapping, you can ignore this warning.

No null value type has been selected

You selected the **Use Default Value When Database Field is Null** field on a mapping's **General** tab but did not specify a default **Type** and/or **Value**. This message may also appear after using the Package Renamer when upgrading an older Mapping Workbench project.

No object type is selected.

You created an object type mapping, but did not select an **Object Type** on the mapping's **General** tab.

No object-type mappings have been specified.

For object-type mappings, you must specify at least one mapping (**Database Value** and **Object Value**) on the mapping's **General** tab.

No primary key(s) specified in [table name] table.

You must specify a primary key for each database table. When importing tables from a database into the Mapping Workbench, the primary key information will be retained only if the JDBC driver supports the `getPrimaryKeys()` method.

Ensure that a primary key is specified for each descriptor on the **Descriptor Info** tab.

Not all query key associations have foreign key fields specified.

You must specify a foreign key field for each query key association on the **Query Key Associations** tab for variable one-one mapping.

No query key associations have been defined.

You created a variable one-to-one mapping, but did not define a key pair on the mapping's **Query Key Association** tab.

No reference descriptor is selected.

You must select a **Reference Descriptor** for each relationship mapping on the mapping's **General** tab.

No relation table is selected.

You created a many-to-many mapping, but did not specify a **Relation Table** on the mapping's **General** tab. The relation table represents the relation between the primary keys of the source and target table.

No sequence field name is selected.

If you select **Use Sequencing** on a descriptor's **Descriptor Info** tab, you must also specify a **Name**, **Table**, and **Field**.

No sequence name is selected.

If you select **Use Sequencing** on a descriptor's **Descriptor Info** tab, you must also specify a **Name**, **Table**, and **Field**.

No source reference is selected.

You created a many-to-many mapping, but did not select (or create) a source table reference on the mapping's **Source Reference** tab.

The source table reference must contain a **Source** field (from the mapping's relation table) and a **Target** field (from one of the descriptor's associated tables).

No table reference is selected.

You must select (or create) a table reference for each relationship mapping on the mapping's **Table Reference** tab.

No target reference is selected.

You created a many-to-many mapping, but did not select (or create) a target table reference on the mappings **Target Reference** tab.

The source table reference must contain a **Source** field (from the mapping's relation table) and a **Target** field (from one of the descriptor's associated tables).


Object values do not match the specified attribute or object type.

For object-type mappings, you entered an **Object Value** that does not match the **Object Type**, as specified on the mapping's **General** tab. For example, you specified an **integer** type, but entered a value of **A**.


One or more field types have not been specified.

You must specify the **Type** for each database field on the **Field** tab.


One of the descriptors in this package is incomplete.

One (or more) of the descriptors in a package contains errors. The Mapping Workbench places an error icon  beside the incorrect package. Expand the package to display its descriptors.

One of the packages is incomplete.

One (or more) of the packages in your project contains errors. The Mapping Workbench places an error icon  beside the incorrect project. Expand the project to display its packages.

One of the tables is incomplete.

One (or more) of the database tables in your project contains errors. The Mapping Workbench places an error icon  beside the incorrect table. Expand the database object to display its tables.

Primary keys do not match across associated tables and no reference(s) specified in multi-table policy info.

Primary key field names must match across associated tables, or references must be defined from the primary table to each secondary table.

Primary key(s) do not match parent's primary key(s).

In an inheritance hierarchy, the child's primary key(s) must match the root's primary key(s)


[descriptor name] references [reference descriptor name] which is not active.

You tried to select an inactive descriptor as a Reference Descriptor on the mapping's **General** tab. Either select a new **Reference Descriptor** or make descriptor active.

Root class does not include an indicator mapping for this descriptor.

The root class in the inheritance hierarchy is set to use class indicator dictionary. It does not contain an indicator value for this child class.

Some mappings are incomplete.

One (or more) of the attributes of a descriptor contains mapping errors. The Mapping Workbench places an error icon  beside the incorrect attribute(s). Expand the descriptor to display its mappings.

The Collection class is a Map, but the key method is not selected.

You created a direct collection type mapping, but did not specify the key method.

The inheritance hierarchy originating in this descriptor cannot contain both aggregate and non-aggregate child descriptors.

Aggregate and class descriptors cannot be in the same inheritance hierarchy.


The [access method type] method for this mapping's method accessing field is no longer visible to this descriptor.

You changed the class hierarchy within the project, causing the method access type (get or set) to no longer be visible to the class.

The method specified for the copy policy on this descriptor is no longer a visible member of this class.

You changed the class hierarchy within the project, causing the copy policy to no longer be visible to the class.

The database is incomplete.

The databases in a project contains errors. The Mapping Workbench places an error icon  beside the database. Expand the database to display its tables.

The event policy's [method type] method is no longer a visible member of this descriptor's associated class.

You changed the class hierarchy within the project, causing the method to no longer be visible to the class.

The following fields have multiple writable mappings: [field name].

Multiple mappings cannot write to the save database field. Each database field must have a single, writeable mapping.

The following Query Keys do not have associated database fields:

The database field(s) for the query key(s) listed have been removed from the associated table. If you want to use these query key(s), you must specify a database field(s) for them.

The following primary key fields have no writable mappings: [field name].

Each primary key field must have a writeable mapping. Ensure that the primary key field mappings *are not* read only.

The following primary key fields are unmapped: [field name].

Each primary key field must have a writeable mapping.

The method specified for the copy policy on this descriptor is no longer a visible member of this class.

You changed the class hierarchy within the project, causing the copy policy to no longer be visible to the class.

The method specified for the inheritance policy's class extraction method on this descriptor is no longer a visible member of this class.

You changed the class hierarchy within the project, causing the inheritance policy to no longer be visible to the class.

The multi-table reference should not be defined on the database.

When using multi-tables with differently named primary keys, you must set a reference from the TOP table to the BOTTOM table. This reference must not be an actual constraint on the database. Select the table in which this is defined and unselect the **On Database** option.

The parent and children of an aggregate descriptor must also be aggregates.

If an aggregate descriptor is in an inheritance policy hierarchy, *all* descriptors in the hierarchy must be aggregates.

The reference [table reference] does not have any field associations.

You selected a table reference for a mapping, but did not add a key pair. You must specify source and target key pairs for the reference.

The reference must have at least one field association.

You selected a table reference for a relationship mapping, but did not define a source and target field key pair. For variable one-to-one mappings, you must define a query key pair (in the source descriptor's tables) to use for the common query key.

The selected parent descriptor for this descriptor's inheritance policy does not have an associated inheritance policy.

You selected a Parent Descriptor for a descriptor's inheritance policy that does not have an inheritance policy. Parent Descriptors must have a valid inheritance policy.

The source reference does not contain any field associations.

You created a many-to-many mapping, but did not define a source and target reference for the *source* reference. You must define a table reference and the appropriate key pairs for each source reference.

The source reference must be specified.

You created a many-to-many mapping, but did not define a source table reference on the mapping's **Source** tab.

The target reference does not contain any field associations

You created a many-to-many mapping, but did not define a source and target reference for the *target* reference. You must define a table reference and the appropriate key pairs for each target reference.

The target reference must be specified.

You created a many-to-many mapping, but did not define a target table reference on the mapping's **Target** tab.

There is no database associated with the query key [query key name].

You did not associated the specified query key with a database table. You must select a database **Name** and **Table** on the **Query Keys** tab.

This class is a subclass of a final class.

If you select the **Final** option on descriptor's **Class Info**, **Class** tab for a class, the class cannot contain subclasses.

This root class has no class indicator mappings for its hierarchy.

When creating an inheritance policy, if you select the **Use Class Indicator Dictionary** option on the root descriptor's **Inheritance** tab, you must specify the indicator values for all subclasses.

Note: TopLink displays a list of each subclass and indicator value if you have identified the subclasses' parent descriptor.

"Use factory" is specified for the Instantiation policy, but all required info is not specified.

You selected the **Use Factory** option on the descriptor's **Instantiation Policy** tab, but did not specify the **Factory Class**, **Factory Method**, or **Instantiation Method** fields.

"Use method" is selected for the Instantiation policy, but no method is selected.

You selected the **Use Method** option on the descriptor's **Instantiation Policy** tab, but did not specify the field.

Writable mappings defined for the class indicator field [field name].

The class indicator field (as defined on the Inheritance tab) should not contain any writable mappings.

Classpath Issues

The Mapping Workbench does not display the class(es) to import.

If the class is not available to import:

- Ensure that the class is in your project's CLASSPATH (on the project's **General** properties tab).
- Ensure that the class is in the `.zip` or `.jar` file. You cannot import compressed classes.

The Mapping Workbench generates an exception error when importing classes.

TopLink's class import utility did not start correctly. One of the classes includes a static initialization method which may cause the import utility to fail.

Ensure that your project's CLASSPATH points to the *root* folder of your package hierarchy. For example, to import the `com.company.class` package in the `C:\classes\com\company` directory, your project CLASSPATH should be `C:\classes\`.

The Mapping Workbench fails to import the class but does not generate an exception error.

Ensure that you have properly indicated the directories that contain the domain class(es) to map on the project's **General** tab.

The CLASSPATH containing your JDBC drivers should still be on your system CLASSPATH. The TopLink Mapping Workbench CLASSPATH is meant for domain classes only.

