



---

# Teradata Aster Big Analytics Appliance 3H

Database User Guide

Release 5.10.00.01

B700-7012-510K

September 2013

The product or products described in this book are licensed products of Teradata Corporation or its affiliates.

Teradata, Active Data Warehousing, Active Enterprise Intelligence, Applications-Within, Aprimo, Aprimo Marketing Studio, Aster, BYNET, Claraview, DecisionCast, Gridscale, MyCommerce, Raising Intelligence, Smarter. Faster. Wins., SQL-MapReduce, Teradata Decision Experts, "Teradata Labs" logo, "Teradata Raising Intelligence" logo, Teradata ServiceConnect, Teradata Source Experts, "Teradata The Best Decision Possible" logo, The Best Decision Possible, WebAnalyst, and Xkoto are trademarks or registered trademarks of Teradata Corporation or its affiliates in the United States and other countries.

Adaptec and SCSISelect are trademarks or registered trademarks of Adaptec, Inc.

AMD Opteron and Opteron are trademarks of Advanced Micro Devices, Inc.

Apache, Apache Hadoop, Hadoop, and the yellow elephant logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries.

Apple, Mac, and OS X all are registered trademarks of Apple Inc.

Axeda is a registered trademark of Axeda Corporation. Axeda Agents, Axeda Applications, Axeda Policy Manager, Axeda Enterprise, Axeda Access, Axeda Software Management, Axeda Service, Axeda ServiceLink, and Firewall-Friendly are trademarks and Maximum Results and Maximum Support are servicemarks of Axeda Corporation.

Data Domain, EMC, PowerPath, SRDF, and Symmetrix are registered trademarks of EMC Corporation.

GoldenGate is a trademark of Oracle.

Hewlett-Packard and HP are registered trademarks of Hewlett-Packard Company.

Hortonworks, the Hortonworks logo and other Hortonworks trademarks are trademarks of Hortonworks Inc. in the United States and other countries.

Intel, Pentium, and XEON are registered trademarks of Intel Corporation.

IBM, CICS, RACF, Tivoli, and z/OS are registered trademarks of International Business Machines Corporation.

Linux is a registered trademark of Linus Torvalds.

LSI is a registered trademark of LSI Corporation.

Microsoft, Active Directory, Windows, Windows NT, and Windows Server are registered trademarks of Microsoft Corporation in the United States and other countries.

NetVault is a trademark or registered trademark of Quest Software, Inc. in the United States and/or other countries.

Novell and SUSE are registered trademarks of Novell, Inc., in the United States and other countries.

Oracle, Java, and Solaris are registered trademarks of Oracle and/or its affiliates.

QLogic and SANbox are trademarks or registered trademarks of QLogic Corporation.

Quantum and the Quantum logo are trademarks of Quantum Corporation, registered in the U.S.A. and other countries.

Red Hat is a trademark of Red Hat, Inc., registered in the U.S. and other countries. Used under license.

SAS and SAS/C are trademarks or registered trademarks of SAS Institute Inc.

SPARC is a registered trademark of SPARC International, Inc.

Symantec, NetBackup, and VERITAS are trademarks or registered trademarks of Symantec Corporation or its affiliates in the United States and other countries.

Unicode is a registered trademark of Unicode, Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

**THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS-IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. IN NO EVENT WILL TERADATA CORPORATION BE LIABLE FOR ANY INDIRECT, DIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS OR LOST SAVINGS, EVEN IF EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.**

The information contained in this document may contain references or cross-references to features, functions, products, or services that are not announced or available in your country. Such references do not imply that Teradata Corporation intends to announce such features, functions, products, or services in your country. Please consult your local Teradata Corporation representative for those features, functions, products, or services available in your country.

Information contained in this document may contain technical inaccuracies or typographical errors. Information may be changed or updated without notice. Teradata Corporation may also make improvements or changes in the products or services described in this information at any time without notice.

To maintain the quality of our products and services, we would like your comments on the accuracy, clarity, organization, and value of this document. Please email: [teradata-books@lists.teradata.com](mailto:teradata-books@lists.teradata.com).

Any comments or materials (collectively referred to as "Feedback") sent to Teradata Corporation will be deemed non-confidential. Teradata Corporation will have no obligation of any kind with respect to Feedback and will be free to use, reproduce, disclose, exhibit, display, transform, create derivative works of, and distribute the Feedback and derivative works thereof without limitation on a royalty-free basis. Further, Teradata Corporation will be free to use any ideas, concepts, know-how, or techniques contained in such Feedback for any purpose whatsoever, including developing, manufacturing, or marketing products or services incorporating Feedback.

**Copyright © 2000-2013 by Teradata Corporation. All Rights Reserved.**

# Table of Contents

---

<b>Preface .....</b>	<b>15</b>
Conventions Used in This Guide .....	16
Typefaces .....	16
SQL Text Conventions .....	16
Command Shell Text Conventions .....	16
Contact Teradata Global Technical Support (GTS) .....	16
About Teradata Aster .....	17
About This Document .....	17
Revision History .....	17

---

## VOLUME 1 Aster Database User Guide

---

<b>Chapter 1: Tables .....</b>	<b>20</b>
Creating Tables.....	20
Fact Tables and Dimension Tables.....	21
Fact Tables.....	21
Dimension Tables .....	22
Example Fact and Dimension Tables .....	22
Another Example: The orders Fact Table .....	23
Table Persistence .....	25
Regular or Persistent Tables.....	25
Temporary Tables .....	25
Analytic Tables .....	25
Distribution and Logical Partitioning Compared.....	29
Table Distribution .....	31
Distribution Key .....	32
Logical Partitioning .....	34
Logical Partitioning Overview .....	34
Benefits of Logical Partitioning .....	34
Automatic Logical Partitioning .....	35
Create a Logically Partitioned Table .....	36

Inheritance with Automatic Logical Partitioning .....	47
Command Cascading with Automatic Logical Partitioning.....	47
More Detailed Logical Partitioning Examples .....	48
Performance and Automatic Logical Partitioning.....	49
Automatic Logical Partitioning FAQs.....	52
Columnar Tables .....	54
When to Use Columnar Tables .....	55
Recommendations When Using Columnar Tables.....	55
When to Avoid Using Columnar Tables.....	56
How Queries Change When Using Columnar Tables.....	56
Load Data into Columnar Tables with Logical Partitioning.....	58
Unsupported Operations on Columnar Tables .....	58

---

<b>Chapter 2: Aster Database Cluster Terminal (ACT) .....</b>	<b>60</b>
ACT Quick Start.....	60
Launch ACT .....	61
Launch ACT on Windows .....	61
Launch ACT on Linux or Solaris.....	62
Launch ACT on Mac.....	62
Launch ACT Directly on the Queen .....	62
Log In to ACT .....	62
Startup Parameters for ACT .....	63
Using the “on-error-stop” Option in ACT.....	66
Use a Configuration File to Pass ACT Startup Parameters.....	67
How to Use ACT .....	68
Issue SQL Queries .....	68
Exit ACT .....	71
Page Through Query Results .....	71
Throttle Query Results in ACT and Aster Database .....	71
ACT Utility Commands .....	74
Repeat Previously Typed Commands .....	74
Tab Completion .....	74
ACT Commands (at the SQL Prompt) .....	75
Database Parameters Set with \set .....	80
Troubleshooting ACT .....	81
ACT Connection Hangs When Using SSL .....	81
Invalid User Name Error in ACT After Password Change .....	81
Misleading Error Message Reports Problem With a Role Instead of With a User .....	81

---

<b>Chapter 3: SQL-MapReduce and Stream.....</b>	82
Introduction to SQL-MapReduce .....	83
What is MapReduce? .....	83
Aster Database SQL MapReduce.....	84
SQL-MapReduce Query Syntax .....	85
SQL-MR with Multiple Inputs .....	87
Benefits of Multiple Inputs.....	87
How Multiple Inputs are Processed .....	88
Types of SQL-MR Inputs.....	88
Semantic Requirements for SQL-MR Functions.....	89
Use Cases and Examples for Multiple Inputs.....	90
SQL-MR Multiple Input FAQ .....	96
SQL-MapReduce Java API .....	97
Prerequisites for Working with SQL-MapReduce in Java.....	97
Write a SQL-MapReduce Function in Java .....	98
Constructor for a SQL-MapReduce Function in Java.....	99
Operate Function: operateOnSomeRows() or operateOnPartition() .....	99
Argument Clauses for Java Functions.....	100
Test Java Functions Locally with TestRunner .....	101
Build and Package the SQL-MapReduce Function .....	101
SQL-MapReduce Examples in Java .....	102
SQL-MapReduce C API.....	104
Types of C Functions .....	104
Get and Unpack the SQL-MapReduce C SDK.....	105
Build the C API Examples and Tests.....	105
Build C API Functions .....	106
Write an SQL-MapReduce Function in C .....	107
Memory Management in C API Functions .....	109
Datatypes in the C API .....	109
Test C API Functions Locally with TestRunner.....	110
Install and Use a Sample Function .....	111
Prerequisites .....	111
Procedure.....	112
Run the Function .....	113
Manage SQL-MapReduce Execution.....	114
Start an SQL-MapReduce Job .....	114
Cancel an SQL-MapReduce Job .....	114
Debug SQL-MapReduce Job and Task Execution.....	115
Troubleshooting and SQL-MR Errors.....	116
Size Limit on Constant String .....	116

Stream API for Python, Perl, and Other Types of Scripts .....	117
Stream Function Usage.....	117
Examples .....	120
STREAM Script Execution Environment.....	122
Stream Behavior .....	123
Troubleshooting Script Crashing .....	124
Using the R Programming Language and Environment .....	125
Overview of R .....	125
Installing R on Aster Database.....	127
Installing R from a Local Repository.....	128
Installing Prerequisites for Optional R Packages .....	129
Execution Model For R.....	130
Writing an R Program to Run Inside the Aster Database .....	131
Data Type Mapping Between R and Aster Database .....	134
Troubleshooting .....	137
SQL-MapReduce Security.....	138
Schema Membership for Installed Functions and Files.....	138
GRANT Privileges for Installed Functions and Files .....	139
User Permissions for Installed Files and Functions.....	139
Upgrade earlier SQL-MapReduce functions to Aster Database 5.0.x .....	140
Manage Functions and Files in Aster Database.....	140
What can I install on the cluster?.....	140
Get information About Installed Functions .....	141
Checklist for Installing a Function .....	141
Test the Function .....	142
Install a File or Function.....	142
Make a Local Copy of a File or Function .....	143
Remove a File or Function .....	144
FAQs About SQL-MapReduce and Stream .....	144

---

## VOLUME 2 Aster Database Administrator Guide

---

<b>Chapter 1: Database Management .....</b>	<b>148</b>
Get Started .....	148
Bring Up a SQL Prompt.....	149
Create an Aster Database User.....	149
Basic Data Import .....	149
System Configuration.....	150

Create and Drop Databases.....	150
Create a Database .....	150
Drop a Database .....	151
Manage Tables .....	151
Table Design Guidelines.....	151
Create Tables Examples .....	154
Load Data Into Tables .....	156
Alter Tables .....	157
Drop Tables.....	159
Export Table Contents .....	159
Manage Database Objects .....	160
Browse database objects .....	161
Analyze Tables.....	161
Manage Integrity Constraints .....	161
Manage Indexes.....	164
Manage Space.....	167
Check Current Disk Capacity and Free Space .....	167
Disk Usage .....	167
Maintain Sufficient Free Disk Space .....	168
Dead Space.....	168
Compression.....	172
EXPLAIN Plan .....	173

---

## Chapter 2: Data Modeling Best Practices.....

Eight Fundamental Rules for Modeling Big Data in Aster Database .....	177
Introduction to Data Modeling in Aster Database .....	179
Distributed and Replicated Tables.....	179
Aster Database is Different.....	179
Aster Database's Three Principles for High Performance on Big Data .....	180
What You'll Learn in This Document: The Aster Database Way to Model Big Data...	180
Review: Eight Fundamental Rules for Big Data .....	180
Dimensionalize Your Schema.....	181
Summary .....	181
The star schema model.....	181
How to dimensionalize your data model .....	182
The retail1 example schema .....	182
Benefits of using star schemas in Aster Database.....	185
Use Columnar Tables When Appropriate .....	186
Distribute Your Data with Joins in Mind .....	187

Summary .....	187
Choosing a Distribution Key .....	187
Examples of Physically Distributed Schemas .....	188
Single-Column Distribution Key.....	189
Replicate Common, Frequently Joined Data.....	190
Summary .....	190
Replicated Dimension Tables in Aster Database .....	190
Very Large Dimension Tables Should Be Distributed, Not Replicated .....	191
Split Data Into Child Partitions .....	191
Summary .....	191
When to Use Logical Partitioning .....	192
When should I not use logical partitioning?.....	193
Benefits of Logical Partitioning .....	193
How Logical Partitioning Works.....	193
Tips for Creating Logically Partitioned Tables.....	194
Interaction of Logically Partitioned Tables and Aster Database Distribution Keys .....	194
Using a date/time column as both the distribution key and logical partitioning constraint.....	195
Date dimension precludes use of child partitions .....	195
Simulating the effect of logical partitioning with clustering.....	196
Verticalize Your Schema .....	197
Summary .....	197
When is verticalization useful?.....	197
Costs.....	198
How to verticalize your schema.....	198
Index Your Tables .....	198
Summary .....	198
When to use indexes in Aster Database .....	199
How indexes improve performance .....	199
Indexes can speed up nested loop and merge joins .....	199
Conditions that must be met in order to use an index scan.....	200
Notice! Enabling index scans can hurt performance.....	201
Logical partitioning as an alternative to indexes .....	201
Multicolumn indexes .....	202
Bitmap index scans (using multiple, single-column indexes) .....	202
Operational considerations when you use indexes .....	204
Tips and FAQs about indexes in Aster Database .....	204
Consider Using a Denormalized Data Model .....	205
Summary .....	205
How denormalization works in Aster Database.....	206
When to use denormalization .....	206
When not to use denormalization.....	206

Denormalization compared with other techniques.....	207
Data Modeling FAQ.....	207

---

## Chapter 3: Query Tuning Best Practices..... 208

Introduction .....	208
Who should read this guide? .....	208
Am I ready to tune my queries?.....	209
Three Principles for High Performance on Big Data .....	209
Networking: Thou shalt not move big data.....	209
Disk: Thou shalt not read irrelevant data. ....	209
Processor: Thou shalt not do redundant processing. ..	210
Top Tips for Analysts.....	210
Top Tips for DBAs.....	210
General Tips on Tuning .....	211
A good data model is the starting point .....	211
Use the AMC and Ganglia to find bottlenecks.....	211
Run ANALYZE regularly to ensure the most optimal query plans .....	212
Run VACUUM to ensure queries achieve best performance .....	212
Use a COMPRESS level that matches the frequency of data access .....	212
When updating or adding data to Aster Database, wrap statements in a transaction ..	213
Optimize Queries for Parallel Execution .....	214
Join tables on the distribution key for best performance .....	214
Rewrite subselect clauses as joins.....	216
Use GROUP BY and COUNT DISTINCT on the distribution key .....	217
Redistributed queries .....	221
Avoid per-row processing .....	222
Operations on native data types .....	222
Sorting vs. hashing .....	222
Favor GROUP BY over DISTINCT.....	223
Read the EXPLAIN Plan .....	223
Introduction to EXPLAIN .....	223
The EXPLAIN Plan.....	223
How does EXPLAIN work? .....	224
What happens behind the scenes? .....	224
Tips for using EXPLAIN effectively.....	224
Read an EXPLAIN plan .....	224
A simple example .....	225
EXPLAIN plan for a large query .....	226
Summary .....	227
A Step-by-Step Approach to Query Tuning .....	227

Eliminate unnecessary data movement.....	227
Examine scan-type operations.....	229
Examine join operations.....	231
Examine aggregation operations .....	231
Why does the planner think that? .....	232
Understand Table Joins.....	232
Hash joins .....	232
Merge Joins .....	233
Nested Loop Joins.....	234
Tuning join operations.....	235
Summary of join tuning tips .....	235
Use Linux Utilities to Isolate Query Performance Issues.....	236
System statistics: vmstat .....	236
CPU Monitoring with TOP .....	238
Find data skew.....	242
Find processing skew .....	242
Additional Resources.....	244
Checklist for Query Tuning .....	244

---

## **Chapter 4: User Management .....** 246

Manage Users and Privileges.....	246
Access Control.....	246
Default Roles and Users .....	246
User Name Rules.....	247
Password Rules .....	247
Add Users.....	248
Database Users and Roles.....	249
User Attributes .....	250
Set Up Read-Only Access for a User .....	252
User Authentication.....	252
Authentication Methods .....	252
Single Sign-On.....	253
Set Up LDAP Authentication.....	253
Set Up Active Directory Authentication with Single Sign-On .....	257
Active Directory Authentication .....	261
Return to Password-Only Authentication .....	263
Schema Search Path.....	263
Show Current Schema Search Path.....	264
Adding to the Schema Search Path .....	264
The search_path Variable.....	264

Setting the Schema Search Path for a Session or Transaction .....	265
Setting a User's Default Schema Search Path.....	265
System Internal Users .....	265
Change System Internal User Passwords .....	265

---

## **Chapter 5: Workload Management .....** 268

Overview.....	268
Admission Control.....	268
Predicate Evaluation.....	269
Admission Limits via the Command Line Interface (ncli) .....	270
Managing Concurrency .....	270
Configure Admission Limits using the AMC.....	271
Resource Management .....	271
Workload Settings via the Command Line Interface (ncli) .....	273
Service Classes.....	273
Workload Policies.....	275
Workload Configuration Example .....	281
Workload Policy Predicate Attributes.....	282
Workload Policy Predicate Syntax .....	284
Resource Allocation and Service Classes.....	285
Monitoring the Priority Execution of Queries using the AMC .....	286
Common Mistakes .....	286
Best Practices .....	288
Troubleshooting.....	289

---

## **Chapter 6: Export and Load Tools.....** 290

---

## **Chapter 7: Teradata-Aster Database Connector .....** 292

Introduction .....	292
Teradata-Aster Database Configuration Overview.....	293
Prerequisites .....	293
Configure networking.....	293
Hardware Best Practices .....	293
Building out the correct hardware configuration .....	293
Configuring for performance.....	295
Copying Data from Teradata to Aster Database .....	296

Syntax of load_from_teradata .....	296
Using load_from_teradata .....	296
Copying Data from Aster Database to Teradata .....	298
Syntax of load_to_teradata .....	299
Output of load_to_teradata .....	299
Using load_to_teradata .....	299
Connector Argument Clauses .....	301
Common Clauses .....	301
load_from_teradata Clauses .....	303
load_to_teradata Clauses .....	304
Datatype Support .....	305
Aster Database to Teradata datatype conversions .....	306
Teradata to Aster Database datatype conversions .....	307
Unsupported Datatypes .....	309
Converting Multibyte Characters to Unicode During a Load .....	310
AMC Support .....	310
Performance Tips and Best Practices .....	311
Interaction Between MAX_SESSIONS and NUM_INSTANCES .....	312
NoSpool Mode Limitations and Functionality .....	314
Troubleshooting .....	315
Troubleshooting common to both Teradata Connector functions .....	315
load_from_teradata Troubleshooting .....	316
load_to_teradata Troubleshooting .....	317
Limitations .....	318

---

<b>Chapter 8: SQL-H: The Hadoop/HCatalog Connector .....</b>	<b>320</b>
Introduction to SQL-H .....	320
Benefits of SQL-H .....	321
Configuring Aster Database and HCatalog .....	321
Prerequisites .....	321
Configure networking for SQL-H .....	322
Hardware configuration .....	324
Create a SQL-H configuration .....	324
The load_from_hcatalog SQL-MR Function .....	325
load_from_hcatalog Usage .....	325
Display the HCatalog Metadata in ACT .....	326
List databases in HCatalog .....	326
List tables or describe a table in HCatalog .....	327
Using SQL-H .....	329

Using load_from_hcatalog to Create a View .....	329
Partition Pruning with load_from_hcatalog .....	330
SQL-H Argument Clauses.....	331
load_from_hcatalog Clauses .....	331
Datatype and Storage Type Support.....	332
HCatalog to Aster Database datatype conversions .....	333
Tips for Working with SQL-H .....	336
Capturing output to a table .....	336
Troubleshooting.....	337
SQL-H Queries Fail After Upgrade.....	337
load_from_hcatalog Returns an Error .....	337
No SQL-H Configuration Defined .....	337
Concurrent SQL-H Queries Return a “Busy, please retry.” Error .....	337
Error Logs .....	338
Errors .....	338
Limitations .....	339
Support for Unsecured Hadoop Clusters Only .....	339
Specifying Partitions by Range.....	339
Special Character Handling Limitations in HCatalog.....	339

---

## **Chapter 9: Backup and Restoration.....**

Introduction to Aster Database Backup.....	342
Overview .....	342
Backup Philosophy .....	343
Backup Architecture .....	344
Managing the Backup Cluster.....	345
Start the Backup Cluster.....	345
Shut down the Backup Cluster .....	345
Restart the Backup Cluster.....	346
Check the Status of the Backup Cluster .....	346
Check the Availability and Capacity of the Backup Cluster .....	346
Troubleshoot an Unavailable Backup Node.....	346
Using Aster Database Backup.....	347
Launching the Backup CLI.....	347
Aster Database Backup Commands .....	347
Help with CLI Commands.....	349
Issuing Backup Commands from a File .....	349
Quitting the CLI .....	349
Physical Backup and Restore.....	350
Introduction .....	350

Physical Backup.....	351
Physical Restore.....	358
Logical Backup and Restore .....	361
Introduction to Logical Backup.....	361
Table-Level Backup.....	363
Table-Level Restore.....	365
Monitoring and Managing Backups through the AMC.....	366
Troubleshooting and Limitations.....	366
Disaster Recovery Backups .....	367
<b>Aster Glossary.....</b>	<b>368</b>
<b>Error Codes.....</b>	<b>380</b>
Parent/Child Tables and Automatic Logical Partitioning Compared.....	386
Benefits of Automatic Logical Partitioning over Parent/Child Tables.....	386
Automatic Logical Partitioning vs. parent-child tables with inheritance .....	388
Logical Partitioning Through Inheritance (Parent/Child).....	389
Table Inheritance Overview.....	389
Benefits of Logical Partitioning through Inheritance .....	389
How to Logically Partition a Table with Inheritance.....	390
Inheritance: What do the child tables inherit? .....	391
Cascading: What commands cascade to child tables? .....	391
A Simple Child Table Partitioning Example.....	391
A Detailed Child Table Partitioning Example .....	393
List of Third-Party Tools in Aster Database .....	396
The Source for these packages are available upon request: .....	399
Text of Third-Party Software Licenses.....	400
<b>Index .....</b>	<b>708</b>

# Preface

This guide explains the software tasks you will perform to use your Aster Database 5.10.00.01 cluster. If you're using a later version, you must download a newer edition of this guide!

The following additional resources are available:

- Aster Database upgrades, clients and other packages:  
<http://downloads.teradata.com/download/aster>
- Documentation for existing customers with a Teradata @ Your Service login:  
<http://tays.teradata.com/>
- Documentation that is available to the public:  
<http://www.info.teradata.com/>

Documentation available from Teradata Aster includes:

- *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide* documents how to administer Aster Database on the appliance, including using the Aster Management Console (AMC).
- *Teradata Aster Big Analytics Appliance 3H Database User Guide* documents how to use Aster Database on the appliance, including best practices, loading and exporting data, and connecting to Teradata and Hadoop.
- *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference* is a reference for SQL and function usage in Aster.
- *Aster Client Guide* documents how to install and use Aster Clients.
- *Aster Analytics Foundation Guide* documents all available SQL-MR analytic functions, including function name, description, syntax, arguments and examples.
- *Teradata Aster Big Analytics Appliance 3H Development Environment User Guide* documents the Teradata Aster plug-in for Eclipse, which enables developers to create their own SQL-MR Java-based functions within a visual development environment.
- *Teradata Aster Big Analytics Appliance 3H Upgrade Guide* explains how to upgrade a Aster Database cluster or Aster Database Backup cluster.

# Conventions Used in This Guide

This document assumes that the reader is comfortable working in Windows and Linux/UNIX environments. Many sections assume you are familiar with SQL.

## Typefaces

This document uses the following typographical conventions. Command line input and output, commands, program code, filenames, directory names, and system variables are shown in a monospaced font. Words in *italics* indicate an example or placeholder value that you must replace with a real value. **Bold type** is intended to draw your attention to important or changed items. Menu navigation and user interface elements are shown using the User Interface Command font.

## SQL Text Conventions

In the SQL synopsis sections, we follow these conventions

- Square brackets ([ and ]) indicate one or more optional items.
- Curly braces ({ and }) indicate that you must choose an item from the list inside the braces. Choices are separated by vertical lines (|).
- An ellipsis (...) means the preceding element can be repeated.
- A comma and an ellipsis (, ...) means the preceding element can be repeated in a comma-separated list.
- In command line instructions, SQL commands and shell commands are typically written with no preceding prompt, but where needed the default Aster Database SQL prompt is shown: beehive=>

## Command Shell Text Conventions

For shell commands, the prompt is usually shown. The \$ sign introduces a command that's being run by a non-root user:

```
$ ls
```

The # sign introduces a command that's being run as root:

```
# ls
```

# Contact Teradata Global Technical Support (GTS)

For assistance and updated documentation, contact Teradata Global Technical Support (GTS):

- **Support Portal:** <http://tays.teradata.com/>
- **International:** 212-444-0443

- **US Customers:** 877-698-3282
- **Toll Free Number:** 877-MyT-Data

## About Teradata Aster

Teradata Aster provides data management and advanced analytics for diverse and big data, enabling the powerful combination of cost-effective storage and ultra-fast analysis of relational and non-relational data. Teradata Aster is a division of Teradata and is headquartered in San Carlos, California.

## About This Document

This is the “Teradata Aster Big Analytics Appliance Database User Guide,” version 5.10.00.01, edition 2. This edition covers Aster Database version AD5.10.00.01 and was published as a revision to AD5.10 on September 30, 2013.

Get the latest edition of this guide! This document is updated very frequently. You can find the latest edition at <http://tays.teradata.com/>

## Revision History

Date	Description
September 30, 2013	Updated for release 5.10.00.01. This update also removed the documentation for Aster Clients (ODBC, JDBC, .NET, loader and export) out of the Aster Database documents and into the <i>Aster Client Guide</i> to better mirror the separation of Aster Client releases from Aster Database releases. ACT documentation is retained to support running ACT on the queen.
May 28, 2013	Procedure changes for upgrade of secondary queen.
May 1, 2013	Initial Release 5.10

# Section I Aster Database User Guide

This volume explains how to connect to and use Aster Database to perform data analysis. For details, see:

- [Tables](#)
- [Aster Database Cluster Terminal \(ACT\)](#)
- [SQL-MapReduce and Stream](#)



# CHAPTER 1 Tables

This section contains:

- [Creating Tables](#): The basics.
- [Table Persistence](#): Use regular, temporary and analytic tables for different levels of persistence.
- Your options for distributing and replicating tables for fast performance:
  - [Fact Tables and Dimension Tables](#)
  - [Distribution and Logical Partitioning Compared](#)
- [Table Distribution](#): Have Aster Database automatically distribute a large table across the cluster.
- [Automatic Logical Partitioning](#): Split up a large table into partitions, according to bucketing criteria that you specify.
- [Columnar Tables](#): Data is stored by column, rather than by row. This means that all of the values for a single column are stored next to each other in the same file. In a columnar table, all of the data for a single row is NOT stored in a single contiguous portion of a file.

## Creating Tables

You create tables in Aster Database in much the same way as you would in any SQL-compliant database, but there are a few important differences:

- Each table is either a fact table or a dimension table. This optimizes the performance of queries. See “[Fact Tables and Dimension Tables](#)” on page 21.
- Most tables are distributed (a.k.a., “physically distributed”) in Aster Database, meaning the rows of each distributed table are spread throughout the cluster. To control how your data is distributed, you use the `DISTRIBUTE BY HASH` clause to specify a distribution key column. Using `DISTRIBUTE BY HASH` with a distribution key is mandatory for fact tables and optional for dimension tables. Joins involving columns that are part of a distribution key constraint will be optimized. See “[Distribution and Logical Partitioning Compared](#)” on page 29.
- You can optionally logically partition the data in a large table into child partitions for easier management and faster query response. To control how your data is partitioned,

you use the PARTITION BY RANGE or PARTITION BY LIST clause to define the child partitions and set the criteria that determine which partition each row belongs to. See “[Automatic Logical Partitioning](#)” on page 35.

You create a table by specifying the table name, along with all column names and their datatypes. Use CREATE FACT TABLE or CREATE DIMENSION TABLE to specify the type of table you want to create. (If you type “CREATE TABLE”, Aster Database creates a fact table.)

```
CREATE FACT TABLE weather (
    cityid          int,
    temp_lo         int,           -- low temperature
    temp_hi         int,           -- high temperature
    prcp            real,          -- precipitation
    obs_date        date)
DISTRIBUTE BY HASH(cityid);

CREATE DIMENSION TABLE cities (
    cityid          int,
    name             varchar(80),
    country          varchar(80))
DISTRIBUTE BY REPLICATION;
```

In the definition of the table `weather`, above, the DISTRIBUTE BY HASH constraint specifies that the table’s contents will be distributed on the `cityid` column. The dimension table `cities` is distributed by replication (replicated on each vworker). See “[Table Distribution](#)” on page 31 for more information about methods of distribution and distribution keys.

## Fact Tables and Dimension Tables

A dimensional model is a type of data model used in data warehousing that consists of fact and dimension tables. The schemas developed following this model are usually referred to as star schemas. A star schema typically consists of two types of tables: there is a single, large fact table that contains the bulk of the raw data, and there is a set of dimension tables that contain the domains of values that describe the various items, actions, and players involved in the activity that the fact table tracks.

To provide optimal performance for large-scale, data analytics workloads, Aster Database natively differentiates between fact tables and dimension tables. Declaring a table as a fact table or a dimension table in CREATE TABLE and CREATE TABLE AS statements affects how Aster Database stores the table contents internally. In each case, the manner in which the data is stored is optimized for the types of operations that the table will be involved in (particularly for JOINs between tables).

### Fact Tables

Fact tables are usually very large (i.e. millions or billions of rows), with each row containing a set of dimension values and a set of measures. These tables contain two types of columns: the columns that contain the facts (the raw data you’re tracking such as units sold or pages clicked) and the columns that are foreign keys to the dimension tables (Note that Aster Database does not enforce referential constraints. Foreign keys are used mainly for joining

tables.) In Aster Database you must declare a distribution key column for each fact table using `DISTRIBUTE BY HASH`. The distribution key tells Aster Database how to divide up the table's contents so that these contents can be physically distributed across the vworkers. Distributing the data in this way is called distribution or physical distribution in Aster Database. See [Distribution and Logical Partitioning Compared \(page 29\)](#).

When creating a table, if `DISTRIBUTE BY HASH` is used, then the table will be a FACT table by default, but may optionally be specified as a DIMENSION table.

## Dimension Tables

Dimension tables are usually smaller than fact tables (that is, a typical dimension table holds only thousands of rows, rather than millions). Each dimension table specifies a set of known values for a particular dimension. For example, a `customers` table is a dimension table that contains detailed information about each customer (for example, `customer_id`, name, address, and phone\_number).

Most dimension tables are replicated in Aster Database, meaning that a copy of the table exists on every node in the cluster. Having a local copy on every node makes it more likely that joins can run locally on each node, providing faster query results. To declare your table as a replicated dimension table, include the `DISTRIBUTE BY REPLICATION` clause in the `CREATE TABLE` statement.



**Tip!** A replicated table in Aster Database is, by definition, a dimension table. As a consequence, when you include the `DISTRIBUTE BY REPLICATION` clause, the `DIMENSION` keyword is optional. You should specify the method of distribution whenever you create a table as a best practice. However, if your `CREATE TABLE` statement includes the `DIMENSION` keyword without the `DISTRIBUTE BY` clause, the default distribution method will be `DISTRIBUTE BY REPLICATION`.

Optionally, you can declare your dimension table as distributed, by declaring a distribution key column using `DISTRIBUTE BY HASH`. In that case the table will be distributed across nodes using the distribution key specified, rather than replicated on every node. There may be a couple advantages to having a distributed dimension table:

- If the fact table is distributed on the column that will be used to perform joins to a dimension table, it can be very practical to distribute that dimension table on the same column, too. Joins between the fact and dimension tables on their respective distribution key fields will be fast because the lookups will be local.
- Additions and updates to a distributed dimension table will be faster because those changes only need to be made in one place, rather than to every instance, as is the case with a replicated table.

## Example Fact and Dimension Tables

To show how we use fact and dimension tables, let's create three tables:

- a fact table `sales` that lists the total value of each sale,
- a dimension table `customers` which contains customer details and is distributed on `customer_id`, and

## Tables

### Fact Tables and Dimension Tables

- another dimension table `states` that contains information about the states.