You may also need to edit the CLASS.PATH section of the `MappingWorkbench.lax` file to include the appropriate files.

Database Connections

If the Mapping Workbench encounters problems communicating or logging into the database, you should:

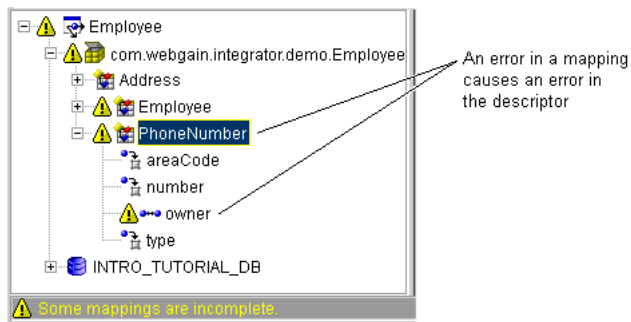
- Ensure that the driver class, login name, password, and JDBC database URL are correct.
- Verify that your PATH includes all files (for example, native .dll files) required by the driver.
- Verify that your CLASSPATH (for TopLink) includes all classes and files (for example, .zip or .jar) required by the driver.
- Consult with your database administrator and confirm that the:
 - database server is setup correctly.
 - database permissions are set correctly
 - database has enough available connections

Troubleshooting Descriptors

TopLink checks each descriptor and mapping to ensure that you have properly defined the required settings.

If a descriptor contains a deficient mapping or property, the Mapping Workbench displays a yellow caution icon to the left of its icon. If you select the error, the Mapping Workbench displays the complete error message in the status bar.

Figure 3–1 Sample Deficient Mapping



Refer to the *Oracle9iAS TopLink: Mapping Workbench Reference Guide* for more information on working with descriptors.

Frequently Asked Questions (FAQs)

This section contains frequently asked questions and their answers. The TopLink documentation set contains more detailed information on each of these items.

The FAQs are grouped in the following categories:

- [General](#)
- [Mappings](#)
- [Sessions](#)
- [Unit of Work](#)
- [Querying](#)
- [TopLink Mapping Workbench](#)
- [Enterprise](#)

General

What is TopLink and for whom is it intended?

TopLink is an object-relational framework, which facilitates the construction of an object-oriented system that can store its objects/data in a relational database. The framework provides facilities, which allow the developer to deal primarily with objects independently of their underlying relational structure.

TopLink is intended for development projects where rapid development, code maintenance, flexibility, and performance are a priority. TopLink provides a proven framework with many advanced features, which make the many tough issues surrounding object-relational development manageable.

Refer to the *Oracle9iAS TopLink: Getting Started Guide* for detailed information.

What are the primary components which compose TopLink?

TopLink is composed of two main logical components: a *development* and a *runtime* component.

The development component provides a graphical user interface and a class library (`tl_tools.jar`). These development tools allow the developer to interact with both the domain classes and the relational database in order to construct their persistent mappings.

The runtime component of TopLink is a series class libraries. The core TopLink classes are contained in `tl_core.jar`, and the enterprise classes are contained in `tl_x.jar`. The `tl_sdk.jar`, `tl_sdkx.jar`, and `xerces.jar` libraries are used if you will be reading TopLink Mapping Workbench-generated XML files, or using the TopLink SDK.

The `toplink.jar` file contains all the TopLink runtime classes from `tl_core.jar`, `tl_x.jar`, `tl_sdk.jar`, `tl_sdkx.jar`, and `tl_tools.jar`.

The TopLink CMP for BEA WebLogic and TopLink CMP for IBM WebSphere Foundation Libraries each contain an additional library (`tl_wlsx.jar`, and `tl_wasx.jar`, respectively).

Is TopLink an application or EJB server?

No, TopLink is not an application or EJB server – it is a pure Java class library which provides an object-relational framework. Since it is certified 100% Pure Java it is capable of running within any certified Java environment. With the enterprise classes, TopLink integrates with application and EJB servers.

TopLink also provides extensions products that integrate with EJB servers to provide Container Managed Persistence (CMP) for Entity Beans.

How do you configure your environment to work with TopLink?

Following the installation of TopLink you must configure your development system to make use of the TopLink classes as well as allow TopLink to access the core Java classes, JVM, and your domain classes.

When modifying your `PATH` and `CLASSPATH` variables, ensure you are using the *environment* variables.

1. Ensure a valid JVM exists in your `PATH`. Some JVMs will not work properly with TopLink since they do not make use of the `CLASSPATH`. In these cases you will have to ensure that another JVM is in your `PATH` before the invalid one.

2. Ensure the core TopLink classes (`tl_core.jar` and `tl_tools.jar`) are in your CLASSPATH.
3. Ensure that all domain classes, which you will want to map persistently using TopLink, are included in your CLASSPATH.
4. Ensure that the path to your JDBC driver classes are included in your CLASSPATH.

What databases and JDBC drivers can be used with TopLink?

TopLink can communicate with any relational database which has a JDBC-compliant driver available such as Oracle, DB2, Sybase, SQL Server, Informix, Cloudscape, and Access. TopLink supports both the JDBC 1.x and 2.0 standards.

What constraints does TopLink enforce on an object model?

TopLink does not force the developer to subclass from one of its classes or use special types to maintain relationships. You could build a domain model to map persistently which has no dependencies on TopLink. TopLink does however require access to instantiate the persistent classes as well as access to get and set values within these objects.

In order to instantiate your objects, TopLink requires either a zero argument public constructor or a factory object and a method to invoke to get a new instance.

In order to gain access to the fields of a persistent class, TopLink requires public get and set methods or public fields. Starting with JDK 1.2 or Java 2 TopLink can access protected or private fields, provided the environment has been configured to grant TopLink access.

What Java types does TopLink support?

TopLink automatically converts database values for the following Java types: `String`, `Number`, `BigDecimal`, `BigInteger`, `Long`, `long`, `Integer`, `int`, `Float`, `float`, `Double`, `double`, `Byte`, `byte`, `Short`, `short`, `java.util.Date`, `java.sql.Date`, `Time`, `Timestamp`, `Calendar`, `byte[]`, `Boolean`, and `boolean`.

For other types, you must use an object-type, type conversion, or transformation mapping; or transform the value in the direct-to-field get/set methods.

What code generation does TopLink perform?

Because TopLink uses a metadata architecture it does not need to generate any code to enable your application's persistence.

TopLink can generate:

- The TopLink project and associated descriptors into a single Java class which, when compiled, can be used in place of the development time TopLink Mapping Workbench files.
- A table creator Java class, which, when compiled can create all of the tables on the database with required constraints.

The TopLink Mapping Workbench can generate:

- Source code for the classes defined in the project
- Java object model, EJB-CMP EntityBean model, and EJB-BMP EntityBean model from an existing database schema

What additional functionality is available in the TopLink CMP for BEA WebLogic and TopLink CMP for IBM WebSphere Foundation Libraries?

The TopLink CMP for BEA WebLogic and TopLink CMP for IBM WebSphere Foundation Libraries are supersets of the TopLink for Java Foundation Library. You get all of the same functionality. However, these libraries also allow you to use TopLink with container-managed persistence (CMP) Entity Beans on BEA WebLogic and IBM WebSphere application servers.

Can I use TopLink for Java Foundation Library in my application server?

TopLink for Java Foundation Library can run in any application server running in a Java Virtual Machine. You can use regular Java objects or integrate with bean-managed persistence (BMP) Entity Beans. You only need the specialized TopLink CMP for IBM WebSphere or BEA WebLogic Foundation Libraries if you are going to use TopLink with container-managed persistence (CMP) Entity Beans in those servers.

Mappings

What types of mappings does TopLink support?

TopLink provides a complete set of mappings for both direct and reference relationships. This allows developers to map from the object model to the relational model without one model dictating to the other. The `oracle.toplink.mappings` package contains the following:

- Direct mappings

- DirectToFieldMapping
- DirectMapMapping
- TypeConversionMapping
- ObjectTypeMapping
- SerializedObjectMapping
- TransformationMapping
- Relationship mappings
 - AggregateObjectMapping
 - OneToOneMapping
 - VariableOneToOneMapping
 - OneToManyMapping
 - ManyToManyMapping
 - DirectCollectionMapping
 - AggregateCollectionMapping
- Object-relational mappings
 - StructureMapping
 - ReferenceMapping
 - NestedTableMapping
 - ArrayMapping
 - ObjectArrayMapping
- SDK Mappings
 - SDKAggregateObjectMapping
 - SDKDirectCollectionMapping
 - SDKAggregateCollectionMapping
 - SDKObjectCollectionMapping

Can TopLink access private and protected fields of my Java class?

TopLink comes with class libraries for Java 2 (JDK 1.2 and higher). With Java 2 TopLink can access non-public fields and/or methods. Grant the

ReflectPermission permission to TopLink to “suppress default Java language access control checks when using reflected members and constructors.” The granting of permissions is done via the security policy file using the policy tool application.

See the internet site

<http://java.sun.com/products/jdk/1.2/docs/guide/security/index.html> for more information.

Can columns, which allow NULL, be mapped to primitive Java types?

Normally no, since primitive types in Java cannot be null. If the field value is null, TopLink will try to assign null to a primitive, which raises a `NullPointerException`.