We create the tables like this:

```
CREATE FACT TABLE sales (
    sale_id int,
    customer_id int,
    total int,
    PRIMARY KEY (sale_id, customer_id))
DISTRIBUTE BY HASH(customer_id);

CREATE DIMENSION TABLE customers (
    customer_id int PRIMARY KEY,
    name text,
    stateid int)
DISTRIBUTE BY HASH(customer_id);

CREATE DIMENSION TABLE states (
    state_id int,
    name text)
DISTRIBUTE BY REPLICATION;
```

The tables `sales` and `customers` are distributed on the column provided in their respective `DISTRIBUTE BY HASH` constraints. Since the table `states` specifies `DISTRIBUTE BY REPLICATION`, this table is replicated across all worker nodes.

By default, new tables are declared as fact tables if `DISTRIBUTE BY HASH` is used, so the `sales` table could be equivalently created by omitting the parameter `FACT`:

```
CREATE FACT TABLE sales (
    sale_id int,
    customer_id int,
    total int,
    PRIMARY KEY (sale_id, customer_id))
DISTRIBUTE BY HASH(customer_id);
```

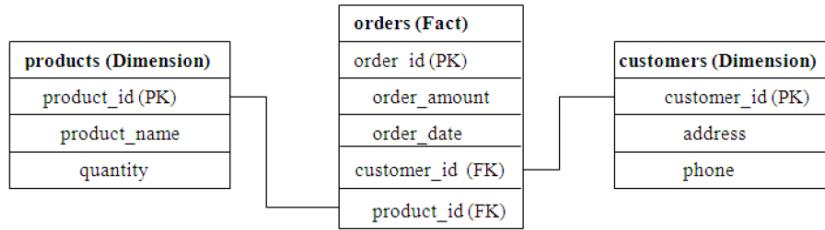
## Another Example: The `orders` Fact Table

Aster Database provides native support for fact and dimension tables, thereby allowing you to optimize the placement of data to support efficient running of queries, for example, JOINs between multiple large tables.

Analytics typically involve calculations and analysis of data, such as COUNT, COUNT DISTINCT, SUM, and AVERAGE in a fact table. For example, an `orders` table could be a fact table containing dimension values, such as `customer_id` and `order_date`, and measures, such as `order_amount`. You could easily calculate the number of customers who placed orders on a specific date using only the `orders` table.

The following figure shows the fact table (`orders`) and the related dimension tables (`products` and `customers`) in a database:

Figure 1: Example: orders Fact Table with products and customers Dimension Tables



The example shown here consists of:

- a fact table, `orders`, that holds a row for every customer order. The `orders` table is distributed by `customer_id`;
- a dimension table, `customers`, that stores a row for each customer. This table is distributed by `customer_id`, just like the fact table `orders`. Joins between `orders` and `customers` on their respective `customer_id` fields will be fast because the lookups will be local.
- a dimension table, `products`, that contains a row for every product the company sells. This table is replicated on every Aster Database node to allow fast lookups.

To declare the tables, you use `CREATE TABLE` or `CREATE TABLE AS` statements. You can specify whether the table is a fact or dimension table and the columns that make up the table. The following statements are used to create the example:

***Example: Creating the “orders” fact table:***

```
CREATE FACT TABLE orders (
  order_id      int PRIMARY KEY,
  customer_id   int NOT NULL,
  amount         int,
  order_timestamp timestamp)
DISTRIBUTE BY HASH(customer_id);
```

***Example: Creating the “customers” dimension table:***

```
CREATE DIMENSION TABLE customers (
  customer_id   int PRIMARY KEY,
  name           text,
  address        text,
  state          char(2))
DISTRIBUTE BY HASH (customer_id);
```

### Efficient Queries Against Our “orders” Example

The model in this example supports efficient querying because Aster Database places all the rows for orders by a specific customer on the same node. This implies that all the following clauses are optimized for local and parallel execution on each node:

- a `JOIN` between `orders` and the `customers` table, which is also partitioned by `customer_id`, or a `JOIN` between `orders` and another fact table that is also distributed on `customer_id`,
- a `GROUP BY` on the `orders` table on the `customer_id` column,
- a `COUNT DISTINCT customer_id` on the `orders` table.

# Table Persistence

There are three different types of tables you can work with, each with its own level of persistence:

- [Regular or Persistent Tables](#)
- [Temporary Tables](#)
- [Analytic Tables](#)

## Regular or Persistent Tables

When you use the syntax CREATE TABLE without specifying TEMP, TEMPORARY or ANALYTIC, you are creating a regular, or “persistent” table. The table continues to exist along with its data, until it is dropped. The table will persist across sessions and transactions. The table survives system restarts, adding or removing nodes, partition splitting, backup and restore, export and load, etc.

## Temporary Tables

Temporary tables are persistent only within one transaction. Once the transaction ends, the temporary table will no longer be available. Temporary tables are used when data must be stored for use within the scope of a transaction, but will not be needed once the transaction has completed.

You can declare temporary tables within a transaction. These temporary tables will not exist after that transaction ends.

### Example

Declare a temporary fact table:

```
BEGIN;
...
CREATE TEMP FACT TABLE ...;
...
END;
```

or using CTAS:

```
BEGIN;
...
CREATE TEMP FACT TABLE AS ...;
...
END;
```

## Analytic Tables

Version 5.10 of Aster Database introduced analytic tables, which have a persistence between that of regular and temporary tables. This special type of table was created to hold data that is useful for operations across the span of several transactions, sessions or days. If the analysis of

the data will last longer than a few days, be prepared to re-generate the table if necessary, or else use a permanent table. The data in an analytic table is not as persistent as data held within a regular table. For example, it is not replicated and will not survive a system restart. Analytic tables should only be used for derived data, and never for the original source data.

Making good use of analytic tables where appropriate can speed up query performance and make multiple explorations on a specific set of data easier to perform. Analytic tables are not replicated, so for very large tables that are based on derived data, they reduce the load on the cluster. They have the benefit over temporary tables of not having to worry about losing the data if a session or transaction is terminated before the user has finished doing the analysis.

Some common use cases for analytic tables are:

- Create an analytic table to hold the output of a SQL-MR function, such as sessionize, attribution or nPath. Then use the analytic table as input to other SQL-MR functions or SQL queries. For example, nPath is sometimes used to filter web sessions based on the behavior of shoppers in an online store (i.e. browsers, cherry pickers, price-sensitive shoppers, etc.). Then further analysis can be done on just the sessions that fit that behavior profile.
- Use an analytic table to hold the results of a resource-intensive JOIN operation, so further exploration can be done on the data without having to perform the JOIN again.
- Employ analytic tables for a complex multistep process for which you need the highest performance and want to keep the end results, but not the intermediate steps. In this case, you can do most of the processing using analytic tables, and then write to a regular (persistent) table at the very end of that process.

## Operations that Invalidate Analytic Tables

The following database operations will make all analytic tables invalid. This means that the tables continue to exist, but their data is no longer accessible:

- Soft restart
- Hard restart
- Node failover
- Activation of a different or new vworker
- Balance data
- Balance process
- Addition of a node
- Partition splitting
- Queen replacement
- Activate

In fact, any operation that changes the state and configuration of the Aster Database cluster will likely invalidate any analytic tables.

The reason these operations invalidate analytic tables has to do with replication. Analytic tables are effectively unreplicated (RF=1) tables, because although their metadata is replicated, the data itself is not. After a worker failover operation, the partition that was previously a

Secondary becomes a Primary. Since the data in the Analytic Tables was never replicated to the Secondary, that Secondary (which is now the new Primary) does not have a copy of the data rows - just an empty table. Therefore, a worker failover must invalidate the Analytic Tables to force the user to recognize that the Analytic Tables don't have any data. Similarly, other operations which may cause the Secondary to be used (i.e. balance data, node failover, etc. will also have the side effect of invalidating the analytic tables.

If an analytic table becomes invalid, you may perform one of the following operations on it in order to make it valid again, depending upon how it was originally created:

- If the table was created using CTAS:
  - a DROP the table.
  - b Re-issue the CTAS query that created the table originally.
- If the table was not a CTAS table:
  - a TRUNCATE the table.
  - b Re-issue the query you used to populate the table originally, to re-load the data.

In this situation, the TRUNCATE cleans up the analytic table and resets its status to 'valid'.

Note that after this operation is complete, the table will contain an updated version of the data, which may not be the same as the version of the data in the original table before it was invalidated. The data will essentially be a refreshed snapshot, taken from the source data at the time the query or function that populates the analytic table is issued.

### ***Example***

This example shows creating an analytic table and repopulating it after it becomes invalid.

- 1 Create an analytic table using CTAS with a three way JOIN:

```
CREATE ANALYTIC TABLE top_customers_activity (customer_id int,
                                             lastname varchar, firstname varchar, buyer_level int, sales_date
                                             timestamp)
DISTRIBUTE BY HASH (customer_id)
AS (SELECT cust.customer_id, cust.last_name, cust.first_name,
       cust.frequent_buyer_level, d.sales_date
    FROM (sales_fact sales JOIN customer_dim cust ON
          sales.customer_id=cust.customer_id) JOIN date_dim d
    ON (sales.date_id=d.date_id)
    WHERE cust.frequent_buyer_level > 3);
```

- 2 Now assume that a soft restart occurs, which invalidates the table. After the soft restart, if you attempt to SELECT from the table, you will see an error like:

```
ERROR: analytic table "public"."top_customers_activity" is invalid
```

- 3 Truncate the table:

```
TRUNCATE TABLE top_customers_activity;
```

- 4 Repopulate the data in the table:

```
INSERT INTO top_customers_activity (
    SELECT cust.customer_id, cust.last_name, cust.first_name,
           cust.frequent_buyer_level, d.sales_date
    FROM (sales_fact sales JOIN customer_dim cust ON
          sales.customer_id=cust.customer_id) JOIN date_dim d
```

```
ON (sales.date_id=d.date_id)
WHERE cust.frequent_buyer_level > 3);
```

- 5 The table now contains a new snapshot of the data from the source tables, and is available for use.

## **Release Space Held by Invalid Analytic Tables**

When analytic tables become invalid, they still occupy space on disk. To free up storage being used by invalid analytic tables:

- 1 Log in to Aster Database as db\_superuser.

- 2 Obtain a list of all invalid tables:

```
SELECT * FROM nc_all_tables WHERE NOT valid;
```

- 3 Issue a DROP or a TRUNCATE on each of the tables.

## **Notes on Using Analytic Tables**

- 1 As with any type of table, users must have the correct permissions granted on the table in order to access it.
- 2 Physical backup and restore are not supported for analytic tables, but logical backup and restore are supported.
- 3 Analytic tables may be created as logically partitioned tables, but not as parent child tables with inheritance.
- 4 If an analytic table is logically partitioned, the whole hierarchy will be analytic.
- 5 If you attempt to attach a non-analytic table as a partition to an analytic logically partitioned table, the operation will fail with an error.
- 6 A detached logically partitioned analytic table remains an analytic table, with all its constraints. If you only TRUNCATE a detached logically partitioned analytic table (and do not repopulate its data), some inconsistencies in the table hierarchy will occur.
- 7 Truncate only will create some curious inconsistencies in the table hierarchy .
- 8 Analytic tables may be fact or dimension tables.
- 9 Analytic tables do not support columnar storage.
- 10 GiST indexes are not supported for analytic tables. This only affects IP4range datatypes, which use GiST indexes. If you need to index a column with a IP4range datatype, use a regular table, not an analytic table.
- 11 Analytic tables are a nonstandard extension to ANSI SQL.
- 12 You can see the persistence of an analytic table, and whether it is valid or invalid by issuing \d <tablename> in ACT:

```
retail_sales=> \d top_customers_activity
Table "public"."top_customers_activity"
Column          |          Type          | Modifiers
-----+-----+-----+
customer_id    | integer
lastname        | character varying
firstname       | character varying
buyer_level     | integer
```

```

sales_date | timestamp without time zone |

Table Type:
fact

Distribution Key:
customer_id

Compression Level:
none

Storage Type:
row

Persistence:
analytic

Valid:
true

```

## Distribution and Logical Partitioning Compared

There are two, unrelated ways in which a large table can be subdivided for better management in Aster Database. Physical distribution (using the keywords DISTRIBUTE BY HASH) divides the data in a table between virtual workers (referred to as “vworkers”) to improve performance and balance data across the cluster. Partitioning is a logical distribution of data into a parent table with child partitions for ease of administration and improved performance. Physical distribution and logical partitioning are independent, and can be used separately or combined. For a typical, large fact table, both of these methods should be used.

- Distribution: Distribution physically divides the data in a table between vworkers. Typically, each worker node (machine) in Aster Database hosts a number of vworkers. When creating a distributed table using DISTRIBUTE BY HASH, you are telling Aster Database to split the table’s data across many vworkers. This is a key Aster Database feature. Each vworker holds a subset of the table’s rows to enable data balancing, scaling and fast performance.

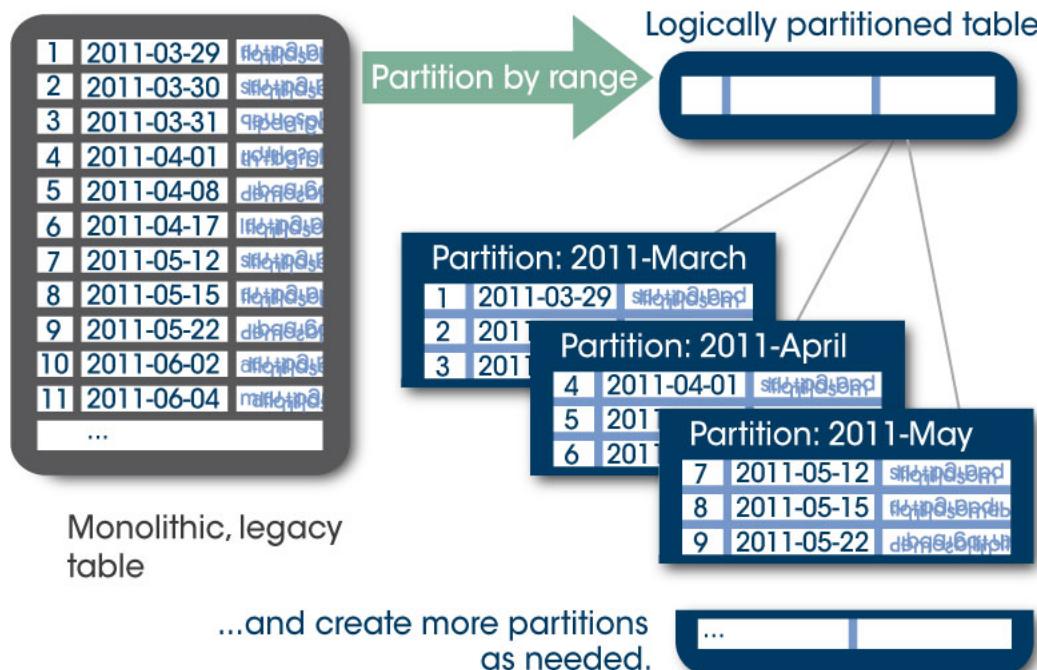
When you insert data into a hash-distributed table, Aster Database performs the distribution automatically based on the distribution key you declared when you created the table. The distribution is also automatic in the sense that you don’t declare the boundaries of each set of rows; you just specify which column will be used to define the distribution, and Aster Database chooses the boundaries to split up the rows and balance the data across vworkers. When you query the table, Aster Database automatically finds the desired records; you don’t need to know which vworker stores them. See also “[Table Distribution](#)” on page 31.

When creating a table, you may either physically distribute rows in a table among vworkers with the DISTRIBUTE BY HASH argument or replicate the entire table on all vworkers using DISTRIBUTE BY REPLICATION. The type of distribution you choose should be determined by the type of table (FACT or DIMENSION) and the ways in which it will be joined to other tables.

- `DISTRIBUTE BY HASH (column_name)` declares that the table will be physically distributed using a hash function. If the table is a FACT table, you must `DISTRIBUTE BY HASH`.
- For a HASH-distributed table, you declare one column to be the distribution key. For each row, a hash of this column's value determines where the row is stored in the cluster, as described in “[Fact Tables and Dimension Tables](#)” on page 21.
- `DISTRIBUTE BY REPLICATION` declares that the entire table will be replicated on all vworkers. For a small table, this can speed up query execution times, particularly those involving joins.
- Logical Partitioning: Logical partitioning is the practice of splitting one large table into smaller segments (child partitions) for ease of management and faster performance. When creating a table with logical partitioning, you define the partitions by range or by a list of values in a logical way (e.g. by date/timestamp, region, store, etc.) using the `PARTITION BY` keyword. The child partitions maintain a relationship to the parent table, and inherit their structure and constraints from the parent table. See also “[Automatic Logical Partitioning](#)” on page 35. The following diagram illustrates a table using logical partitioning:

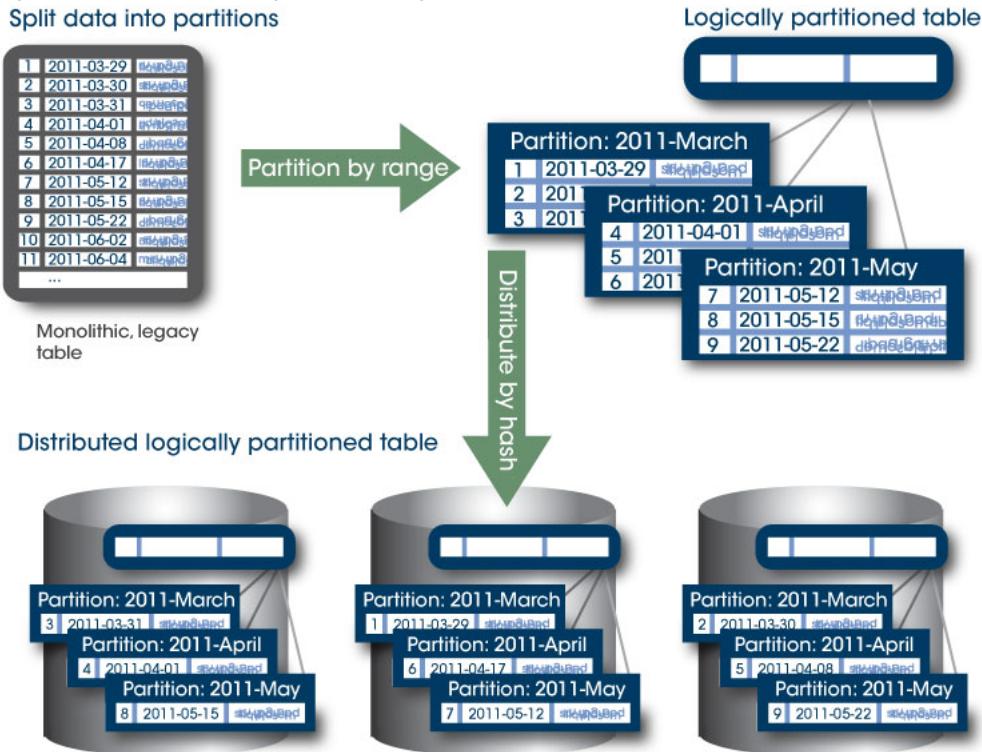
Figure 2: A table using logical partitioning

### Split data into partitions



In summary, most logically partitioned tables are also distributed tables. (You may choose to make them either distributed or replicated tables. See “[Rules for DISTRIBUTE BY on a logically partitioned table](#)” on page 33.) The following diagram illustrates the difference between distribution and logical partitioning when they occur within the same table:

Figure 3: Distribution and logical partitioning compared



## Table Distribution

As mentioned earlier, distribution is the practice of physically splitting a table's data across vworkers in Aster Database. You control distribution by declaring a `DISTRIBUTE BY HASH` constraint on a table. (See “[Example Fact and Dimension Tables](#)” on page 22 for a syntax example.) Data in a distributed table is distributed across the vworkers such that all the rows that share a given distribution key value are guaranteed to be on the same vworker.

Your distribution options depend on what type of table you’re distributing:

- A fact table is always a distributed table. Each fact table must use `DISTRIBUTE BY HASH` with a distribution key so that Aster Database can split its rows among vworkers.
- A dimension table can be a distributed table or a replicated table:
  - **distributed** – you define a distribution key on it using `DISTRIBUTE BY HASH(column_name)`, so Aster Database can split its rows among vworkers; or
  - **replicated** – you define it without a distribution key by using `DISTRIBUTE BY REPLICATION`, which tells Aster Database to keep a full copy of the dimension table on each vworker. This can speed up joins against the table.



**Tip!** Don't confuse distribution with logical partitioning! See “[Distribution and Logical Partitioning Compared](#)” on page 29.

## Distribution Key

The distribution key constraint (defined using the `DISTRIBUTE BY HASH (column_name)` argument in the `CREATE TABLE` statement) specifies the column of a table that determines distribution of the table across the cluster.

### Rules for distribution keys

- 1 Single-column: The distribution key consists of a single column only.
- 2 Not nullable: No null values are allowed in the distribution key column.
- 3 Datatype support: Only columns of the following types may be used as distribution keys:

```
int
smallint
bigint
numeric
numeric(p)
numeric(p,a)
text
varchar
varchar(n)
UUID
bytea
```

- 4 Primary key: If the table has a primary key, the distribution key must be one of the columns from the primary key.
- 5 No modifications allowed: Once you have declared your distribution key:
  - you cannot specify a new distribution key for the table.
  - you cannot modify the distribution key column in any way. That is, you cannot drop or rename the distribution key column, nor can you change its type.
  - you cannot update a value in the distribution key column.If you wish to change your distribution key column, use `CREATE TABLE ... AS SELECT ...` to create a new table with the desired distribution key.
- 6 DEFAULT rules not recommended: In Aster Database you can place a DEFAULT rule on a column to provide a default value that will be used if a row is inserted but the INSERT fails to provide a value for that column. Defining a DEFAULT rule on a distribution key column is not recommended because it increases the risk of data skew in the cluster.

### Rules for the distribution key on a fact table

- 1 Every fact table must have a distribution key. This includes logically partitioned tables. If there is no distribution key, the `CREATE TABLE` or `CREATE FACT TABLE` statement will fail.  
You declare the distribution key using the `DISTRIBUTE BY HASH (column_name)` keyword in your `CREATE TABLE` statement.
- 2 If the table has a primary key defined, then the distribution key must be one of the columns from the primary key. The `CREATE TABLE` or `CREATE FACT TABLE` statement will fail, unless the primary key is used as the distribution key.

If you have specified a single-column primary key of a distribution-key-suitable datatype, it will automatically be used as the distribution key. If you create a fact table with multiple columns in the primary key, then you must declare a distribution key explicitly.

- 3 If DISTRIBUTE BY HASH is used, then the table will be a FACT table by default, but may optionally be specified as a DIMENSION table.
- 4 The column you choose as your distribution key must have one of the datatypes listed in “[Distribution Key](#)” on page 32.

### **Rules for the distribution key on a dimension table**

Dimension tables optionally can have distribution keys. To define a distribution key on a dimension table, you declare the distribution key using the `DISTRIBUTE BY HASH (column_name)` keyword in your `CREATE TABLE` statement.

If you do not declare a distribution key on a dimension table, that table will be a replicated dimension table, which means that Aster Database will maintain an up-to-date copy of the table on every vworker in the cluster. If DISTRIBUTE BY REPLICATION is used, and FACT/DIMENSION are omitted, then the table will be created as a DIMENSION table (the DIMENSION keyword is optional).

### **Rules for DISTRIBUTE BY on a logically partitioned table**

- 1 Every statement that creates a logically partitioned table must use the DISTRIBUTE BY clause. Logically partitioned tables do not support a PRIMARY KEY. For this reason, you must include a DISTRIBUTE BY clause when creating a logically partitioned table.
- 2 A logically partitioned table can be distributed (spread across the cluster) or replicated (every worker has a copy).
  - To distribute a logically partitioned table, include `DISTRIBUTE BY HASH (column_name)` in the `CREATE TABLE` statement.
  - To replicate a logically partitioned table (all child partitions are copied to all nodes), specify `DISTRIBUTE BY REPLICATION`.
- 3 If you forget to include a DISTRIBUTE BY clause when you define a logically partitioned table, the `CREATE TABLE` will fail with the error message:  
`ERROR: use of PARTITION BY clause requires the use of the DISTRIBUTE BY clause at or near ";"`
- 4 When using DISTRIBUTE BY HASH, the column you choose as your distribution key must have one of the datatypes listed in “[Distribution Key](#)” on page 32.
- 5 You should take dynamic child partition pruning into consideration when choosing a distribution key. When performing an INNER JOIN that is either a hash join or a merge join using the distribution key, dynamic pruning can reduce the number of partitions scanned in the right hand table. This occurs automatically in cases where the WHERE clause contains a constant(s) that can be used to prune the table being queried. Beginning in Aster Database version 5.0, it also can occur indirectly, using information from the left hand side of the JOIN to prune tables from the right hand side of the join before the right hand table is scanned and the JOIN is performed. This can increase performance significantly.

# Logical Partitioning

Logical partitioning is the practice of splitting off what is logically one large table into smaller child partitions for faster performance. Logical partitioning also makes it easier to manage data in the table. This is a common database practice as well as a popular feature of Aster Database. For example, a typical logically partitioned table might partition the set of rows based on date, with a child partition for each week's data. Logical partitioning is also known as “autopartitioning,” “list partitioning,” and “range partitioning.” You may logically partition both fact and dimension tables.

There are two methods of creating a logically partitioned table in Aster Database, each with its own syntax and behavior:

- [Automatic Logical Partitioning](#) is the preferred method.
- [Logical Partitioning Through Inheritance \(Parent/Child\)](#) is retained for backwards compatibility.



**Tip!** Don't confuse logical partitioning with distribution!

- See “[Distribution and Logical Partitioning Compared](#)” on page 29.
- A logically partitioned table can be distributed or replicated. See “[Rules for DISTRIBUTE BY on a logically partitioned table](#)” on page 33

## Logical Partitioning Overview

A table can be partitioned or split into child partitions to optimize for queries that require only well-defined subsets of the table's data. For example, you can split the `orders` table by the order date, and divide it into separate child partitions, one per month. This is useful in an environment where each monthly report query requires only one month's worth of records from the `orders` table, and where analysts are not interested in scanning the entire `orders` table.

## Benefits of Logical Partitioning

Logical Partitioning provides several benefits:

- 1 Query performance is improved dramatically for certain types of queries.
- 2 Update performance is improved, since each partition of the table has indexes smaller than an index on the entire data set would be.
- 3 Bulk deletes may be accomplished by simply removing a child partition, if that requirement is planned into the partitioning design. `ALTER TABLE . . . DETACH` followed by `DROP TABLE` are far faster than a bulk `DELETE`.
- 4 Removing a large segment of data does not leave a big hole in the table as it would when using only one large table.
- 5 Archiving of data can be automated through scripting archiving activities to occur in batches corresponding to the logical partitioning plan.

## Automatic Logical Partitioning

Aster Database version 4.6 introduced a new logical partitioning mechanism, referred to as “automatic logical partitioning”. Under this approach, you create a logically partitioned table using the new PARTITION BY RANGE or PARTITION BY LIST clause. This syntax replaces the old approach that used the INHERITS keyword. Aster Database continues to support the old approach for backward compatibility. For more information on support for child tables using inheritance, see [“Parent/Child Tables” on page 386](#).

### Automatic Logical Partitioning vs. non-partitioned tables

- You may have a PRIMARY KEY or UNIQUE constraints on logically partitioned tables, but only if you follow the guidelines in [PRIMARY KEY or UNIQUE with automatic logical partitioning \(page 39\)](#).
- A logically partitioned table may not have an inheritance child or an inheritance parent.
- A logically partitioned table may not be a temporary table.

### Commands to support automatic logical partitioning

The CREATE TABLE command has been extended with the addition of the PARTITION BY keyword to support automatic logical partitioning.

There are also some ALTER TABLE commands to support automatic logical partitioning. These ALTER TABLE commands are as follows:

- [ALTER TABLE...ADD PARTITION](#)
- [ALTER TABLE...DROP PARTITION](#)
- [ALTER TABLE...ATTACH PARTITION](#).
- [ALTER TABLE...DETACH PARTITION](#).
- [ALTER TABLE...ALTER PARTITION...RENAME TO](#)
- [ALTER TABLE ... ALTER PARTITION ... COMPRESS](#)

### System tables related to automatic logical partitioning

The following system tables show logical partitioning on a per-child basis:

- nc\_all\_child\_partitions
- nc\_user\_child\_partitions
- nc\_user\_owned\_child\_partitions

The following system tables show logical partitions on a per-parent basis:

- nc\_all\_parent\_partitions
- nc\_user\_parent\_partitions
- nc\_user\_owned\_parent\_partitions

A logically partitioned table will have one row in the child partition tables for each child partition. It will also have a row in the parent partition tables, even if there are no children.

System tables are documented in the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.

## Internal and leaf partitions

An internal partition is one that contains a PARTITION BY clause. The base table is also considered to be an internal partition. Note that an internal partition does not need to have any child partitions: the PARTITION BY clause can give an empty list. All insertions that match an internal node with no child partitions will fail. An internal node without child partitions is useful if the user wants to just specify the partition expression, and add child partitions later.

A leaf partition is one that does not have a PARTITION BY clause, which means that it is at the lowest generation of the hierarchy. Internally, only the leaf partitions have physical storage associated with them.

## Create a Logically Partitioned Table

To create a logically partitioned table, you use the keyword PARTITION BY when defining the table. The PARTITION BY clause causes the table to be logically partitioned in separate files, and defines the layout of the partitions. Each logical partition is created as a child partition of a single parent table. You may create a multi-generational tree, for example with a parent table that is the parent of child partitions for each year, which in turn are the parents of child partitions for each month, which in turn are the parents of child partitions for each day. For the complete SQL Reference for CREATE TABLE, see the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.

### PARTITION BY clause

The PARTITION BY clause causes the table to be logically partitioned in separate files, and defines the layout of the partitions. There are a few things to keep in mind when choosing how to partition the data:

#### Partitioning by list or range of values

To define which records belong to which partition, you supply list or range partitioning check constraints using PARTITION BY LIST or PARTITION BY RANGE:

- A list partitioning constraint (PARTITION BY LIST) provides a list of key values that belong to a partition, for example country or store number. A list partition is specified by simply providing the list of values.
- A range partitioning constraint (PARTITION BY RANGE) specifies a numeric, alphabetical, or date range of key values that belong to a partition, for example a timestamp range.

### PARTITION BY LIST

A list partition is specified by using PARTITION BY LIST and simply providing the list of values. If NULL is in the list, then the partition will include NULL values as well as the other values in the list. For example, if the list partition specifies VALUES(1, 2, NULL) for column x, then the partition matches rows where x=1 OR x=2 OR x IS NULL.

## PARTITION BY RANGE

A range partition is specified by using PARTITION BY RANGE with a START value and an END value.

### ***START Value***

The START value specifies the lower bound of the partition.

- Allowed values: For the START value, you may specify a constant or the keyword, "MINVALUE". Specifying "START MINVALUE" says that there is no lower bound in the partition. MINVALUE does not correspond to a real value. Conceptually, MINVALUE is something less than all possible values (including NULL).
- INCLUSIVE/EXCLUSIVE: The START value is by default INCLUSIVE. This means that if you specify a constant A as the START VALUE, the partition will match rows where the expression is  $\geq A$ . You may force the START value to be exclusive by including the keyword, "EXCLUSIVE", in which case the partition will match rows where the expression is  $> A$ .
- Required? No, declaring a START value is not required. If you omit it, Aster Database will create a default START value for you. See "[Default START Values](#)" below.

### ***Default START Values***

You may omit the START value from a partition definition.

- If you omit the START value from the first range partition, it is equivalent to declaring START MINVALUE.
- If you omit the START value from any subsequent range partition, that partition uses the END value of the preceding partition as its start value.



**Tip!** When you omit a START value, the automatically generated START value is considered INCLUSIVE as is the default for START values. Teradata Aster recommends that you do not use the keywords EXCLUSIVE and INCLUSIVE when using defaulted START values. If you do, please note:  
For each partition whose END clause is specified as INCLUSIVE, you must declare a START value for the subsequent partition. Otherwise, the missing START value will result in an error, because Aster Database will use the preceding partition's END value (INCLUSIVE) as the START value (INCLUSIVE by default) of the subsequent partition, the two partitions will overlap, which is not allowed.

---

### ***END Value***

The END value specifies the upper bound of the partition.