The easiest way to avoid this problem is to ensure all of the database fields that store primitive values cannot be null. You could also configure the mapping to use a default value for null by using the `NullValue` property on all direct mappings:

```
mapping.setNullValue(new Integer(0));
```

You can also configure `NullValue` for all mappings and descriptors on the `DatabaseLogin`:

```
login.setDefaultNullValue(int.class, new Integer(0));
```

Does TopLink support reference mappings to interfaces as well as classes?

Yes, you can define mappings that reference interfaces instead of classes. Interfaces can be defined for a single class or for a set of implementors. Querying on the interface will query each of the implementors.

What does it mean when a reference mapping is *privately owned*?

TopLink provides the ability to define any reference mapping (1:1, 1:M, and M:M) as *privately owned*. This means that when the source object is written or deleted so should all of its privately owned parts.

Is it possible to have multiple mappings to a single column?

Yes, provided that only one of the mappings is made writable. TopLink requires one writable mapping for which it will retrieve the required value from the object when constructing the row for the database. You can configure a mapping to be read-only or, in the case of a 1:1 mapping, may configure a target foreign key. Both of these will allow an additional mapping of a field.

How can TopLink use sequencing to assign the generated primary key value to the object at creation time?

TopLink supports sequencing by assigning the object a unique identifier when it is inserted into the database. TopLink assigns the identifier here because this is the first time TopLink gets access to the object. When using native sequencing on Sybase, SQL Server and Informix the identifier cannot be obtained until after the insert.

If you want the sequence number assigned at creation time instead of during the insert:

1. Create your new object and assign it its identifier by using the following protocol. You must provide the sequence name.

```
Number nextID = session.getPlatform().getSequenceNumberNamed(" SEQ_NAME" ,  
session, "SESSION_NAME");
```

Note that this is not a public API protocol, although it is publicly accessible.

2. Write the object normally through TopLink.

Within a unit of work all registered new objects can also be assigned sequence numbers through the `assignSequenceNumber(Object)` and `assignSequenceNumbers()` methods.

How do TopLink inheritance mappings compare with Java class inheritance?

To map a Java inheritance hierarchy in TopLink, each class in the hierarchy will define a TopLink descriptor. Each subclass' descriptor will define its parent to be its superclass.

This inheritance hierarchy can be reflected in the database in many ways. TopLink requires a common table for the root class that must be shared by all of the subclasses. The subclasses are free to define additional tables to store their additional data or can share their parent's tables. TopLink uses an indicator field in the root table to determine the class to instantiate from the row, however this mechanism can be customized if required.

It is a common misconception that a system with a root domain class must be mapped using this kind of inheritance mapping. Typically this should only be done where the database schema dictates or the object model allows similar types to be stored together. If it does not make sense to define a table for the root class, this class can not be mapped at all, or can be mapped as an interface. TopLink supports querying on interface descriptors and does not require that they be mapped to a database table.

In a mapped inheritance hierarchy all subclasses must share the root's primary key. In case of multiple tables the primary key in the additional table may be named differently than in the root table. In this case a name mapping must be provided in the descriptor, or in complex cases a multiple table join expression. The primary key that will be used internally in TopLink and for caching will always be the root table's.

How can I define a selection criteria that is not only on foreign keys in a foreign reference mapping?

Foreign reference mappings internally use a selection query to read objects from the target. This selection query is like any other read query, and can be accessed to provide advanced customization on the mapping's query. The selection query can itself be changed to provide a custom query if complex querying is required. If you have a relationship that is not solely based on foreign keys, an expression can be given to the mappings selection criteria.

```
ExpressionBuilder builder = new ExpressionBuilder();
mapping.setSelectionCriteria( builder.getParameter("EMPLOYEE.ADDR_
ID").equal(builder.getField("ADDRESS.ID").and(builder.getField(
"ADDRESS.TYPE").equal("home")));
```

When do I have to define a 1:1 mapping back from the target to source object in a 1:M mapping? Are there alternatives?

When a source object defines a 1:M mapping to a target object, you must also create a 1:1 mapping from the target object back to the source object in order to populate the foreign key field in the target table. Without this back-mapping, TopLink does not know how to populate this foreign key field in the target table.

You can use an `AggregateCollectionMapping` to represent the aggregate (privately-owned) relationship between a single source object and a collection of target objects. Unlike the normal `AggregateObjectMapping`, there is a target table being mapped from the target object; unlike the normal 1:M mapping, there is no need for setting the 1:1 back reference, as the foreign key relationships have already been resolved by the aggregation. The `AggregateCollectionMapping` was introduced in version 3.0.

You can avoid setting the 1:1 back reference by defining a direct-to-field mapping on the foreign key field. In this case the application must make sure that the object model picks up the correct value from the source object. However, this alternative may result in an overly complex implementation or object model. It is recommended you use a 1:1 back reference.

Another alternative is to define a M:M mapping from source to target and create a relation table. This is the best solution as it ensures that the object model is consistent with the data model. Neither model stores information of the relationship in the target object and multiple source objects referencing the target can be supported.

How is a 1:1 variable class mapping defined using TopLink?

TopLink supports variable 1:1 mappings in which the source object has a reference to one of two or more non-inherited classes. To define a variable 1:1 mapping the target classes of the mapping must be related through an *interface*.

You must define an interface descriptor for the interface and the target classes must implement the interface. The selection criteria of the mapping will be defined using query keys from the target interface and a type field in the source object's table will store the type of object that is referenced.

How can TopLink's TypeConversionMapping be customized to make use of a third party class?

TopLink provides a rich set of type conversions between database types and the Java types provided with JDK. It is possible to make use of additional or third party types in a domain class through customization of TopLink.

TopLink makes use of a singleton `ConversionManager` (found in the `oracle.toplink.helper` package). Developers can subclass this class and create a custom conversion manager with the additional conversions required. At system startup an instance of this class must be provided as the singleton conversion manager.

What types can be used to hold the values for a collection mapping (1:M, M:M, and DirectCollection)?

By default, TopLink uses the `Vector` class for collections. JDK 1.1 supports only `Vector` and `Hashtable` (or their subclasses). Java 2 can use any implementor of the `Collection` or `Map` interface. TopLink supports collections through the mapping's container policy. You can define a new container policy that uses other types of containers that do not implement the `Collection` or `Map` interfaces.

How can I map BLOB data and what are the issues?

TopLink maps BLOB data from the database into a byte array in the object model. BLOBs can also be mapped through transformation or serialized mappings to be complex objects in the object model.

Because many JDBC drivers have problems dealing with large binary data, TopLink supports multiple options. By default TopLink binds binary data into the SQL statement. TopLink also supports printing the data using the JDBC binary escape clause or native SQL.

Binding is enabled/disabled on the `DatabaseLogin` through the `dont/useByteArrayBinding()` property. When binding is enabled, TopLink will parameter bind the binary value. Because some JDBC drivers may have problems binding large values (but can support large values through streams), TopLink also supports stream binding. This is enabled in `DatabaseLogin` through the `useStreamBinding()` property. Some drivers have limits to BLOB sizes that may require your application to split the binary data into multiple fields or use a direct collection or 1:M mapping to map the data.

Large string data may also require binding. TopLink can be configured to bind string data through the `useStringBinding(int)` property on `DatabaseLogin`.

- **Serialized Mapping** – TopLink also provides the ability to map object structures, which would not map easily to relational tables, with a `SerializedObjectMapping`. Mappings of this type will serialize the object structure from the root attribute and store it into a BLOB field on the database. When the object is read from the database it will be de-serialized into its original structure. This mapping does not require the developer to do the serialization, but instead to make sure that objects in the structure are serializable and self contained.
- **Oracle 8 BLOB note** – Before driver version 8.1.7, the Oracle 8 JDBC drivers did not support Oracle 8 BLOB types correctly. This has been fixed in the Oracle 8.1.7 JDBC 2.0 driver. We suggest the usage of LONGRAW fields if you are using an Oracle 8 JDBC 1.x driver.

What causes the "invalid column name" message in an SQL Exception?

This message is given because SQL is trying to write to a non-existent column in the table. Possible reasons for this could be:

- The field names specified in the foreign reference mapping are not properly specified.
- The column name is misspelled or in the wrong case. Check the methods where field names are set in the mapping.

I've set up sequencing, but I'm getting an error when inserting a new object. What's wrong?

It's likely that, although you have set up the descriptor's and project's sequencing information, that the sequence table is not yet initialized. The sequence table must contain one row for each sequence name you are using, and have an initial value in the count column. For example, if your `Employee` and `Address` descriptors are using sequencing, an initialized sequence table might look like this:

Table 4-1 Sequence Table

SEQ_NAME	SEQ_COUNT
EMPLOYEE	0
ADDRESS	0

I have a domain model with a lot of inheritance. Do I have to model my inheritance in TopLink?

No. Inheritance in TopLink only needs to be modeled relationally if you plan to query or write the super classes.

Consider the example where class **A** has a subclass called **B**. **A** has two attributes: `id` and `description`. On the database, there is both an `ID` and `DESCRIPTION` field on **B**'s table, but **A** has no table associated with it. All you need to do is map **B**; you can ignore its superclass **A**. The fields contained in **A** can be mapped through mapping **B**. This allows you to map an object model but ignore certain classes that shouldn't logically be persisted (e.g. abstract classes, "PersistentObject" superclasses, etc.).

I have specified Transparent Indirection for my mappings. Do I have to do anything special to get it to work?

In general, no. You do, however, have to make sure that your attribute is of the correct type. In the case of collection indirection, you must make sure that your attribute type is `java.util.Collection`. In the case of one-to-one indirection (available only in JDK 1.3), you must make sure that the attribute type is an interface. This is because TopLink replaces the target object with a special wrapper object that implements the same interfaces as the target class. If the variable is not of the correct type, TopLink will not be able to put the value into it.

Serialization can also cause trouble with Transparent Indirection. Unless Remote Sessions are being used, Transparent Indirection can not be instantiated on the client. The same holds true with regular indirection; however, the point at which

the indirection is instantiated is now different with Transparent Indirection. Using traditional `ValueHolders`, the indirection is triggered as soon as the `getAttribute()` method is called. Before an object that is referenced using Transparent Indirection can be accessed on the client, a message must be sent to it on the server (e.g. `getSize()` for transparent collections, `hashCode()` for transparent one-to-ones, etc.).

Sessions

What is the difference between a DatabaseSession and ServerSession?

TopLink provides two types of sessions for developers to connect with the database:

- Use the `DatabaseSession` for two-tier applications in which a single JDBC connection will service all database requests.
- Use the `ServerSession` for three-tier applications. This allows multiple clients to access the database through a configurable JDBC connection pool. TopLink's 3-Tier solution also includes a `ClientSession` and a `RemoteSession`. Each client to the `ServerSession` must communicate through a `ClientSession`. The `RemoteSession` can be used across RMI, CORBA or EJB from a client applet or application.

`DatabaseSession` provides the unit of work functionality and much lower level access to the database through `insertObject(...)`, `updateObject(...)`, `deleteObject(...)`, `beginTransaction()`, `commitTransaction()`, and `rollbackTransaction()`. These are not provided through the `ServerSession`, `ClientSession` and `RemoteSession` since merging of objects into a shared cache requires the more complicated behavior delivered with TopLink's `UnitOfWork`.

How does TopLink maintain object identity?

Each class type has an identity map in a TopLink `DatabaseSession` or `ServerSession`. Objects are stored in these identity maps based on the object's defined primary key field values.

TopLink guarantees object identity by checking the cache during reads to ensure that for any given set of relational data (which maps to an object) only a single instance of the class is created. This means that multiple reads of the same object will return the same reference. This is extremely valuable in ensuring that object changes made within the application stay in sync with other users of the objects. Object identity also allows the building of complex structures involving circular references without any extra work or potential infinite loop.

Violating object identity refers to creating objects which either already exist in the cache (same class and primary key) or in the database instead of using the object returned from TopLink.

How does TopLink's object caching work?

TopLink uses an object's primary key field values, as defined in the descriptor, to identify an object uniquely. The object cache, maintained within either the `DatabaseSession` or `ServerSession`, caches objects based on class and primary key values. TopLink uses six types of caches which can be configured on a per class basis.

- None (`NoIdentityMap`)
- Full (`FullIdentityMap`)
- Cache (`CacheIdentityMap`, fixed size, drops least recently used objects)
- Weak (`WeakIdentityMap`)
- Soft Cached Weak (`SoftCacheWeakIdentityMap`, default for Java 2)
- Hard Cached Weak (`HardCacheWeakIdentityMap`)

How does TopLink know when to INSERT an object versus UPDATE it?

TopLink determines whether to INSERT or UPDATE an object based upon the following existence-checking settings (provided in each persistent class's descriptor):

- Check cache (default)
- Check database
- Assume existence
- Assume non-existence

Use the **Check Database** option only when caching cannot be guaranteed for modified objects. This option will lower performance (as compared to **Check Cache**) because it must access the database each time.

Does TopLink support table qualifiers and creators?

In some databases or database configurations table names must be qualified with either a table owner/creator name or a table space name. TopLink allows you to specify these either on a per session basis or individually in each descriptor.

By sending the following method to the `DatabaseLogin` object before you login to the session, the table names of all of the registered descriptors will be qualified with the creator or qualifier name.

```
login.setTableQualifier("PERSONNEL");
```

It is also possible to define individual qualifiers by qualifying the table name in the descriptor using the dot notation. This must be done through an amendment method and is not supported from the TopLink Mapping Workbench.

Why do dates not work through my JDBC driver?

Some JDBC drivers do not support the JDBC date syntax. In this case TopLink also supports printing date values in their native database syntax. To enable native SQL printing set the `useNativeSQL()` property in `DatabaseLogin`.

Why do I get a conversion exception when reading date values from my database?

TopLink optimizes the way that data is converted from the JDBC driver. This includes date type where TopLink handles the conversion from strings internally. Some JDBC drivers, when returning the string representation of the date, return the wrong string syntax. In this case you must disable TopLink's data optimization through the `dontOptimizeDataConversion()` property in `DatabaseLogin`.

Unit of Work

What functionality does a UnitOfWork provide?

TopLink's `UnitOfWork` provides developers with an object transaction mechanism. TopLink's `UnitOfWork` provides a working copy of an object and its related objects so that changes can be made in isolation. Upon commit the `UnitOfWork` identifies what has changed with the registered object's working copies. This minimal change set is then written and committed to the database. Upon successful commit of the changes the modifications are then applied to original copies of the objects.

The `UnitOfWork` provides the following functionality:

- Synchronizes changes to the databases and the object model
- Isolates edits of objects into their own transaction space
- Supports parallelism and nesting

- Sends a minimal amount of SQL to the database during the commit, by updating only the exact changes down to the field level
- Maintains referential integrity by ordering the inserts, updates, and deletes
- Resolves bi-directional references
- Avoids database deadlock through ordering table access
- Supports merging of serialized RMI or CORBA clones

What does *registration* mean in a UnitOfWork?

Registration is a process of letting the UnitOfWork maintain and isolate an object's changes. That is why any object you want to change must be registered before you start making changes. The UOW maintains a backup copy of the original object and returns you a working copy to work on. The original object must be left untouched until the UnitOfWork is committed. This ensures that uncommitted changes are not seen by other users of the object.

There are two ways of registering objects in the UnitOfWork:

- You can explicitly register an object in a UnitOfWork by using the `registerObject()` method. This must be done for objects read outside the UOW. The returned clone from the `registerObject()` method must be used for editing.
- You can read objects through the UnitOfWork, or traverse the relationships of registered objects, in which case they will automatically be registered and cloned.

How many levels of nested units of work are possible? Can units of work be used in parallel?

There is no limit to the number of units of work that can be nested or used in parallel. When a nested unit of work is committed, nothing is changed on the database – instead, the parent unit of work is notified of the changes, and they will be made when the parent is committed. A possible application for nested units of work is a user interface “wizard” in which each page of the wizard could be a separate nested unit of work committed when the user clicks **Next**. At the last page of the wizard, clicking an **OK** button commits the parent unit of work, thereby committing all of the nested units of work. Clicking the **Cancel** button cancels the entire transaction.

Units of work can also be used in parallel. Essentially, this means that multiple units of work can be acquired from a single TopLink Session, and used simultaneously.

Do I have to register new objects created in the unit of work? What about non-private parts of the new objects? Should they be registered too?

Yes, new objects created in the UnitOfWork must be registered but their non-private parts should not be registered as TopLink uses “persistence by reachability” for such objects. This means that if a new object is referenced by another registered object it does not have to be registered. New objects must be registered before they are related to other existing object to ensure the correct clone is referenced by the other object.

Why do I get the error "The object <object name> is not from this unit of work space but the parent session"?

This exception is thrown when the unit of work detects that an object referenced from another registered object was never registered.

This can occur when:

- An object read from the session was assigned to a variable in an object registered in the unit of work.
- A new object was first related to an existing object and then registered into the unit of work. A helpful way to debug this type of problem is to change your `toString` to include the object's `hashCode` to ensure that you are not confused about which object you are having a problem with.

Querying

Do TopLink's queries support EJBQL?

A `ReadAllQuery` and `ReadObjectQuery` have the message `setEJBQLString(String EJBQLString)` which can be used to set the selection criteria of the query to a `String` containing EJBQL.

The Mapping Workbench also supports EJBQL queries.

What capabilities are supported by TopLink's expressions?

TopLink's rich expression framework allows almost any query to be specified in terms of the object model. These expressions are then dynamically turned into the appropriate SQL at runtime. Features of the expression framework include:

- Operators, such as =, !=, <, <=, >, >=, in, like, between
- Logical operators, such as AND, OR, NOT
- Traversing 1:1, 1:M, M:M relationships, joins (including outer joins on most databases)
- Object comparison
- Common database functions and user defined functions
- Keyword searching
- Query by Example

How can I setup a Class to query only the instances of the parent class but not the subclass instances?

By default reading a parent class will return all the instances of the parent and its subclasses. If only the instances of the parent class are desired, then the inheritance property "should read subclasses" can be turned off on the parent descriptor. If the subclass instances are not desired for a specific query, then the query expression can filter the children. The following gives an example of a subclass filtering expression:

```
Expression builder = new ExpressionBuilder();  
Expression expression = builder.getField("typeIndicatorField").equal("parent");
```

Can different ExpressionBuilders be used to generate sub expressions for a single expression?

No, doing this might seem to be working in some cases but would certainly fail in expressions using joins or complex expressions. The recommended way is to always use the same ExpressionBuilder in a single expression.