- Allowed values: For the END value, you may specify a constant or the keyword, "MAXVALUE". Specifying "END MAXVALUE" says that there is no upper bound in the partition. MAXVALUE does not correspond to a real value. Conceptually, MAXVALUE is greater than all possible values (including NULL).
- INCLUSIVE/EXCLUSIVE: The END value is by default EXCLUSIVE. This means that if you specify a constant A as the END VALUE, the partition will match rows where the expression is  $< A$ . You may force the END value to be inclusive by including the keyword, "INCLUSIVE", in which case the partition will match rows where the expression is  $\leq A$ .
- Required? Yes, declaring an END value is required.

## NULLs in Range Partitions

The range partition can also specify NULLS FIRST or NULLS LAST. This says that the NULL value is ordered before (NULLS FIRST) or after (NULLS LAST) all the other values in the partition. Regardless of which option for NULL ordering is designated, any NULL values will come after MINVALUE and before MAXVALUE. The default is NULLS LAST.

MAXVALUE and MINVALUE do not correspond to real values. Conceptually, MINVALUE is something less than all possible values (including NULL) and MAXVALUE is greater than all possible values (including NULL).

If NULLS LAST is chosen (that is the default), then the ordering will be:

```
MINVALUE, <actual values, e.g. Albania, Zambia>, NULL, MAXVALUE
```

If NULLS FIRST is chosen, then the ordering will be:

```
MINVALUE, NULL, <actual values, e.g. Albania, Zambia>, MAXVALUE
```

The following example creates the partitioned table `customer_addresses` with a partition for rows with NULL values for `zipcode`:

```
CREATE FACT TABLE customer_addresses( customer_id int NOT NULL,
street_addr1 varchar, street_addr2 varchar, city varchar, state varchar,
zipcode int )
DISTRIBUTE BY HASH(customer_id)
PARTITION BY RANGE(zipcode NULLS FIRST) (
PARTITION zipcode_is_NULL( START NULL END NULL INCLUSIVE),
PARTITION zipcodes_00000_01000 (START 0 END 1000 EXCLUSIVE),
PARTITION zipcodes_01000_02000 (END 2000 EXCLUSIVE),
PARTITION zipcodes_02000_03000 (END 3000 EXCLUSIVE),
... );
```

## DISTRIBUTE BY clause with automatic logical partitioning

The DISTRIBUTE BY clause is mandatory when creating a logically partitioned table. For more information, see “[Table Distribution](#)” on page 31.

### Automatic logical partitioning: A simple example

A simple, logically partitioned table can be set up with one statement, using the following steps.

- 1 Define the column names, types and constraints of the table.
- 2 Declare a distribution policy using DISTRIBUTE BY HASH or DISTRIBUTE BY REPLICATION. This step is required.
- 3 Declare the type of partitioning to use (PARTITION BY RANGE or PARTITION BY LIST).
- 4 Give each child partition a name and define the range or list of values for it. Each child partition is created as a logical partition of its parent. (Note that the same naming conventions apply to partitions as to tables - i.e. name cannot start with a symbol or number, etc.)

Below is an example of a table `orders_simple` that is partitioned by date range:

```
CREATE FACT TABLE orders_simple( order_id int, customer_id int NOT NULL,
amount int, ts timestamp )
  DISTRIBUTE BY HASH(order_id)
  PARTITION BY RANGE(ts) (
    PARTITION oldrecords( END '2011-01-01' ), -- everything pre-2011
    PARTITION jan01_2011( END '2011-01-02' ),
    PARTITION jan02_2011( END '2011-01-03' ),
    PARTITION jan03_2011( END '2011-01-04' ),
    ... );
```

The following query provides the count of all rows in all child partitions of the `orders_simple` table:

```
SELECT COUNT(*) FROM orders_simple;
```

When querying the `orders_simple` table as in this example, the optimizer automatically queries all child partitions and compiles a count of all their rows.

### **PRIMARY KEY or UNIQUE with automatic logical partitioning**

When using a PRIMARY KEY or UNIQUE constraint with a logically partitioned table:

- 1 You can CREATE a logically partitioned table with a UNIQUE or PRIMARY KEY constraint.
- 2 You may ALTER TABLE to ADD or DROP the UNIQUE or PRIMARY KEY constraint.
- 3 When issuing one of these command on a logically partitioned table that has a UNIQUE or PRIMARY KEY constraint, the constraint will be applied automatically:
  - ALTER TABLE...ADD PARTITION
  - ALTER TABLE...ATTACH PARTITION
- 4 In all of the above cases, the UNIQUE/PRIMARY KEY constraint must contains all of the columns used to partition the table in the PARTITION BY RANGE or PARTITION BY LIST clause.

### **STORAGE clause with automatic logical partitioning**

Just as when creating a flat table, you may use the STORAGE clause to specify either COLUMN or ROW. For more on columnar storage, see “[Columnar Tables](#)” on page 54. This setting automatically cascades to all child partitions. If neither is specified, ROW is used:

```
CREATE TABLE records2(id int, country varchar, ts timestamp)
DISTRIBUTE BY HASH(id)
STORAGE COLUMN
PARTITION BY RANGE(ts) (
  PARTITION daily_0 ( START '2010-01-01'::date + interval '0 days' END
  '2010-01-01'::date + interval '1 days'
    PARTITION BY LIST(country) (
      PARTITION countrylist0 ( VALUES ('a', 'b', 'c') ),
      PARTITION countrylist1 ( VALUES ('x', 'y', 'z', NULL) ),
      PARTITION countrylist2 ( VALUES ('d', 'e') )
    )
  ),
  PARTITION daily_1 ( START '2010-01-01'::date + interval '1 days' END
  '2010-01-01'::date + interval '2 days'
    PARTITION BY RANGE(country) (
      PARTITION countrylist0 ( START 'A' END 'J' ) ,
```

```

        PARTITION countrylist1 ( START 'J' END 'Z' )
    )
),
PARTITION daily_2 ( START '2010-01-01'::date + interval '2 days' END
'2010-01-01'::date + interval '3 days'
    PARTITION BY RANGE(country) (
        PARTITION countrylist0 ( START 'A' END 'E' ) ,
        PARTITION countrylist1 ( START 'F' END 'J' ) ,
        PARTITION countrylist2 ( START 'J' END 'N' ) ,
        PARTITION countrylist3 ( START 'N' END 'Z' ) ,
    )
),
...
;

```

### **COMPRESS with automatic logical partitioning**

Compression is supported at every level in the logical partition hierarchy. That is, you may compress the table itself, its index, and/or any of its child partitions. If compression is specified for the table or one of its partitions, the compression will cascade to any partitions below it in the hierarchy, unless they have their own compression explicitly specified.

Suppose we want to divide the contents of a table into daily child partitions for 2010 and 2011. Child partitions prior to 2011 will be compressed. Child partitions in 2011 will be uncompressed, and subpartitioned by region. This table can be created as

```

CREATE FACT TABLE records(id int, country varchar, ts timestamp)
DISTRIBUTE BY HASH(id)
PARTITION BY RANGE(ts) (
    PARTITION oldrecords( END '2010-01-01' COMPRESS LOW), -- everything
pre-2011
    PARTITION jan01_2010( END '2010-01-02' COMPRESS LOW),
    PARTITION jan02_2010( END '2010-01-03' COMPRESS LOW),
    PARTITION jan03_2010( END '2010-01-04' COMPRESS LOW),
    ...
    PARTITION dec31_2010( END '2011-01-01' COMPRESS LOW),
    PARTITION jan01_2011( END '2011-01-02'
        PARTITION BY LIST(country) (
            PARTITION na ( VALUES ('usa', 'canada', 'mexico') ),
            PARTITION eu ( VALUES ('germany', 'spain') )
        )
    ),
    PARTITION jan02_2011( END '2011-01-03'
        PARTITION BY LIST(country) (
            PARTITION na ( VALUES ('usa', 'canada', 'mexico') ),
            PARTITION eu ( VALUES ('germany', 'spain') )
        )
    ),
    PARTITION jan03_2011( END '2011-01-04'
        PARTITION BY LIST(country) (
            PARTITION na ( VALUES ('usa', 'canada', 'mexico') ),
            PARTITION eu ( VALUES ('germany', 'spain') )
        )
    ),
    ...
    PARTITION dec31_2011( END '2012-01-01'
        PARTITION BY LIST(country) (

```

```

        PARTITION na ( VALUES ('usa', 'canada', 'mexico') ),
        PARTITION eu ( VALUES ('germany', 'spain') )
    )
)
;

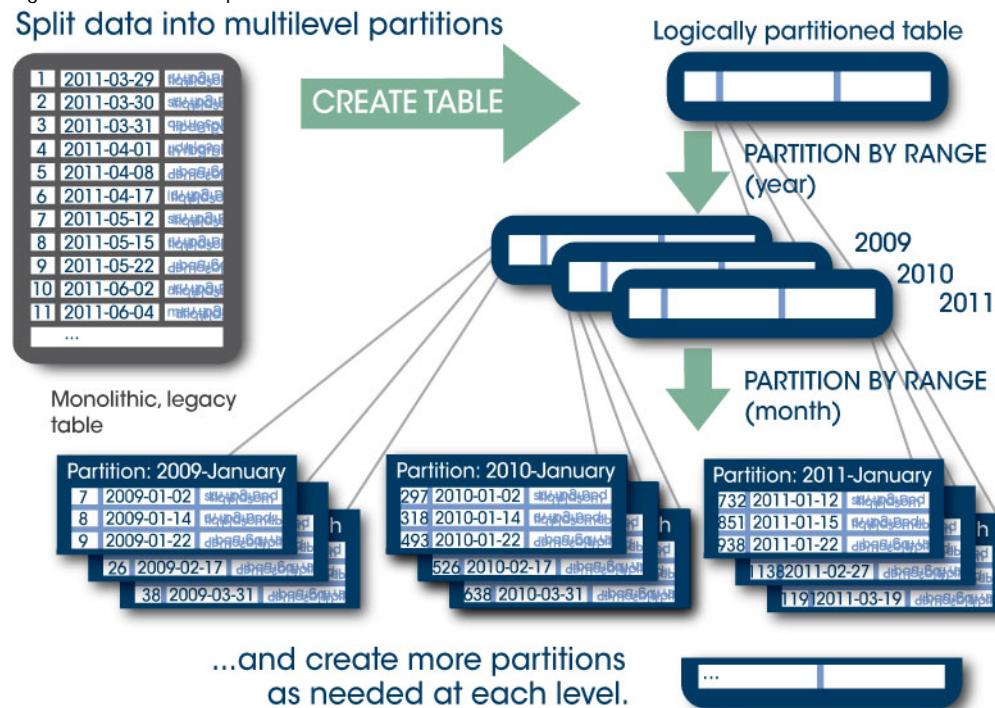
```

## Notes

Note also these tips for defining logical partitions:

- When using ranges, ensure that ranges do not overlap. When using values, ensure that each value is assigned to only one partition. If you attempt to insert rows with values that do not fall within the defined list of ranges or values for any partition, the insert will fail.
- Note that you can create multilevel partitioned tables in one CREATE TABLE command by nesting PARTITION BY statements.

Figure 4: A multilevel partitioned table



For example you could create a sales\_by\_region table partitioned by date, and then by region (for sales occurring beginning in 2010) as follows:

```

CREATE FACT TABLE sales_by_region(id int, country varchar, ts timestamp)
DISTRIBUTE BY HASH(id)
PARTITION BY RANGE(ts NULLS FIRST) (
PARTITION yearBefore2010 (
    START MINVALUE END '2010-01-01'
    COMPRESS
),
PARTITION year2010 (
    END '2011-01-01'
    PARTITION BY LIST(country) (
        PARTITION region1( VALUES('India', 'Japan') ),
        PARTITION region2( VALUES('Australia', 'New Zealand') ),
        PARTITION region3( VALUES('Denmark', 'Ireland') )
    )
);

```

```
) ) ,
PARTITION year2011 (
    END '2012-01-01'
    PARTITION BY LIST(country) (
        PARTITION region1( VALUES('India', 'Japan') ),
        PARTITION region2( VALUES('Australia', 'New Zealand') ),
        PARTITION region3( VALUES('Denmark', 'Ireland') )
    ) );
)
```



**Tip!** Take care that the constraints you define do not create overlapping logical partitions. A simple mistake would be to set up range constraints like this:

```
PARTITION jan01_2011( END '2011-01-02' INCLUSIVE ),
PARTITION jan02_2011( START '2011-01-02' END '2011-01-03' ),
```

In this example, it is not clear in which partition the ymdh value '2011-01-02' resides. To correct this, you would format the partitions as follows:

```
PARTITION jan01_2011( END '2011-01-02' ),
PARTITION jan02_2011( END '2011-01-03' ),
```

This is the same as specifying END <value> EXCLUSIVE for each partition, because END defaults to EXCLUSIVE. Specifying a START value for the second partition is not necessary, as it will default to the END value from the previous partition (which defaults to EXCLUSIVE) for its START value (which defaults to INCLUSIVE).

Aster Database will detect overlapping constraints, if present, and the CREATE TABLE will fail.

## Partition References

A “partition reference” is a construct which allows you to refer to an individual partition of a table hierarchy. The syntax is as follows:

```
partition_reference = PARTITION ( <identifier> [ . <identifier> ... ] )
```

The partition reference is used in ALTER TABLE actions. It begins with the name of the root table, followed by the name of each of the child partitions down the hierarchy to get to the partition we wish to access. Partition names are separated by “.”. From the previous example, the reference PARTITION (year2011.region1) would identify the partition that contains records from India and Japan in 2011.

The following example detaches the partition that contains those sales records from India and Japan in 2011:

```
ALTER TABLE sales_by_region DETACH PARTITION (year2011.region1) INTO
japan_india_2011_sales_temp;
```



**Tip!** You can do the same ALTER TABLE operations on a descendant partition as on a top level partition. To refer to a partition several levels down the hierarchy, you must use a partition reference. To do this, list each of its parent partitions in order, separated by “.”, and then the child partition you wish to access. For example, to refer to the partition three levels down the partition hierarchy with the name “2001\_11\_30” you would reference it as (year2001.2001\_november.2001\_11\_30).:

## ALTER TABLE...ADD PARTITION

The ALTER TABLE...ADD PARTITION operation allows adding a new partition to a logically partitioned table. The new partition will have the same columns, indexes, permissions and distribution key as the logically partitioned table to which it will be added. The syntax for ALTER TABLE...ADD PARTITION can be found in the *Teradata Aster Big Analytics Appliance*

*3H SQL and Function Reference.*

**Examples**

The following example adds a partition “south\_america” to an existing table of distributors partitioned by a list of country names:

```
ALTER TABLE distributors
    ADD PARTITION south_america (
        VALUES ('Brazil','Chile')
) ;
```

The following example adds a partition “year\_2014” to an existing table of records partitioned by timestamp range:

```
ALTER TABLE records ADD PARTITION year_2014
    (START '2014-01-01' END '2015-01-01')
;
```

When adding a partition, you can create it with nested subpartitions, all in one command.

The following example shows this. This command creates a new partition at the top level of the records table, which is partitioned by LIST. The new partition is called “p2” and it is subpartitioned by RANGE(c). The two nested partitions are called “p21” and “p22”. Partition “p22” is compressed.

```
ALTER TABLE records
ADD PARTITION p2 (VALUES('y'))
    PARTITION BY RANGE (c) (
        PARTITION p21 (END '2000-1-1'),
        PARTITION p22 (END MAXVALUE COMPRESS LOW)
) ;
```

***Some additional details for ALTER TABLE...ADD PARTITION***

- 1 Only the table owner (or the superuser) may add a partition to a table.
- 2 To add a partition to an existing table, the table must already have a partition column defined. Similarly, to add a subpartition to an existing partition, the parent of the new subpartition must already have a partition column defined. To put this another way, there are two types of partitions in a logically partitioned hierarchy: those that contain partitions and those that contain data. You cannot change a partition from one type to another.

So if you have a partition that contains data, and you want the data to be divided into subpartitions instead, you should follow this sequence:

- Detach the partition.
  - Add a new partition with subpartitions.
  - Reinsert the data into the table. It will be automatically inserted into the correct subpartitions.
  - Reattach the partition.
- 3 If the table uses list partitions, the ALTER TABLE...ADD PARTITION statement must also use a list partition. The same rule applies for range partitions. You do not specify the partition expression, since it must be the same as the existing child partitions.

- 4 When adding partitions, they need to satisfy the same consistency criteria described for CREATE TABLE. This includes being mutually distinct from all previously existing child partitions.
- 5 When using ALTER TABLE...ADD PARTITION, compression can be specified for the new partition and its sub-partitions. If not specified, the default compression level of the partition will match that of its parent.
- 6 If the parent table or partition is RANGE partitioned, the added partition must include a START clause (this is not optional as it is in CREATE TABLE).
- 7 The new partition will inherit appropriate properties from the parent. For instance:
  - The new partition's default values will match the parent.
  - Any NOT NULL or CHECK constraints will apply to the new partition.
  - The new partition's storage type (ROW or COLUMN) will match the parent.
  - All indexes will be extended to cover the new partition.
  - The new partition will use the distribution key of the parent, if any.
  - All ownership and permission properties of the table will apply to the new partition.If ADD PARTITION is performed followed by DETACH, all the above properties will match on the detached table.

## ALTER TABLE...DROP PARTITION

The ALTER TABLE...DROP PARTITION operation drops an existing partition and all of its data from a logically partitioned table. If the partition to be dropped is a subpartition, the reference to it must include references to all partitions above it in the hierarchy (i.e. *partition\_name.subpartition\_name*). If the partition to be dropped includes subpartitions, they will be deleted as well, along with their data.

### Examples

Assume a table “p2” with subpartitions “p21”, “p22” and “p23”. The following example drops the subpartition “p21” only:

```
ALTER TABLE records DROP PARTITION(p2.p21);
```

The following example drops the partition “p2” and all of its subpartitions:

```
ALTER TABLE records DROP PARTITION(p2);
```

### Some additional details for ALTER TABLE...DROP PARTITION

Only the table owner (or the superuser) may drop a partition from a table.

## ALTER TABLE...ATTACH PARTITION

The ALTER TABLE...ATTACH PARTITION operation allows an existing table to become a partition of a logically partitioned table. The tables need to have the same columns and distribution key. All permissions and indexes on the table being attached will be wiped out and replaced with the corresponding attributes of the parent table.

The main use case for ATTACH is adding a child partition, by attaching an empty table with the same columns as the partitioned table. ATTACH can also be used to reattach a partition

that was previously detached because of a requirement to perform an action on only the child partition. You might also need to attach a partition to add archival data back into a partitioned table for reporting purposes.

### **Examples**

The following example attaches a partition “north\_america” to an existing table of distributors partitioned by a list of country names:

```
ALTER TABLE distributors
    ATTACH PARTITION north_america (
        VALUES ('US', 'Canada')
    ) FROM north_america_distributors;
```

The following example attaches a partition “year\_2013” to an existing table of records partitioned by timestamp range:

```
ALTER TABLE records ATTACH PARTITION year_2013
    (START '2013-01-01' END '2014-01-01')
FROM records_2013;
```

### **Some additional details for ALTER TABLE...ATTACH PARTITION**

- 1 If the table uses list partitions, the ALTER TABLE...ATTACH PARTITION must also use a list partition. The same rule applies for range partitions. You do not specify the partition expression, since it must be the same as the existing child partitions.
- 2 When attaching partitions, they need to satisfy the same consistency criteria described for CREATE TABLE. This includes being mutually distinct from the previously existing child partitions.
- 3 The tables do not need to reside in the same schema. Any table in a schema other than the current schema must be schema qualified.
- 4 The database user must be an owner of both tables, and must possess the USAGE privilege for the schema(s).
- 5 The table to be attached may not be a temporary table.

### **ALTER TABLE...DETACH PARTITION**

The ATLER TABLE...DETACH PARTITION operation detaches a partition from a logically partitioned table. The partition is created as a new table with the provided name in the same schema as the original parent table, unless another schema is specified.

### **Example**

In the following example, we DETACH a partition before dropping it:

```
ALTER TABLE records DETACH PARTITION (daily_1) INTO daily_temp;
DROP TABLE daily_temp;
```

### **Some additional details for ALTER TABLE...DETACH PARTITION**

When detaching a partition, you should be aware of the following properties that the table created from it will have:

- 1 The new table will have the same column definitions and constraints as the logically partitioned table. There will also be one or more additional CHECK constraints that

correspond to the individual partition constraints prior to the DETACH operation. These extra CHECK constraints allow a table to be detached and reattached without extra validations of the partition constraints. If the extra CHECK constraints are not needed, they can be dropped. The names of the constraints are system generated, and can be viewed using `\d <tablename>` in ACT.

- 2 The owner and permissions on the detached table will match the logically partitioned table.
- 3 The detached table will have a set of indexes with definitions that match the indexes on the logically partitioned table. The indexes will have system-generated names, which can be viewed using `\d <tablename>` in ACT.
- 4 If there are any SERIAL or BIGSERIAL columns in the table, the detached table will share the sequence generator(s) used by the logically partitioned table (its former parent). Insertion into either table will increment the shared sequence. This allows the detached table to be re-attached as a partition to the parent table at a later time.  

Note that the sequence will be deleted if the logically partitioned parent table is dropped with the CASCADE option, meaning that any tables that used that sequence will not continue to automatically generate serial column values. For more information on how serial datatypes work, see the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.
- 5 For a list of properties the new table inherits, see [Inheritance with Automatic Logical Partitioning \(page 47\)](#).

## **ALTER TABLE...ALTER PARTITION...RENAME TO**

The ALTER TABLE...ALTER PARTITION...RENAME TO operation allows you to rename a partition.

### ***Example***

For example, suppose a logically partitioned table “Records” has a child partition “jan2000”. We can rename this to “january2000” using the following statement:

```
ALTER TABLE Records ALTER PARTITION(jan2000) RENAME TO january2000;
```

Now suppose PARTITION (january2000) has a subpartition called “europe”. We can change the name of that subpartition to “eu” as follows:

```
ALTER TABLE Records ALTER PARTITION(january2000.europe) RENAME TO eu;
```

## **ALTER TABLE ... ALTER PARTITION ... COMPRESS**

The ALTER TABLE...ALTER PARTITION...COMPRESS [ HIGH | MEDIUM | LOW ] operation allows you to compress a partition. Similarly, you can remove compression from a partition using ALTER TABLE...ALTER PARTITION...NOCOMPRESS.

### ***Example***

This example compresses the partition “jan2000”:

```
ALTER TABLE Records ALTER PARTITION(jan2000) COMPRESS LOW;
```

## ALTER TABLE... Errors

The following error messages are shared by all the preceding ALTER TABLE commands:

- If the table (or referenced partition) is not logically partitioned, Aster Database will return the error message "<relation> is not logically partitioned".
- If the table or partition reference does not exist, Aster Database will return the error message "<relation> does not exist".

## Inheritance with Automatic Logical Partitioning

A partition inherits the following from its parent table:

- column names and their datatypes
- distribution key constraints
- constraints (e.g. NOT NULL)
- owner
- permissions
- index definitions
- compression (if not explicitly specified for the partition)
- sequence generators for columns of type SERIAL and BIGSERIAL

## Command Cascading with Automatic Logical Partitioning

The following SQL commands operate on the hierarchy as if it were one table. Issue the command against the top level table, and it will automatically cascade to the correct child partition(s):

- INSERT
- CREATE INDEX
- SELECT
- UPDATE
- DELETE
- MERGE
- COPY
- TRUNCATE
- VACUUM
- ANALYZE
- CLUSTER
- REINDEX
- ALTER
- GRANT, and
- REVOKE

## More Detailed Logical Partitioning Examples

The following examples apply automatic logical partitioning to some real world scenarios to give you more detailed information about how they work in practice and how their use impacts performance.

### Create a logically partitioned table with index

Suppose we are constructing a database for a large website. The company measures traffic volumes every day as well as impression volumes per page in each topic. Conceptually, we want a fact table defined as shown below. Recall that Aster Database's CREATE TABLE creates a fact table by default, so we can type CREATE FACT TABLE or we can type simply CREATE TABLE.

The main use of this table will be to prepare online reports for clients. We know that most queries will access just the last week's, month's or quarter's data. In this situation we can use partitioning to help us meet all of our different requirements for the impressions table.

Following the steps outlined above, automatic logical partitioning can be set up like this, using a single SQL statement:

- 1 The base table is the impressions table, declared below:

```
CREATE TABLE impressions (
    impression_id    int NOT NULL,
    page_id          int NOT NULL,
    logdate          date NOT NULL,
    topic_id         int)
```

- 2 Next, we create a distribution key on impression\_id:

```
DISTRIBUTE BY HASH(impression_id)
```

- 3 Next, we create one partition for each active month, with non-overlapping ranges, using PARTITION BY:

```
PARTITION BY RANGE (logdate) (
PARTITION impressions_yy04mm02 (END '2004-03-01' ),
PARTITION impressions_yy04mm03 (END '2004-04-01' ),
...
PARTITION impressions_yy05mm11 (END '2005-12-01' ),
PARTITION impressions_yy05mm12 (END '2006-01-01' ),
PARTITION impressions_yy06mm01 (END '2006-02-01' ))
```

- 4 And finally, we close the statement:

```
) ;
```

- 5 We add indexes on the topic\_id column in the partitioned table. These are automatically inherited by its partitions:

```
CREATE INDEX idx_topic_id_impressions ON impressions (topic_id);
```

### Create a logically partitioned table from data in another table

When creating a logically partitioned table using data from another table, you cannot create the table using CREATE TABLE...AS SELECT in one statement as you would with a flat table. You must first create the table with its partitions, and then INSERT INTO the partitioned table. Data will automatically be inserted into the correct partitions.

In this example we will create a partitioned table “sales\_by\_country” to hold sales data by region:

```
CREATE FACT TABLE sales_by_country
(
    id int,
    country varchar,
    ts timestamp,
    revenue int
)
DISTRIBUTE BY HASH(id)
PARTITION BY LIST(country)
(
    PARTITION region1( VALUES('India', 'Japan') ),
    PARTITION region2( VALUES('Australia', 'New Zealand') ),
    PARTITION region3( VALUES('Denmark', 'Ireland') )
);
```

Then we will insert data from an existing table “all\_sales” into the partitioned table by selecting the appropriate rows:

```
INSERT INTO sales_by_country
SELECT * FROM all_sales
WHERE country IN ('India', 'Japan', 'Australia', 'New
Zealand', 'Denmark', 'Ireland');
```

## Performance and Automatic Logical Partitioning

Queries benefit from the logical partitioning of data through Child Table Pruning (“partition pruning” or “pruning” for short). Pruning is a process by which the query planner eliminates (prunes) portions of a table in a query before reading the rows into memory and performing any operations on them. For tables with automatic logical partitioning, entire partitions may be pruned, enabling much better performance.

Consider the following query, which selects a count of the impressions occurring within the month of December 2005:

```
SELECT count(*) FROM impressions WHERE logdate >= DATE '2005-12-01' and
logdate < DATE '2006-01-01';
```

The optimizer knows that this query will find qualifying rows only in the `impressions_yy05_mm12` partition, and hence completely skips scanning all of the other partitions. If the `impressions` table contains one year of data, we can finish the query by looking at only a twelfth of the data!

Such optimization works best when the query's WHERE clause contains constants. But for queries whose WHERE clauses can be evaluated to constants at runtime, some optimization still applies. A discussion of some of the special considerations that apply to partition pruning follows.

### Dynamic vs. static pruning

There are two times at which partition pruning may occur:

- Compile Time partition pruning occurs when all input values are known prior to execution.
- Execution Time (Dynamic) partition pruning can only be done at runtime, because the query depends on one or more inputs whose value is not known until execution.

It is quite common for Business Intelligence (BI) tools like Microstrategy to write queries that use period and time dimensions. These queries cannot be optimized until execution time. This is because they use a schema design that has a fact table with a column whose values correspond to a date column in a lookup table. In this case, queries cannot perform partition pruning at compile time, because the fact table contains date lookups, not actual dates. The time dimension must be scanned before the fact table can be pruned. Similar problems can arise with ad hoc queries where the user does not know or take into account how constraint exclusion works.

The optimizer performs dynamic child table pruning as in the following example. Suppose that the following query is issued:

```
SELECT * FROM D AS d JOIN F AS f ON d.timestamp1 = f.timestamp1 WHERE d.day = '20011009';
```

Suppose also that table F is partitioned on column timestamp1. The WHERE clause isn't on timestamp1, so we cannot directly prune any child tables of F. However, if we scan table D, we can select only rows where d.day = '20011009'. Then we can collect the timestamp1 values for all of the rows that we selected from D, which then allows us to eliminate some of the child tables of F. The same approach can be used in some JOINs that involve more than one table.



**Tip!** For Aster Database to prune any partitions, the fact table must be partitioned on the column that is used in the JOIN.

---

The planner will optimize queries that join tables with logical partitioning using indirect pruning. That is, it is smart enough to know that if the WHERE clause explicitly filters the dimension table, it can also prune the table on the other side of the JOIN in the same way. This assumes that the partitioned table is partitioned on the column used in the JOIN.

Note the following when using dynamic child table partitioning:

- 1 To take advantage of dynamic partitioning, you must use an INNER JOIN. Dynamic pruning does not occur with OUTER JOINS.
- 2 The optimizer generally treats the partitioned table as being on the "left" side of a JOIN, using the table on the "right" side to determine which partitions on the "left" side table can be pruned. However, the optimizer makes this choice; you cannot control whether a table is treated as being on the "left" vs. "right" in a JOIN.
- 3 Dynamic pruning applies to Hash Joins (or Merge Joins, with the proper hints). Hints that influence the type of join that will be used are `enable_hashjoin` and `enable_merge_join`. Dynamic pruning will not be applied to Nested Loop Joins.

- 4 The optimizer hints are indeed "hints", not rules. For example, you can tell the optimizer to use merge joins and not hash joins or nested loop joins, but it may choose a hash join anyway.
- 5 IMPORTANT: If a transfer (to other nodes) is required in order to JOIN the fact table with the rest of the query, then pruning will not occur. To avoid this type of transfer, most queries should do one of the following:
  - Replicate the dimension table, so that joins with the fact table partitions can occur on the same node as the node that hosts the particular partition; or
  - Partition the dimension table (or the second fact table, if you're using 2 fact tables) the same way that the fact table was partitioned.
- 6 The Aster Database optimizer uses cost-based heuristics to choose the most efficient execution plan for a given query. The optimizer will not always choose dynamic pruning whenever possible, because there might be a plan with lower expected overall cost, which does not perform dynamic pruning.
- 7 Dynamic pruning may be disabled. Toggling the dynamic pruning feature on and off enables you to measure the performance improvement that occurs with dynamic pruning in order to tune your queries. Dynamic partition pruning can be configured by changing the variable `constraint_exclusion` in the SET command. There are three supported settings:
  - `constraint_exclusion = 'ON'` enables all forms of child table pruning. This is the default setting.
  - `constraint_exclusion = 'STATIC'` just disables dynamic pruning, and still allows static child table pruning to occur.
  - `constraint_exclusion = 'OFF'` disables all child table pruning.

### Use EXPLAIN to find out if pruning will occur

When testing queries, you can display the pruning information using EXPLAIN. An example is shown below. When pruning will occur, the EXPLAIN plan shows a "Partition Filter". To make the following example interesting, the table is partitioned in two levels. Each level is pruned by a separate join operator. The two hash joins each build a partition filter. One builds `pf_0` on the partition key (`b`) of `myroot` and the other builds `pf_1` on (`a`). The EXPLAIN output shows both partition filters being applied on the scan of the `myroot` hierarchy.

```
EXPLAIN SELECT *
  FROM myroot r, dim1 x, dim2 y
 WHERE r.a = x.a AND r.b = y.b;
                                         QUERY PLAN
-----
Hash Join  (cost=430.40..1246327.32 rows=96802445 width=52)
  Hash Cond: (r.a = x.a)
  Partition Filter: build pf_1 on myroot(a)
    -> Hash Join  (cost=203.73..31044.00 rows=2010435 width=48)
        Hash Cond: (r.b = y.b)
        Partition Filter: build pf_0 on myroot(b)
          -> Append  (cost=0.00..567.00 rows=46700 width=40)
              Partition Filter: apply pf_0
              Partition Filter: apply pf_1
              -> Seq Scan on myroot r  (cost=0.00..56.70 rows=4670 width=40)
```

```

-> Seq Scan on ppc1 r  (cost=0.00..56.70 rows=4670 width=40)
-> Seq Scan on gc11 r  (cost=0.00..56.70 rows=4670 width=40)
-> Seq Scan on gc12 r  (cost=0.00..56.70 rows=4670 width=40)
-> Seq Scan on ppc2 r  (cost=0.00..56.70 rows=4670 width=40)
-> Seq Scan on gc21 r  (cost=0.00..56.70 rows=4670 width=40)
-> Seq Scan on gc22 r  (cost=0.00..56.70 rows=4670 width=40)
-> Seq Scan on ppc3 r  (cost=0.00..56.70 rows=4670 width=40)
-> Seq Scan on gc31 r  (cost=0.00..56.70 rows=4670 width=40)
-> Seq Scan on gc32 r  (cost=0.00..56.70 rows=4670 width=40)
-> Hash  (cost=96.10..96.10 rows=8610 width=8)
    -> Seq Scan on dim2 y  (cost=0.00..96.10 rows=8610 width=8)
-> Hash  (cost=106.30..106.30 rows=9630 width=4)
    -> Seq Scan on dim1 x  (cost=0.00..106.30 rows=9630 width=4)

```

## Automatic Logical Partitioning FAQs

### **How do I run SELECT (or UPDATE, DELETE, etc.) on a single child partition?**

UPDATE and DELETE are issued on the top level partitioned table, and automatically act on the appropriate partitions. However, you can constrain a query to a single child by using an equivalent WHERE filter. For example, suppose that you have an `orders` table which is split into months based on an `order_date` column. To only select from the partition containing data for January 2011, issue the following statement:

```
SELECT order_date, sum(total_cost) FROM orders WHERE order_date >=
'2011-01-01' AND order_date < '2011-02-01'
```

A word of warning: if the `orders` table has many children, query planning time will be longer than it would be on a single table without children. The query planning time will be impacted even more if the query contains many references to a logically partitioned table (e.g. through a UNION ALL).