If your expression contains sub-selects, then the sub-query may have a different ExpressionBuilder, but this should be the only case.

Why does it take longer to read objects from the database the first time when compared to reading objects the second time?

This will happen when using a cache. When objects are read for the first time they are cached into the identity map. When the second read is done the cached objects are returned. This takes less time than constructing objects from the database. The caching is done based upon primary key. If a `ReadObjectQuery` is executed with an exact equal clause for all elements of the primary key then it may be possible to get a cache hit and not access the database at all.

How can stored procedures be used with TopLink?

Stored procedures are normally accessed through TopLink's custom SQL and stored procedure call support. The stored procedure can be called directly using the session API `executeSQL(...)` and `executeNonSelectingSQL()` or used in a descriptor's or mapping's custom SQL properties. Stored procedures with output parameters are supported through the `StoredProcedureCall` object. Stored procedure calls with output parameters can also be used inside the descriptor's and mapping's properties and executed directly on the session through the `DataReadQuery` and `DataModifyQuery` objects.

I keep getting multiple joins when I use multiple anyOf()s in my expression, which is returning too many results. How do I get it down to one join instead?

Essentially each time you use `anyOf()` in your expression, you will get another join in your database. For example, the TopLink expression:

```
emp.anyOf("managedEmployees").get("lastName").equal("Smith").and(emp.anyOf("managedEmployees").get("firstName").equal("John"));
```

will return you any `Employee` who manages a John or a Smith, not only John Smith. This is because a separate join is performed for each `anyOf()`. In order to force a single join, the same `anyOf` node must be used for each query:

```
Expression managedEmployees = emp.anyOf("managedEmployees");  
...  
managedEmployees.get("lastName").equal("Smith").and(managedEmployees.get("firstName").equal("John"));
```

What is the difference between a QueryByExample object and a selection object?

A `QueryByExample` does not replace a selection object in a query. A selection object has always been and continues to be an easy mechanism to query by the primary key only. Any other attributes in this object will be ignored. A `QueryByExample` object allows you to set up an object with various attributes set to give you better control over your query.

TopLink Mapping Workbench

Refer to the *Oracle9iAS TopLink Mapping Workbench Reference Guide* for detailed information on use the Mapping Workbench.

What is the difference between the TopLink Builder and the TopLink Mapping Workbench?

The TopLink Mapping Workbench is a completely new graphical mapping tool that replaces the TopLink Builder. It is written in Java, and can be used on any platform that supports the Java 2 runtime environment. It uses XML files to store TopLink Project information, and adds new functionality such as optimistic locking, transparent indirection, and enhanced database support. It also uses a multi-document interface, allowing the user to open multiple TopLink Projects simultaneously, and drag-and-drop descriptors and tables between Projects.

When I try to import classes into the TopLink Mapping Workbench, the dialog is empty. What's wrong?

The TopLink Mapping Workbench maintains its own CLASSPATH – separate from your system CLASSPATH. To import classes into your Mapping Workbench project, you must first indicate the directories that contain the domain class(es) to map on the project's **General** tab.

1. Choose the project object in the **Project Tree** pane.
2. Click on the **General** tab in the **Properties** pane.
3. To add a new classpath, click on **Add Entry** and select the directory, `.jar` file, or `.zip` file to add.

To add the system's classpath entries to the project, click on **Add System Entries**.

Note: The CLASSPATH containing your JDBC drivers should still be on your system CLASSPATH. The TopLink Mapping Workbench CLASSPATH is meant for domain classes only.

Why am I having trouble reading the .mwp file into my TopLink application?

Unlike the `.project` file used by earlier versions of TopLink and TopLink Mapping Workbench, the Mapping Workbench 4.6 `.mwp` file *cannot* be loaded directly into your code to create a TopLink Session. Instead, you must first deploy your Mapping Workbench project as either a single XML file or a Java class.

- **Deploying to an XML File**

Select **File > Generate Deployment XML** from the Mapping Workbench menu to create an XML file. After generating the file, use the following code to load the project into your application:

```
import oracle.toplink.sessions.Project;
import oracle.toplink.tools.workbench.XMLProjectReader;
...
Project p = XMLProjectReader.read("C:\\Projects\\Project Name.xml");
```

- **Deploying to a Java Class**

Select **File > Export to Java Source** from the Mapping Workbench menu to create a `.java` source file that you can then compile and instantiate in your code.

When I try to read a project, or create a new project, I am getting an XML exception. What could be causing this?

The TopLink Mapping Workbench uses an XML parser, found in `xerces.jar`, to read and write your project information. You may encounter an XML exception if a different XML parser appears first in your system or IDE CLASSPATH before `xerces.jar`, this can cause conflicts. Ensure that `xerces.jar` is the first XML parser in your CLASSPATH.

Alternatively, use the class loader (included with the TopLink SDK) to use the specific version of Xerces included with TopLink. Refer to the *Oracle9iAS TopLink: Foundation Library Guide* for more information.

Can I use older Mapping Workbench projects with the current version?

Yes. You can upgrade your previous version Mapping Workbench projects to use with the current version.

Caution: To upgrade from a previous version of TopLink, you **must** follow specific upgrade procedures and use the TopLink Package Renamer. Refer to the TopLink Release notes and *Oracle 9iAS TopLink Getting Started Guide* for more information.

Enterprise

What versions of CORBA does TopLink support?

TopLink's remote support for CORBA has been upgraded to the latest versions of the industry leading CORBA ORBs. TopLink's remote session and cache synchronization features now support deployment stubs and classes for Borland Inprise Visibroker 4.1, Iona Orbix 2000 and Sun's Java 2 ORB.

If I am using BEA WebLogic or IBM WebSphere do I need the specialized TopLink CMP foundation libraries?

It is important to note that the TopLink CMP for BEA WebLogic or TopLink CMP for IBM WebSphere Foundation Libraries are extensions to the TopLink for Java Foundation Library. If your application architecture involves using session beans with regular Java objects, or if you will be implementing Bean-Managed Persistence (BMP) then you can use the standard TopLink for Java Foundation Library. For these kinds of architectures you can use the TopLink for Java Foundation Library with any enterprise Java application server.

The TopLink for Java Foundation Library includes the classes necessary to synchronize with JTS for BEA WebLogic Server, IBM WebSphere Application Server, Oracle8i, and Inprise Application Server. The TopLink CMP for BEA WebLogic and IBM WebSphere Foundation Libraries allow you to use Container-Managed Persistence (CMP) for your entity beans.

Can TopLink interact with non-relational data sources, such as XML?

Yes, the TopLink SDK allows you to extend TopLink to access objects stored in non-relational data stores. TopLink also comes with an XML extension that allows you to easily read and write your objects from XML files. This extension is used by the TopLink Mapping Workbench to read and write your TopLink Projects.

To take advantage of the SDK you will need to develop a number of classes that can be used by TopLink to access your particular non-relational data store. You will also need to take advantage of a number of new TopLink mappings and use many of the customization hooks provided by TopLink that are not used by a “typical” TopLink application. The new mappings include:

- SDKAggregateObjectMapping
- SDKAggregateCollectionMapping
- SDKDirectCollectionMapping
- SDKObjectCollectionMapping

The XML extension is a simple example of how to implement the classes required by the TopLink SDK to access a non-relational data store. By default, the XML extension reads and writes your objects from XML documents located in native files. But it is flexible enough that you can plug in your own classes that get the XML documents from elsewhere (e.g. a messaging service, a URL). There are numerous other areas where you can plug in your own classes to modify the behavior of either the SDK or the XML extension.

Entity Deployment

This section discusses some of the general troubleshooting issues surrounding entity bean configuration and deployment, and lists many of the common exceptions and error messages that you may run across when attempting to deploy and persist entity beans using the Oracle 9iAS TopLink Foundation Library.

If you encounter any problems installing TopLink, using the Mapping Workbench, or require more information on any runtime exceptions generated by TopLink, consult the appropriate documentation.

Generating Deployment JARs

If you experience trouble generating the JARs for deployment,

- Verify that all environment entries (CLASSPATH, etc.) are configured properly.
- Identify which step of the build is failing (copying, compiling, running EJB compiler, etc.)

Running the EJB compiler utility involves several processes (compiling, code-generation, EJB compliance verification, compiling RMI stubs by running `rmic`). If an error occurs during the running of the EJB compiler utility, try to determine which stage may be causing the failure. Refer to the application server documentation for more information.

Common BEA WebLogic Deployment Exceptions

Following are some of the most common errors encountered when deploying to a BEA WebLogic applications server. Refer to ["Common BEA WebLogic 6.1 Exceptions"](#) on page 5-10 and ["Common WebLogic 7.0 Exceptions"](#) on page 5-14 for version-specific information.

Assertion Error

```
weblogic.utils.AssertionError: ***** ASSERTION FAILED ***** [
  Could not load class
  'oracle.toplink.internal.ejb.cmp.wls.WlsCMPDeployer':
  java.lang.ClassNotFoundException:
  oracle.toplink.internal.ejb.cmp.wls.WlsCMPDeployer
ERROR: ejbc found errors
```

Action: This error occurs if the `tl_wlsx.jar` file is not properly set on your classpath. The `tl_wlsx.jar` should be located in the `<INSTALL_DIR>/wls_cmp` folder and should be specified on your system classpath.

Error Deploying Application

Cause: A `DeploymentException` has occurred.

Action: Refer to the specific error code. The error code appears in the square brackets in the exception message, such as `[TopLink-8001]`). These errors may refer to errors in the specification of the project location reading in the properties file or validation errors due to improper mappings.

Exception 8001

```
<Error> <J2EE> <Error deploying application Account:
Unable to deploy EJB: AccountBean from Account.jar:
LOCAL EXCEPTION STACK:
EXCEPTION [TOPLINK-8001] (TopLink (WLS CMP) - X.X.X):
oracle.toplink.ejb.DeploymentException
EXCEPTION DESCRIPTION: No TopLink project was specified for
this bean.
    at
oracle.toplink.ejb.DeploymentException.noProjectSpecified(
DeploymentException.java:132) at
oracle.toplink.internal.ejb.cmp.ProjectDeployment.readProject(
ProjectDeployment.java:378)
```

Action: This error can occur if the location of the TopLink Project file for the bean is not properly specified. Please double check the file name as it is specified in the `toplink-ejb-jar.xml.xml` file, and the actual location of the project file on the file-system.

Exception 8016

```
<Error> <J2EE> <Error deploying application Account:
Unable to deploy EJB: AccountBean from Account.jar:
LOCAL EXCEPTION STACK:
EXCEPTION [TOPLINK-8016] (TopLink (WLS CMP) - X.X.X):
oracle.toplink.ejb.DeploymentException
EXCEPTION DESCRIPTION: An error occurred while setting up the
project: [java.io.FileNotFoundException: Account.xml]
INTERNAL EXCEPTION: java.io.FileNotFoundException: Account.xml
    at
oracle.toplink.ejb.DeploymentException.errorCreatingProject (Un
known Source)
```

Action: This error can occur if the location of the TopLink Project file for the bean is not properly specified. Please double check the file name as it is specified in the `toplink-ejb-jar.xml.xml` file, and the actual location of the project file on the file-system.

Cannot Startup Connection Pool

```
<Error> <JDBC> <Cannot startup connection pool "ejbPool"
weblogic.common.ResourceException: Cannot load driver class:
org.hsqldb.jdbcDriver>
...
```

Action: An error has occurred in setting up the connection pool. Check the nested SQL exception to determine the cause of the error. Typical problems include:

- The driver is not on the classpath.
- The user or password is incorrect.
- The database server url or driver name is not properly specified.

Please consult the BEA WebLogic documentation and your JDBC Driver documentation for help on the specific error raised by BEA WebLogic.

Error Message

```
weblogic.utils.AssertionError: ***** ASSERTION FAILED ***** [ Could
not create an instance of class 'null':
java.lang.NullPointerException
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:120)
at weblogic.ejb20.persistence.PersistenceType.
loadClass(PersistenceType.java:309)
```

Action: This problem occurs if using the GA version of BEA WebLogic Server 6.0. Service Pack 1 (SP1) of BEA WebLogic Server 6.0 is required to use TopLink CMP for BEA WebLogic. You must upgrade to at least SP1.

EJBC Found Errors

```
ERROR: ejbc found errors
Error from ejbc: Error while loading persistence resource
TopLink_CMP_Descriptor.xml Make sure that the persistence type
is in your classpath.
```

Cause: This error occurs if the `tl_wlsx.jar` file is not properly set on your CLASSPATH.

Action: The `tl_wlsx.jar` should be located in the `<INSTALL_DIR>/wls_cmp/lib` directory, and should be specified on your system CLASSPATH.

EJB Deployment Exception

```
weblogic.ejb20.EJBDeploymentException: Error Deploying CMP EJB;;
nested exception is: weblogic.ejb20.cmp.rdbms.RDBMSException:
An error occurred setting up the project:
EXCEPTION [TOPLINK-13000] (vX.X [TOPLink for WebLogic X.X]
JDK1.2): oracle.toplink.xml.XMLDataStoreException
EXCEPTION DESCRIPTION: File not found...
```

Action: This error occurs if the location of the TopLink Project file for the bean is not properly specified. Please double check the file name as it is specified in the `toplink-ejb-jar.xml` file, and the actual location of the TopLink project file on the file system.

Deploying EJB Component

```
Error deploying EJB Component: ...
    weblogic.ejb20.EJBDeploymentException: Exception in EJB
    Deployment; nested exception is:
Error while deploying bean..., File ... Not Found at
weblogic.ejb20.persistence.PersistenceType.setup
Deployer(PersistenceType.java:273)
```

Action: A typical cause of this error is that the `toplink-ejb-jar.xml` file is referring to a local DTD file using a file name or location that is incorrect. Please verify that all XML files refer to valid DTD files and locations.

Cannot Startup Connection Pool `ejbPool`

```
Cannot startup connection pool "ejbPool"
    weblogic.common.ResourceException:
Could not create pool connection. The DBMS driver exception
was:
...
```

Action: An error has occurred in setting up the connection pool. Check the nested SQL exception to determine the cause of the error. Typical problems include:

- The driver is not on the CLASSPATH.
- The username or password is incorrect.
- The database server URL or driver name is not properly specified.

Please consult the BEA WebLogic documentation and your JDBC driver documentation for help on the specific error raised by BEA WebLogic.

Other Errors

Occasionally, changes made to the server's configuration (`config.xml`) do not appear to be applied when the server is restarted. If this occurs, try removing the temp directories created by BEA WebLogic. These can be found under the `wlserver6.1` directory, at the same level as the config folder.

Problems on IBM WebSphere Server Startup or IBM WebSphere Bean Deployment

This section lists some of the common exceptions and errors that can occur when deploying beans in IBM WebSphere Application Server.

Please consult the IBM WebSphere Application Server documentation for more information on deploying beans.

When the IBM WebSphere Server is started, it attempts to deploy the jar files that are specified for deployment within the application server.

Errors that occur when the server is started are usually configuration problems that involve CLASSPATH issues, environment variable configuration, and database login configuration. Review the IBM WebSphere Server documentation on starting the server, and TopLink for Java Foundation Library's *Installing, Configuring and Testing*.

Some of the exceptions and errors that can be encountered when running the IBM WebSphere Server are listed in this section, along with their possible causes and recommended solutions.

Class Not Found Exceptions

Cause: The class not found is not included on the WebSphere application extensions classpath or in the EJB or WAR module.

Action: Ensure that all required classes are included in the correct location. Refer to the "Setting classpaths" information in the *IBM WebSphere InfoCenter* for detailed information on classpath locations.

Cause: The required TopLink JARs have not been copied into the application extensions classpath.

Action: Ensure that `toplink.jar`, `tl_wasx.jar`, `antlr.jar` are copied into the `<WebSphere install>\lib\app` directory. For running the examples, `hsqldb.jar` and `hsqldb_ds.jar` must also be copied into the `<WebSphere install>\lib\app` directory.

`oracle.toplink.exceptions.DatabaseException`

Cause: A TopLink Exception has occurred.

Action: Refer to "[Runtime and Development Exceptions](#)" on page 1-1 and look up the error code that is observed (the error code should be displayed in square brackets, such as [TopLink-1016]). Errors observed here may be errors in reading in the properties file, or validation errors due to improper mappings.

Exception [6066]

oracle.toplink.exceptions.QueryException: The object <Object> of class <class> with identity hashcode <hashcode> is not from this unit of work object space but the parent session's. The object was never registered in this unit of work, but read from the parent session and related to an object registered in the unit of work. Ensure that you are correctly registering your objects. If you are still having problems, you can use the `UnitOfWork.validateObjectSpace()` method to help debug where the error occurred. Please see the manual and FAQ for more information.

Cause: A bean was created outside of a transaction and then a second was created either in or out of a transaction.

Action: Ensure that all creates are done with in the context of a transaction.

Cause: The bean was not cleared out during `ejbPassivate`.

Action: Ensure that the `ejbPassivate` clears out the bean.

Cause: A bean-to-object relationships is not privately owned.

Action: Ensure that bean-to-object relationships are privately owned.

Exception [7064]

oracle.toplink.exceptions.ValidationException: Exception occurred in reflective EJB bean primary key extraction, please ensure your primary key object is defined correctly: key = 301, bean = <beanName>

Cause: An incorrect primary key object is being used with a bean.

Action: Ensure that you are using the correct primary key object for a bean.

Exception [7066]

`oracle.toplink.exceptions.ValidationException`: Cannot create or remove beans unless a JTS transaction is present, bean=<bean>

Cause: An attempt was made to create or remove a bean outside of a transaction.

Action: Ensure that all removing and creating of beans is done within a transaction.

Exception [7068]

`oracle.toplink.exceptions.ValidationException`: The project class <projectclass> was not found for the <toplink_session_name> using default class loader.

Cause: The project class specified in the `TOPLink.properties` file for the session specified on the **toplink_session_name environment** variable can not be found.

Action: Check that the project class given in the exception is on the WebSphere dependent classpath.

Exception [7069]

`oracle.toplink.exceptions.ValidationException`: An exception occurred looking up or invoking the project amendment method, <amendmentMethod> on the class <amendmentClass>;

Cause: An amendment method was called but can not be found.

Action: Check that the required amendment method exists on the class specified.

Exception [7070]

`oracle.toplink.exceptions.ValidationException`: A `TOPLink.properties` resource bundle must be located on the classpath in a `TOPLink` directory.

Cause: The `TOPLink.properties` file can not be found.