### **Can I use UPDATE to move a row to another partition?**

UPDATE cannot migrate data between child partitions. If you attempt to move a row from one child partition to another child partition, an error will be produced and the UPDATE will not be performed.

### **How do I run ANALYZE (or VACUUM, CLUSTER etc.) on a single child partition?**

The partition must first be separated via DETACH, the command is then run on the resulting table, then the table is added back to its parent as a partition with ATTACH. Note that if these commands run in a single transaction, the parent table will be locked and completely unavailable to other queries. If these commands run in separate transactions, be aware that the detached partition will be completely dissociated with the parent table, so it will not participate in SELECT queries until reattached. An example of this workflow is as follows:

```
ALTER TABLE orders DETACH PARTITION (jan11) INTO orders_jan11;
ANALYZE orders_jan11;
ALTER TABLE orders ATTACH PARTITION jan11(START '2011-01-01' END '2011-
02-01');
```

## How do I interpret the output of the EXPLAIN command?

Each child partition accessed by a SELECT query will be displayed as a separate operation (SeqScan, IndexScan, etc). The operation will show the internal name of the partition. An example of an internal partition name is `_bee_p768_northamerica` which could be seen for a partition named `northamerica`. The number "768" in the name is just an arbitrary ID which is attached to make the name unique across the database.

## What partitions are there on my table?

When using ACT, a `\d` command will not display information about logical partitioning. Instead, complete information on the partitions is available by accessing system tables. Queries on the table `nc_user_child_partitions` will be useful in most cases, but there are additional system tables related to logical partitioning.

For example, this will show all the partitions that lie beneath the `orders` table:

```
SELECT p.* FROM nc_user_child_partitions p, nc_user_tables t WHERE t.tablename = 'orders' AND t.tableid = p.tableid;
```

If the table has a multi-level partitioning, the previous query will show all levels. To just show the first level below `orders`, change the selection to use the `parentid` of `nc_user_child_partitions`:

```
SELECT p.* FROM nc_user_child_partitions p, nc_user_tables t WHERE t.tablename = 'orders' AND t.tableid = p.parentid;
```

These queries will return nothing if `orders` has zero children. A logically partitioned table may have zero children, but it will still remember the column and format of the partitioning (for example, PARTITION BY LIST(`country`)). To see the partition definition of a table without looking at children, use the `nc_user_parent_partitions` system table:

```
SELECT p.* FROM nc_user_parent_partitions p, nc_user_tables t WHERE t.tablename = 'orders' AND t.tableid = p.parentid;
```

## What is the simplest way to split a table into date ranges?

There are some techniques that can simplify the CREATE TABLE syntax:

- Include an initial child table to contain "old" data prior to the partitioned range.
- Omit the START clause to just assume the end of the previous range.
- Use date arithmetic for the END values, just adding the proper period of time from the beginning of the range.

Creating weekly partitions using this method would look something like this:

```
CREATE TABLE orders(....)
...
PARTITION BY RANGE (order_date) (
    PARTITION old_orders(END date '2010-01-01'),
    PARTITION week1(END date '2010-01-01' + interval '1 week'),
    PARTITION week2(END date '2010-01-01' + interval '2 week'),
    PARTITION week3(END date '2010-01-01' + interval '3 week'),
    PARTITION week4(END date '2010-01-01' + interval '4 week')
    ...
);
```

## How many child partitions should I create?

When a table has a large number of child partitions, some operations on the table will be slower, such as the DROP TABLE command. The planning stage of a SELECT query can also become slower, although the extra planning time can pay off if child partitions are excluded from the execution plan. The following rules of thumb can help choose the number of child partitions:

- 1-200 child partitions: The overhead work for maintenance commands and query planning will usually be insignificant.
- 200-2000 child partitions: It may be possible to notice some degradation for certain operations, but these negative effects should be weighed against the benefits to data management and query execution.
- More than 2000 child partitions: The benefits of logical partitioning do not usually extend beyond 2000 child partitions. A table with many thousands of partitions will still be fully functional, but it is advisable to measure the performance of critical operations before choosing this approach. A good way to do this is to measure the changes in execution times on empty tables, which will indicate the worst-case penalty of using many child partitions. If loading performance is important, it should also be benchmarked separately.

Note that these rules of thumb are applicable to tables with ROW storage. Tables with COLUMN storage may have more overhead from logical partitioning, particularly with DROP, TRUNCATE, and loading performance. The limits depend on the table's column definitions, so there is no concise rule of thumb for the number of child tables. It is worthwhile to benchmark loading performance for logically partitioned columnar tables before finalizing a design.

## How can I create a "catch-all" or "default" partition that contains any unmatched records?

There is no straightforward way to do this with the PARTITION BY clause. It is sometimes possible to do this with equivalent RANGE or LIST constraints, but it would be necessary to keep these constraints up to date if the sibling partitions are subject to change.

# Columnar Tables

As an alternative to row-oriented tables, Aster Database supports tables with a column-oriented storage layout. This option is available for distributed, replicated, and logically partitioned tables, including temporary and persistent tables. These tables were append-only in Aster Database version 4.6. Beginning in version 5.0, you can use VACUUM FULL and CLUSTER on column-oriented tables.

To create a column-oriented table, you append the 'STORAGE COLUMN' clause at the end of the CREATE TABLE statement. For example, this statement creates a column-oriented table:

```
CREATE TABLE films
(
    code integer,
    title varchar(40) NOT NULL,
```

```
    did integer NOT NULL,  
    date_prod date  
)  
DISTRIBUTE BY HASH(code)  
STORAGE COLUMN  
COMPRESS;
```

### The STORAGE clause

The CREATE TABLE statement accepts the STORAGE ROW or STORAGE COLUMN flag to specify the desired storage layout of the table:

- STORAGE ROW indicates the table will be a traditional row-wise table. This is the default.
- STORAGE COLUMN indicates the table will use a column-oriented storage layout.

If no STORAGE clause is supplied, it defaults to STORAGE ROW.

## When to Use Columnar Tables

The obvious benefit of columnar tables is the fact that, for a given query, only the required columns will be fetched from disk. In some situations, this can substantially reduce the amount of I/O required to answer a given query. Thus, as a general rule of thumb, if a majority of your queries access a low percentage of the columns of a given table, then it may be a good candidate for columnar. Of course there are other limitations which may preclude the usage of columnar (see the next section for details).

The performance gains you'll see will depend on the non-trivial cost associated with transposing values from a column-wise layout to a row-wise layout during retrieval. This cost goes up with the total width of columns selected. As the total width of the selected columns increases, the cost of I/O and transposition can exceed that of straight selection from a row table. Note that we say "column width," not "number of columns."

For example, consider a table that consists of five columns, of which four are integer-typed columns, and the other is a very wide, varchar-typed column. If most of your queries select only the integer columns (and even if they select all of the integer columns), then it makes sense to have the table be a columnar table. Doing so allows Aster Database to store the wide, varchar values separately from the other columns, so that queries can load the other columns without paying the price of scanning over the wide values.

To summarize, it makes sense to consider using a columnar table if:

- your table does not need any of the operations which are unsupported in columnar; and
- the majority of the queries going against this table in your workload access only a few of the columns.

See more tips in the section, "["Use Columnar Tables When Appropriate"](#) on page 186.

## Recommendations When Using Columnar Tables

As a best practice, you should use CLUSTER for tables using columnar storage if the following are all true:

- 1 The data rarely changes (no UPDATE, no DELETE, and few INSERT statements or load/COPY operations).
  - 2 A large percentage of the queries use aggregate functions.
  - 3 A large percentage of the queries use ORDER BY or GROUP BY on a specific column (which is of course the column on which we would want to create the index that will be used to CLUSTER).
  - 4 VACUUM FULL and CLUSTER on columnar tables may run Out of Memory. The workaround for this is to execute a dummy ALTER TABLE command that triggers a rewrite of the table. That's preferable to CTAS because it preserves permissions, indexes, and constraints. For example:
    - Choose a column of the table which is not the distribution key.
    - Determine its data type (which can be shown using the \d command in ACT). For this example, suppose there is a column X of type bigint.
    - Execute an ALTER TABLE command that sets the column type to its current type, For example:
- ```
ALTER TABLE t ALTER COLUMN X TYPE bigint;
```

## When to Avoid Using Columnar Tables

The actual benefit, if any, seen from columnar tables depends heavily on the kind of query you are running. For example, if your query requires a significant, non-trivial amount of work besides the sequential scan of the input tables (for example, if Aster Database must redistribute the data or sort the data (GROUP BY / ORDER BY)), then the I/O of sequentially scanning a row table, by itself, may not be the biggest bottleneck in your query's execution.

As a rule of thumb, if the sequential scan of input tables is not a significant bottleneck in your query, then columnar tables are unlikely to make the query run faster.

As specified above, the cost of I/O and transposing values from a columnar table may exceed straight selection from a row table if you're selecting a large portion of the total table width. This means that if a large percentage of your queries are simply doing a `SELECT *`, or selecting a large number of columns, you may be better off using traditional row tables. In addition, there are certain operations that Aster Database supports for row tables that are not supported on columnar tables. If a table needs these operations, then define it as a row table. See “[Unsupported Operations on Columnar Tables](#)”, below.

## How Queries Change When Using Columnar Tables

A table's columnar storage format is essentially transparent to the query that runs on it. The following SQL constructs are supported on columnar tables:

- CREATE TABLE:
  - CREATE and CREATE TABLE ... AS SELECT (also known as “CTAS”)
  - with and without compression
  - partitioned and replicated tables
  - persistent and temporary tables

- with and without PRIMARY KEYS
- CREATE INDEX (Supported index access methods are: btree and gist. Bitmap indexes are also supported)
- INSERT and INSERT INTO ... SELECT (also known as “IIS”)
- COPY TO and COPY FROM:
  - with autopartitioning
  - with error logging
- DROP and TRUNCATE
- ALTER TABLE (on a columnar table, only the following ALTER commands are supported):
  - ALTER TABLE .. NOCOMPRESS | COMPRESS [HIGH | MEDIUM | LOW]. This is used for altering compression levels
  - ALTER TABLE *tablename* STORAGE [ ROW | COLUMN ]
  - ALTER TABLE .. ADD/DROP COLUMN. Additional column constraints can be specified.
  - ALTER TABLE .. ALTER COLUMN TYPE
  - ALTER TABLE .. SET/DROP DEFAULT
  - ALTER TABLE .. SET/DROP NOT NULL
  - ALTER TABLE .. ADD/DROP CONSTRAINT
  - ALTER TABLE .. INHERIT/NO INHERIT *parent-table*
- CLUSTER
- DECLARE CURSOR (non-updatable, forward-fetch only)
- ANALYZE
- SELECT: All types of queries including SQL/MapReduce queries
- VACUUM and VACUUM FULL (Note that VACUUM does not release any disk space. It does make the space occupied by dead tuples available for future INSERT operations. Even though columnar tables do not support UPDATE and DELETE, VACUUM may still be useful, because rollback operations of INSERT statements may leave dead space in the columnar table.)

The unsupported SQL constructs are listed in the next section.

There are no restrictions on supported data types. Field-length limits are the same as those for row tables. You can involve row and column oriented tables in the same query (in JOINS or sub-SELECTS).

Be aware that you can create temporary columnar tables using the 'CREATE TEMP TABLE' command, but all internal temporary tables that get created as part of Aster Database query execution will always be row tables. The rationale behind this is that these internal tables are created using results from queries run on Aster Database persistent (or other temporary) tables. If the right columns were selected while writing the query, and consequently the right projections were pushed down during query planning, then the selected list of columns will be optimal.

## Load Data into Columnar Tables with Logical Partitioning

A loading operation using the Aster Database Loader Tool, COPY, or INSERT can be expensive when the following conditions exist:

- the target table uses columnar storage, AND
- the target table has many logical partitions, AND
- the loaded data matches many different logical partitions.

In this case, the memory allocated to perform the load may be as large as the amount of source data. To avoid high memory requirements, it is best to divide a large load into batches. There are two alternative approaches:

- Ensure that each batch only loads a small number of logical partitions in the columnar table. For example, when inserting data into a columnar table with weekly partitions, each batch may insert data for a single month.
- Ensure that the size of each batch is a small fraction of system memory available at the worker nodes. This should only be done if the data being loaded into the columnar table has a mixture of records matching many different logical partitions. As an example, suppose that a year's worth of data is being loaded into a columnar fact table with weekly partitions, and the cluster has four physical worker nodes with 100GB of system memory in each node. Loading data in 40GB batches will use 10GB memory per physical worker, which is 10% of the overall available memory.

## Unsupported Operations on Columnar Tables

Below we list the operations that are not supported on column-oriented tables. Where a workaround is available, we list it.

Table 1 - 1: Unsupported operations on column-oriented tables

| Unsupported Feature                               | Workaround                                                                                                                                                                                           |
|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cursors: Backward fetch using cursors             | None                                                                                                                                                                                                 |
| Cursors: Updatable cursors                        | None                                                                                                                                                                                                 |
| DELETE                                            | Combination of CTAS / TRUNCATE / DROP / ALTER RENAME, or, if you know you will need to expire old data, place the data in child tables and DROP each child table when its rows are no longer needed. |
| MERGE                                             | Combination of CTAS / TRUNCATE / INSERT INTO ... SELECT ("IIS") / ALTER RENAME                                                                                                                       |
| Custom field script:<br><code>gettablesize</code> | Use the <code>nc_relationstats</code> function instead.                                                                                                                                              |
| Script:<br><code>ncluster_storagestat</code>      | Use the <code>nc_relationstats</code> function instead.                                                                                                                                              |

## Tables

### Columnar Tables

Table 1 - 1: Unsupported operations on column-oriented tables

| Unsupported Feature | Workaround                                                                                                                                                                                     |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UPDATE              | Combination of CTAS / TRUNCATE / DROP / ALTER RENAME. That is, CREATE a new table AS SELECT FROM the old table, DROP the old table, and RENAME the new table back to the dropped table's name. |

## CHAPTER 2 Aster Database Cluster Terminal (ACT)

This section explains how to use Aster Database Cluster Terminal (ACT) to query and manage databases. ACT is a terminal-based query tool that connects with Aster Database. ACT lets you connect to the database (optionally using SSO and/or SSL), type queries, issue them to Aster Database, and get query results. Alternatively, you can source your queries from a file. ACT can return your query results to the command line or to a file, which makes it useful for extracting data. Meta-commands and shell-like features are provided to facilitate writing scripts and automating tasks.



**Tip!** Beginning with ACT version 4.6, the ACT client cannot connect to versions of Aster Database prior to version 4.6. If you attempt to connect to a pre-4.6 version of Aster Database with a 4.6 or later version of ACT, you will see an error message indicating that there is a version mismatch between Aster Database and the client. You should obtain the version of ACT that matches the version of Aster Database to which you are attempting to connect.

The following sections explain how to launch and use ACT:

- [ACT Quick Start](#)
- [Launch ACT](#)
- [Startup Parameters for ACT](#)
- [How to Use ACT](#)
- [ACT Commands \(at the SQL Prompt\)](#)
- [Troubleshooting ACT](#)

### ACT Quick Start

You run ACT from the command line. The launch command is:

```
$ act -d <db name> -h <hostname> -U <username>  
[-w <password>] [argument flags]
```

where

- *db name* is your database name,

- *hostname* is the name or IP address of the queen,
- *username* is your SQL login name,
- *password* is your SQL password in Aster Database (this is optional; ACT prompts you for a password if you do not pass a -w parameter), and
- *argument flags* are any of the parameter/value combinations listed in “[Startup Parameters for ACT](#)” on page 63.