Action: Make sure that the location of the `TOPLink.properties` files is on the classpath. If the exception is within VisualAge the `TOPLink.properties` file must be in the project resources for the `TopLink` project.

Exception [7079]

EXCEPTION DESCRIPTION: The descriptor for [<bean class>] was not found in the session [<session name>]. Check the project being used for this session.

Cause: The descriptor listed was not found in the session specified on the deployment descriptor.

Action: Check that the project specified in the `toplink-ejb-jar.xml` file is the desired project. Also check that the project includes a descriptor for the missing bean class.

Exception [7101]

No "meta-inf/toplink-ejb-jar.xml" could be found in your classpath. The CMP session could not be read in from file.

Cause: The `toplink-ejb-jar.xml` file was not found.

Action: Make sure that the `toplink-ejb-jar.xml` file is located in the deployed `ejb jar` file under the `meta-inf` directory.

Exception [9002]

EXCEPTION [TOPLINK-9002] (TopLink - X.X.X):
`oracle.toplink.exceptions.SessionLoaderExceptionEXCEPTION`
DESCRIPTION: Unable to load Project class [<project class>].

Cause: The project class specified for the session in the `toplink-ejb-jar.xml` file cannot be found.

Action: Check that the project class has been included in the deployed jar with the entity beans.

Problems Occurring at Runtime Using TopLink for WebSphere Foundation Library

This section lists some of the common exceptions and errors that can occur at runtime when using the TopLink CMP for IBM WebSphere Application Server.

Exception [6026]

`oracle.toplink.exceptions: Query is not defined`

Cause: A required named query does not exist.

Action: Implement the named query. The stacktrace of the exception will contain the finder that failed.

Common WebSphere Exceptions

This section lists common exceptions and errors that may occur when running the TopLink for IBM WebSphere Foundation Library Deploy Tool.

Class Not Found Exceptions

Cause: The class specified was not found; it is not included on the deploy tool classpath or the system classpath.

Action: Ensure that all required classes are included on the correct classpath. Refer to the *IBM WebSphere Getting Started* for detailed information regarding classpath setup.

Note: The Deploy Tool calls external IBM classes to generate deployed code. Any exceptions thrown from these classes will be described on `System.out`. Check the **Tracing** button to view the most detailed information possible.

Common BEA WebLogic 6.1 Exceptions

Following are a few of the most common errors you may encounter when deploying `.jars` with TopLink and BEA WebLogic 6.1.

Development Exceptions

Missing Persistence Type

```
ERROR: Error from ejbc: Persistence type 'TopLink_CMP_2_0' with
      version 'X.X which is referenced in bean 'Account' is not
      installed. The installed persistence types are: (WebLogic_CMP_
      RDBMS, 6.0), (WebLogic_CMP_RDBMS, 5.1.0).
```

```
ERROR: ejbc found errors
```

Cause: There is no entry in the `persistence.install` file for TopLink CMP. This may occur if to TopLink installation was interrupted or a BEA WebLogic service pack was applied.

Action: In the `<WebLogic InstallDir>/wlserver6.1/lib/persistence` directory, edit the `persistence.install` file to add a new line `TopLink_CMP_Descriptor.xml` or replace your existing `persistence.install` file with the version of the file in the `<INSTALL_DIR>/wls_cmp` folder.

Error Loading Persistence Resource

Error while loading persistence resource `TopLink_CMP_Descriptor.xml` Make sure that the persistence type is in your classpath.

Cause: The `tl_wlsx.jar` file is not properly set in your classpath.

Action: Ensure that the classpath includes the `<INSTALL_DIR>/wls_cmp/core/lib/tl_wlsx.jar` file.

Wrong BEA WebLogic Version

```
C:\<INSTALL_DIR>\Examples\wls61\
  examples\ejb\cmp20\singlebean\Account.java:10: cannot
  resolve symbol
symbol   : class EJBLocalObject
location: interface examples.ejb.cmp20.singlebean.Account
public interface Account extends EJBLocalObject {
```

Cause: You are trying to compile using BEA WebLogic 6.0.

Action: Compile using BEA WebLogic 6.1.

Deployment/runtime Exceptions

Missing Persistence Type

Persistence type 'TopLink_CMP_2_0' with version 'X.X which is referenced in bean 'Account' is not installed. The installed persistence types are: (WebLogic_CMP_RDBMS, 6.0), (WebLogic_CMP_RDBMS, 5.1.0).

Cause: There is no entry in the `persistence.install` file for TopLink CMP. This may occur if to TopLink installation was interrupted or a BEA WebLogic service pack was applied.

Action: In the `<WebLogic InstallDir>/wlserver6.1/lib/persistence` directory, edit the `persistence.install` file to add a new line `TopLink_CMP_`

Descriptor.xml, or replace your existing persistence.install file with the version of the file in the <INSTALL_DIR>/wls_cmp folder.

Error Loading Persistence Resource

```
<DATE and TIME> <Error> <J2EE> <Error deploying application ejb20_
  cmp_order:
```

```
Unable to deploy EJB: C:\<INSTALL_
  DIR>\examples\wls61\server\.\config\TopLink_
  Domain\applications\.\wlnotdelete\wlap64280\ejb20_cmp_order.jar
  from ejb20_cmp_order.jar:
```

```
Error while loading persistence resource TopLink_CMP_
  Descriptor.xml Make sure that the persistence type is in your
  classpath.
```

```
at
```

```
weblogic.ejb20.persistence.InstalledPersistence.initialize(Ins
  talledPersistence.java:214)
```

```
at
```

```
weblogic.ejb20.persistence.InstalledPersistence.getInstalledTy
  pe(InstalledPersistence.java:113)
```

Cause: The tl_wlsx.jar file is not properly set in your classpath.

Action: Ensure that the classpath includes the <INSTALL_DIR>/wls_ cmp/lib/tl_wlsx.jar file.

Wrong Persistence Version

```
DATE and TIME> <Error> <J2EE> <Error deploying application ejb20_
  cmp_account:
```

```
Unable to deploy EJB: Account from ejb20_cmp_account.jar:
```

```
java.lang.AbstractMethodError
```

```
at
```

```
weblogic.ejb20.deployer.ClientDrivenBeanInfoImpl.deploy(Client
  DrivenBeanInfoImpl.java:807)
```

```
at
```

```
weblogic.ejb20.deployer.Deployer.deployDescriptor(Deployer.jav
  a:1234)
```

```

    at
weblogic.ejb20.deployer.Deployer.deploy(Deployer.java:947)
    at
weblogic.j2ee.EJBComponent.deploy(EJBComponent.java:30)

```

Cause: You may be using a <persistence-version> meant for BEA WebLogic 7.0.

Action: Use a <persistence-version> of 4.0.

Cannot Startup Data Source

```

EXCEPTION [TOPLINK-7060] (TopLink (WLS
    CMP)-X.X):oracle.toplink.exceptions.ValidationException
EXCEPTION DESCRIPTION: Cannot acquire data source
[jdbc/ejbNonJTSDDataSource].

INTERNAL EXCEPTION: javax.naming.NameNotFoundException: Unable
to resolve jdbc.ejbNonJTSDDataSource Resolved: ''
Unresolved:'jdbc' ; remaining name 'ejbNonJTSDDataSource'

```

Cause: An error has occurred in setting up the datasource.

Action: Check the nested SQL exception to determine the cause of the error. See "[Error code: 7060](#)" for more information. Consult the BEA WebLogic documentation and your JDBC Driver documentation for help on the specific error raised by WebLogic.

Wrong WebLogic Version

```

<DATE and TIME> <Error> <Management> <Error parsing XML
    descriptor for application TopLink_Domain:Name=ejb20_cmp_
    account, Type=Application
weblogic.xml.process.ProcessorFactoryException: Could not
locate processor for public id = "-//Sun Microsystems,
Inc.//DTD J2EE Application 1.3//EN"
    at
weblogic.xml.process.ProcessorFactory.getProcessor(ProcessorFa
ctory.java:181)
    at

```

```
weblogic.xml.processor.ProcessorFactory.getProcessor(ProcessorFactory.java:164)
```

Cause: CaYou are trying to compile using WebLogic 6.0.

Action: Compile using WebLogic 6.1.

Common WebLogic 7.0 Exceptions

Following are a few of the most common errors you may encounter when deploying .jars with TopLink and WebLogic 7.0.

Development-time Exceptions:

Missing Persistence Type

```
ERROR: Error from ejbc: Persistence type 'TopLink_CMP_2_0' with version 'X.0 which is referenced in bean 'Account' is not installed. The installed persistence types are: (WebLogic_CMP_RDBMS, 6.0), (WebLogic_CMP_RDBMS, 5.1.0), (WebLogic_CMP_RDBMS, 7.0).
```

```
Persistence type 'TopLink_CMP_2_0' with version 'X.0 which is referenced in bean 'Account' is not installed. The installed persistence types are: (WebLogic_CMP_RDBMS, 6.0), (WebLogic_CMP_RDBMS, 5.1.0), (WebLogic_CMP_RDBMS, 7.0)
```

```
ERROR: ejbc found errors
```

Cause: There is no entry in the persistence.install file for TopLink CMP. This may occur if to TopLink installation was interrupted or a WebLogic service pack was applied.

Action: In the `<WebLogic InstallDir>/weblogic700b/lib/persistence` directory, edit the persistence.install file to add a new line `TopLink_CMP_Descriptor.xml` or replace your existing persistence.install file with the version of the file in the `<INSTALL_DIR>/wls/cmp` folder.

Error Loading Persistence Resource

```
ERROR: Error from ejbc: Error while loading persistence resource TopLink_CMP_Descriptor.xml
```

```
Make sure that the persistence type is in your classpath.
```

```
ERROR: ejbc found errors
```

or

ERROR: at

```
weblogic.ejb20.persistence.InstalledPersistence.initialize(InstalledPersistence.java:214)
```

at

```
weblogic.ejb20.persistence.InstalledPersistence.getInstalledType(InstalledPersistence.java:113)
```

at

```
weblogic.ejb20.deployer.MBeanDeploymentInfoImpl.getPersistenceType(MBeanDeploymentInfoImpl.java:584)
```

Cause: The `tl_wlsx.jar` file is not properly set in your classpath.

Action: Ensure that the classpath includes the `<ISNTALL_DIR>/wls_cmp/lib/tl_wlsx.jar` file.

Wrong WebLogic Version

ERROR: Error processing 'META-INF/weblogic-ejb-jar.xml': The public id, "-//BEA Systems, Inc.//DTD WebLogic 7.0.0 EJB//EN", specified in the XML document is invalid. Use one of the following valid public ids:

```
"-//BEA Systems, Inc.//DTD WebLogic 5.1.0 EJB//EN"
```

```
"-//BEA Systems, Inc.//DTD WebLogic 6.0.0 EJB//EN"
```

ERROR: ejbc found errors

Cause: You are trying to compile using WebLogic 6.1.

Action: Compile using WebLogic 7.0.

Deployment/runtime Exceptions:

Missing Persistence Type

Error from ejbc: Persistence type 'TopLink_CMP_2_0' with version 'X.0' which is referenced in bean 'Account' is not installed. The installed persistence types are: (WebLogic_CMP_RDBMS, 6.0), (WebLogic_CMP_RDBMS, 5.1.0), (WebLogic_CMP_RDBMS, 7.0).

Persistence type 'TopLink_CMP_2_0' with version 'X.0' which is referenced in bean 'Account' is not installed. The installed persistence types are: (WebLogic_CMP_RDBMS, 6.0), (WebLogic_CMP_RDBMS, 5.1.0), (WebLogic_CMP_RDBMS, 7.0)

Cause: There is no entry in the persistence.install file for TopLink CMP. This may occur if to TopLink installation was interrupted or a WebLogic service pack was applied.

Action: In the <WebLogic InstallDir>/weblogic6.1/lib/persistence directory, edit the persistence.install file to add a new line TopLink_CMP_Descriptor.xml or replace your existing persistence.install file with the version of the file in the <INSTALL_DIR>/wls_cmp folder.

Error Loading Persistence Resource

```
java.lang.NullPointerException
    at
weblogic.ejb20.deployer.EJBDeployer.deactivate(EJBDeployer.java:1513)
    at
weblogic.ejb20.deployer.EJBDeployer.undeploy(EJBDeployer.java:301)
    at
weblogic.ejb20.deployer.Deployer.deploy(Deployer.java:875)
    at
weblogic.j2ee.EJBComponent.deploy(EJBComponent.java:70)
```

Cause: The tl_wlsx.jar file is not properly set in your classpath.

Action: Ensure that the classpath includes the <INSTALL_DIR>/wls_cmp/lib/tl_wlsx.jar file.

Cannot Startup Data Source

```
EXCEPTION [TOPLINK-7060] (TopLink (WLS CMP) -
X.X.X): oracle.toplink.exceptions.ValidationException
EXCEPTION DESCRIPTION: Cannot acquire data source
[jdbc/ejbNonJTSDatasource].
```

```
INTERNAL EXCEPTION: javax.naming.NameNotFoundException: Unable
to resolve jdbc.ejbNonJTSDatasource Resolved: ''
Unresolved:'jdbc' ; remaining name 'ejbNonJTSDatasource'
```

Cause: An error has occurred in setting up the datasource.

Action: Check the nested SQL exception to determine the cause of the error. See "[Error code: 7060](#)" for more information. Consult the BEA WebLogic documentation and your JDBC Driver documentation for help on the specific error raised by WebLogic.

Index

B

BLOB data, 4-9
builder exceptions, error codes, 2-36
BuilderException, 1-2

C

cache, object, 4-13
class libraries, 4-2
CLASSPATH, 4-2, 4-19
code generation, 4-3
common deployment errors, 5-1
CommunicationException, 1-3
concurrency exceptions, error codes, 2-45
ConcurrencyException, 1-3
conversion exceptions, error codes, 2-46
ConversionException, 1-3
CORBA, supported, 4-21

D

database drivers, 4-3
database exceptions, error codes, 2-47
DatabaseException, 1-4
DatabaseSession, 4-12
deployment errors, solutions, 5-1
deployment JARs, generating, 5-1
descriptor exceptions, error codes, 2-1
DescriptorException, 1-4
development components, TopLink, 4-2
development exceptions
 BuilderException, 1-2
 ConcurrencyException, 1-3

 ConversionException, 1-3
 DescriptorException, 1-4
 QueryException, 1-5
 ValidationException, 1-6
direct mappings, 4-4
domain model, inheritance, 4-11

E

EJBQL, supporting, 4-16
environment, configuring, 4-2
error codes and descriptions, 2-1
error messages
 Mapping Workbench, 3-1
exceptions
 about, 1-1
 Builder Exceptions, error codes, 2-36
 BuilderException, 1-2
 CommunicationException, 1-3
 Concurrency Exceptions, error codes, 2-45
 ConcurrencyException, 1-3
 Conversion Exceptions, error codes, 2-46
 ConversionException, 1-3
 Database Exceptions, error codes, 2-47
 DatabaseException, 1-4
 Descriptor Exceptions, error codes, 2-1
 DescriptorException, 1-4
 development, 1-1
 java.security.AccessControlException, 5-6, 5-8
 Optimistic Lock Exceptions, error codes, 2-50
 OptimisticLockException, 1-5
 Query Exceptions, error codes, 2-52
 QueryException, 1-5
 runtime, 1-1

- TopLinkException, 1-6
- TopLinkException class, 1-1
- Validation Exceptions, error codes, 2-67
- ValidationException, 1-6
- ExpressionBuilder, 4-17
- expressions, supported, 4-17

F

- foreign key reference, 3-2
- frequently asked questions (FAQs), 4-1

G

- generating code, 4-3
- generating deployment JARs, 5-1

I

- identity map, 4-13
- identity, object, 4-12
- importing classes in Mapping Workbench, 4-19
- indirection
 - transparent, 3-1, 4-11
 - value holder, 3-1
- inheritance, 4-7, 4-11
- inserting objects, 4-13

J

- jars
 - common deployment errors, 5-1
- Java class
 - generating, 4-4
 - private and protected fields, 4-5
- Java types, supported, 4-3
- java.security.AccessControlException, 5-6, 5-8
- JDBC drivers, 4-3, 4-14
- JVM, 4-2

K

- key pair
 - database table, 3-2

L

- .lax file, 3-13
- locking policy, 3-2

M

- Mapping Workbench
 - about, 4-19
 - error messages, 3-1
 - upgrading projects, 4-21
- mappings, supported, 4-4
- messages, error, 2-1

N

- neediness error messages, 3-1
- non-relational data sources, using, 4-21

O

- object
 - caching, 4-13
 - insert and update, 4-13
 - instantiating, 4-3
 - maintaining identity, 4-12
 - model, 4-3
- object-relational mappings, 4-5
- optimistic lock exceptions, error codes, 2-50
- optimistic locking, 1-5
- OptimisticLockException, 1-5

P

- PATH, 4-2
- private fields, accessing, 4-5
- projects
 - upgrading from 2.x or 3.x, 4-21
- protected fields, accessing, 4-5

Q

- query exceptions, error codes, 2-52
- QueryByExample, 4-19
- QueryException, 1-5
- querying

- parents and subclasses, 4-17
- supported expressions, 4-17

R

- registering `UnitOfWork`, 4-15
- relationship mappings, 4-5
- renamer, project, 4-21
- runtime components
 - `TopLink`, 4-2
- runtime exceptions, 1-1
 - `CommunicationException`, 1-3
 - `DatabaseException`, 1-4
 - `OptimisticLockException`, 1-5

S

- SDK mappings, 4-5
- sequencing, 4-7, 4-11
- `ServerSession`, 4-12
- `SQLException`, 1-4
- stored procedures, 4-18

T

- table qualifiers and creators, 4-13
- `tools.jar`, 4-2
- `TopLink`
 - about, 4-1
 - components, 4-2
 - configuring your environment, 4-2
 - generating code, 4-3
 - mappings, 4-4
- `toplink_core.jar`, 4-2
- `TopLinkException`, 1-6
- transparent indirection, 3-1, 4-11

U

- `UnitOfWork`
 - register, 4-15
- updating objects, 4-13
- upgrading Mapping Workbench projects from prior versions, 4-21

V

- validation exceptions, error codes, 2-67
- `ValidationException`, 1-6
- value holder indirection, 3-1

W

- `WebLogic`, 4-4, 4-21
- `WebSphere`, 4-4, 4-21

X

- XML exceptions, in Mapping Workbench, 4-20