For example:

```
$ act -d emea_sales -h 10.48.58.100 -U mjones
```



**Tip!** When using SSO (single sign-on), the -U and -w options are **not** used, because the username and password are passed directly to the host via SSO.

To log in to the default database that is provided in your installation, type this, replacing the IP address with the hostname or IP address of your Aster Database queen:

```
$ act -d beehive -h 10.42.52.100 -U beehive -w beehive
```

To see a list of ACT command line arguments, type:

```
$ act --help
```

## Launch ACT

See the appropriate section below for instructions on launching ACT:

- [Launch ACT on Windows](#)
- [Launch ACT on Linux or Solaris](#)
- [Launch ACT on Mac](#)
- [Launch ACT Directly on the Queen](#)



**Tip!** On an Aster Database where LDAP authentication is enabled, if during logon an ACT user gets the error message:

'ERROR: An internal error has occurred.', make sure the username is present in Aster Database with proper privileges.

- [Log In to ACT](#)

### Launch ACT on Windows

- 1 Open a command prompt.
- 2 Change directories to the folder which contains the act.exe executable.
- 3 [Log In to ACT](#).

## Launch ACT on Linux or Solaris

- 1 Ensure that your path includes the directory where ACT is installed.
- 2 [Log In to ACT](#).



**Tip!** ACT for Linux requires glibc version 2.6.18 or higher. If you do not have glibc version 2.6.18 or higher, you must use the IP address instead of the hostname for the `-h` flag when running ACT. To check the version of glibc, issue the command `ldd --version`.

## Launch ACT on Mac

- 1 Open a terminal.
- 2 [Log In to ACT](#).

## Launch ACT Directly on the Queen

Your Aster Database queen also contains an installation of the ACT client. To run it there, you'll need a user account on the queen machine, and you'll need an SSH client on your workstation.

To run the ACT on the queen:

- 1 Open a SSH connection to the queen. If you do not have a user account on the queen, ask the machine's administrator for one.



**Tip!** If you need an SSH client, do one of the following:

- Install the OpenSSH client as explained on the OpenSSH homepage at <http://www.openssh.org/>
- or, for Windows only, install the PuTTY SSH client.

- 2 Change directories to the directory where ACT is installed (by default, `/home/beehive/clients`).

- 3 Log in to ACT:

```
$ act -d <db name> -U <username> -w <password> [argument flags]
```

Note that if you do not provide the hostname using `-h`, ACT defaults to the localhost. For details on the command line options, see “[Startup Parameters for ACT](#)” on page 63

## Log In to ACT

- 1 Run ACT by typing a command like:

```
act -d <db name> -h <hostname> -U <username> [-w <password>]  
[argument flags]
```

For details on the command line options, see “[Startup Parameters for ACT](#)” on page 63

- 2 Provide your database password by:

- adding `-w <password>` to the ACT login string, or
- omitting the `-w` argument and providing your database password at the prompt.

- 3 Choose a database by adding `-d <database name>` to the ACT login string. If `-d` is not used, ACT places you in the system database (with the default name “beehive”).
- 4 You will see a welcome message, followed by the database prompt, which shows the database name, followed by “=>”. For example:

```
Welcome to act 5.10.00.01, the Aster Database Terminal.  
beehive=>
```

## Startup Parameters for ACT

ACT takes a variety of startup parameters when you launch it. (Don’t confuse these with ACT’s at-the-SQL-prompt flags, which you can read about in [“ACT Commands \(at the SQL Prompt\)” on page 75](#).)

You can pass these parameters at the command-line (see [“Using the “on-error-stop” Option in ACT” on page 66](#)) or in a configuration file or [“Use a Configuration File to Pass ACT Startup Parameters” on page 67](#)). The same startup parameter cannot be repeated in a single invocation of ACT, or an error will be returned.

To list the command line parameters, type `act --help`.

Descriptions of each parameter are below:

Table 2 - 1: Summary of the most common command-line parameters for ACT

| Flag                                                                    | Description                                                      |
|-------------------------------------------------------------------------|------------------------------------------------------------------|
| <code>-d [ --dbname ] DBNAME</code>                                     | Specify database name to connect to (default: “beehive”).        |
| <code>-h [ --host ] HOSTNAME</code>                                     | Aster Database server host (default: “localhost”).               |
| <code>-U [ --username ] NAME</code>                                     | Aster Database username (default: “beehive”). Not used with SSO. |
| <code>-l (the letter “ell”)</code><br><code>[ --list-databases ]</code> | List available databases, then exit.                             |



**Tip!** Note the default values for the connection parameters. If you do not specify the parameters `-d` (database name), `-h` (hostname), `-U` (username), and/or `-p` (port) in the connect string, ACT will use the default values. The default values are:

- “beehive” for database name
- “localhost” for hostname
- “beehive” for username, and
- “2406” for port.

If `-w` is not used, ACT will prompt for a password.

Table 2 - 2: General-purpose command-line parameters for ACT

| Flag                                                                         | Description                                                                                                                                                                             |
|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-d [ --dbname ] DBNAME</code>                                          | Specify database name to connect to (default: “beehive”).                                                                                                                               |
| <code>--config-file FILENAME</code>                                          | Lads startup parameters from a configuration file specified by FILENAME. See <a href="#">“Use a Configuration File to Pass ACT Startup Parameters” on page 67</a> for more information. |
| <code>-c [ --single-command ] COMMAND</code>                                 | Run only single command (SQL or internal) and exit.<br>For example:<br><br><code>act -c "COPY MyTable FROM stdin;" &lt; myDataFile.dat</code>                                           |
| <code>-f [ --input-file ] FILENAME</code>                                    | Execute commands from file, then exit. Run a SQL script.                                                                                                                                |
| <code>-1 (the numeral “one”)</code><br><code>[ --single-transaction ]</code> | Execute command file as a single transaction.                                                                                                                                           |
| <code>-l (the letter “ell”)</code><br><code>[ --list-databases ]</code>      | List available databases, then exit.                                                                                                                                                    |
| <code>-? [ --help ]</code>                                                   | Show command line help, then exit.                                                                                                                                                      |
| <code>-v [ --version ]</code>                                                | Output version information, then exit.                                                                                                                                                  |
| <code>--on-error-stop</code><br>or<br><code>-E</code>                        | Enables the “on-error-stop” option, by default this option is disabled. See <a href="#">“Using the “on-error-stop” Option in ACT” on page 66</a> for more information.                  |

Table 2 - 3: Input- and output-related command-line parameters for ACT

| Flag                                                                  | Description                             |
|-----------------------------------------------------------------------|-----------------------------------------|
| <code>-a [ --echo-script-input ]</code>                               | Echo all input from script.             |
| <code>-e [ --echo-all-input ]</code>                                  | Echo commands sent to server.           |
| <code>-o [ --redirect-query-results ]</code><br><code>FILENAME</code> | Send query results to file (or   pipe). |

Table 2 - 4: SSL and SSO related command-line parameters for ACT

| Flag                      | Description                                                                                                    |
|---------------------------|----------------------------------------------------------------------------------------------------------------|
| <code>--enable-ssl</code> | Enables Secure Socket Layer (ssl) support. <i>Must</i> be used if any of the other SSL/SSO arguments are used. |

Table 2 - 4: SSL and SSO related command-line parameters for ACT (continued)

| Flag                                        | Description                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>--ssl-encrypt-reads</code>            | SSL Encrypt Reads. <i>Must</i> be used if <code>secureWrites=true</code> on the server. Conversely, <i>must not</i> be used if <code>secureWrites=false</code> on the server. See <a href="#">Setting Configuration Parameters on the Queen (page 40)</a> for information on how to set the <code>secureWrites</code> parameter on the server. |
| <code>--ssl-self-signed-peer</code>         | Indicates that ACT will connect to a Queen which will provide a self-signed certificate.                                                                                                                                                                                                                                                       |
| <code>--ssl-private-key-path PATH</code>    | The SSL Private Key Path indicates where the private key is stored on the client (ACT) machine.                                                                                                                                                                                                                                                |
| <code>--ssl-certificate-path PATH</code>    | The SSL Certificate Path indicates where the certificate is stored on the client (ACT) machine.                                                                                                                                                                                                                                                |
| <code>--ssl-trusted-ca-dir DIRECTORY</code> | When using a chain of certificates rather than a single certificate, use the SSL Trusted CA Dir to set the directory on the client machine where the chain of trusted certificates is stored.                                                                                                                                                  |
| <code>--ssl-trusted-ca-file FILENAME</code> | Use SSL Trusted CA Filename to provide the location of the signed copy of the server's certificate on the client machine.                                                                                                                                                                                                                      |
| <code>--ssl-cert-filetype ARG</code>        | SSL Certificate File Type (use 1 for PEM; 2 for ANS1; default: 0).                                                                                                                                                                                                                                                                             |
| <code>--enable-sso</code>                   | Enables Single sign-on (SSO) support.                                                                                                                                                                                                                                                                                                          |
| <code>--gss-lib-path PATH</code>            | For Linux, sets the GSS shared library path (default on linux is /opt/guest/lib32 or /opt/guest/lib64). Ignored on Windows.                                                                                                                                                                                                                    |



**Tip!** The SSL settings in ACT have interdependencies, and in most cases they rely on the SSL settings on the queen. See [Setting up SSL for ACT \(page 52\)](#).

Table 2 - 5: Output format-related command-line parameters for ACT

| Flag                                      | Description                                                                                                                          |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <code>-q [ --quiet ]</code>               | Run quietly and do not print messages, only query output. Use this for clean query output. Often used with the <code>-c</code> flag. |
| <code>-t [ --print-rows-only ]</code>     | Print rows only.                                                                                                                     |
| <code>-x [ --expanded ]</code>            | Turn on expanded table output.                                                                                                       |
| <code>-A [ --unaligned ]</code>           | Turn on unaligned table output                                                                                                       |
| <code>-F [ --field-separator ] ARG</code> | Set field separator (default: ' ')                                                                                                   |

Table 2 - 6: Connection-related command-line parameters for ACT

| Flag                                    | Description                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-h [ --host ] HOSTNAME</code>     | Aster Database queen hostname or IP address (default: "localhost"). Note that ACT supports glibc version 2.6.18 or higher. If you do not have glibc version 2.6.18 or higher, you must use the IP address instead of the hostname. To check the version of glibc, issue the command <code>ldd --version</code> .                                                                    |
| <code>-p [ --port ] PORT</code>         | When using SSO, you should specify a fully qualified hostname using the <code>-h</code> option, as in the example: <code>&lt;hostname&gt;. &lt;domain&gt;. &lt;com   org   etc&gt;</code> . If only the hostname is used with SSO, ACT will append the local domain name before attempting to look up the host. Using an IP address with <code>-h</code> is not supported with SSO. |
| <code>-U [ --username ] USERNAME</code> | Aster Database username (default: "beehive").                                                                                                                                                                                                                                                                                                                                       |
| <code>-w [ --password ] PASSWORD</code> | Aster Database password. This parameter is optional; ACT will prompt for a password if you do not pass a <code>-w</code> parameter. Not used with SSO.                                                                                                                                                                                                                              |

## Using the “on-error-stop” Option in ACT

The “on-error-stop” option can be used to stop ACT if an error occurs while running queries. It discontinues executing the multi-statement SQL or the SQL file.

When the “on-error-stop” option is set, ACT halts query processing and exits as follows:

- If ACT is in interactive mode, it returns to the ACT prompt when meeting an error.
- If ACT is executed in the command line, it exits with status “3” when meeting an error.

**Note!** Unless the “`-f`” or “`-c`” options are used, ACT is always in interactive mode once it is launched. The “`\set`” command can be used at the command line at any time. For more details on the `\set` syntax for the “on-error-stop” option, see “[Database Parameters Set with \set](#)” on page 80.

### Setting “on-error-stop” in the command line

To enable the “on-error-stop” option (by default this option is disabled) from the command prompt, use either of the following parameters:

- `--on-error-stop`
- `-E`

For example, to enable the “on-error-stop” in a running session of ACT, type the following at the ACT prompt:

```
beehive$ ./act -h 153.65.197.120 -U beehive -w beehive
-E -f test_queries.sql
```

```
beehive$ ./act -h 153.65.197.120 -U beehive -w beehive
--on-error-stop -f test_queries.sql
```

## Use a Configuration File to Pass ACT Startup Parameters

Specifying all the necessary startup parameters for ACT on the command line can become cumbersome, especially when using SSL. Because of this, Teradata Aster has provided a way to specify startup parameters in a configuration file, to streamline starting ACT. On Windows and Linux/UNIX-based operating systems, ACT looks for the configuration file in a specific location and loads it automatically. Alternatively, the configuration file may be invoked using the `--config-file` parameter when launching ACT.

All of the startup parameters in ACT are supported in the config file, *except* for the following:

- `--config-file`
- `-V [ --version ]`
- `-? [ --help ]`

To use a configuration file, first create a text file of startup parameters. The following rules apply when creating the config file:

- 1 Lines starting with a `#` character are ignored (considered as comments).
- 2 Blank lines are ignored (including lines containing just spaces).
- 3 Parameters are entered using the format

`flagname: value`

where `flagname` is same as the name of the command line flag without the preceding hyphens (`--`) and `value` is the flag value as it would be provided on the command line. Note that the short notations of flags are not supported. For example:

`host: <ip>`

will work but the following:

`h: <ip>`

will not work.

- 4 Flags which do not take any argument on the command line should be given a value of either `true` or `false`.
- 5 Flag names are case-sensitive.
- 6 If the config file includes invalid flag names or repeated entries, ACT will not launch, and an error will display.
- 7 If the config file includes the “`on-error-stop`” option with the parameters set to enable this option, ACT will stop if an error occurs while running SQL queries. See [Set “on-error-stop” in the ACT config file \(page 67\)](#) for information on setting this option.

### Set “on-error-stop” in the ACT config file

To enable the “`on-error-stop`” option, add it to the ACT config file: `actconfig.ini` and set the value to `“true”`. Set the value to `“false”` to turn off this option. By default, this feature is disabled or `“off”`.

The following is an example from of a config file for ACT:

```
# ACT configuration file example
# Contains setting to enable on-error-stop
host: 10.10.10.10
dbname: sampledb
username: beehive
on-error-stop: true
```

The default location for the config file on Linux/UNIX-based systems is \$(HOME) /.actconfig. For Windows, the default location for the config file is %HOMEPATH% / actconfig.ini. Upon launching, ACT will look for the config file in the default location, and if found, will use it by default.

Command line flags can be specified when starting ACT, in addition to the flags in the configuration file. If a flag is present in the config file and specified at the command line as well, the value on the command line will override the value in the config file. ACT will notify the user that the command line flag was used upon startup.

The following is an example of a config file for ACT:

```
# ACT configuration file example
# Contains settings for connecting securely to a specific host and
database

host: 10.10.10.10
dbname: sampledb
username: sampleuser

# SSL settings
enable-ssl: true
ssl-self-signed-peer: true
ssl-encrypt-reads: false
```

To start ACT, explicitly using the config file, issue a command like this example:

```
$ act --config-file /home/beehive/.act_ssl_config
```

To start ACT, explicitly using the config file and also specifying an additional parameter to redirect query results to a file for this session only, issue a command like this example:

```
$ act --config-file /home/beehive/.act_ssl_config -o /home/beehive/
query_results_file
```

## How to Use ACT

### Issue SQL Queries

To use ACT to run a SQL query against Aster Database, you can do one of the following:

- Enter the query at the ACT command line
- Run the query from a file

However, before you can run queries, you must first launch ACT, as described in “[Launch ACT](#)” on page 61.

For example, to run queries against the retail\_sales database, run this command on the Queen to launch ACT:

```
[root@localhost ~]# act -d retail_sales -U beehive -w beehive
Welcome to act 5.10.00.01, the Aster nCluster Terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with act commands
      \g or terminate with semicolon to execute query
      \q to quit

retail_sales=>
```

To list the tables in the database, enter \d at the ACT prompt (in this case, `retail_sales=>`). For example:

```
retail_sales=> \d
      List of relations
 Schema |      Name       | Type | Owner
-----+-----+-----+-----+
 public | customer_dim | table | beehive
 public | date_dim    | table | beehive
 public | geo_dim     | table | beehive
 public | product_dim | table | beehive
 public | region_dim  | table | beehive
 public | sales_fact   | table | beehive
 public | store_dim   | table | beehive
(7 rows)
```

## Run queries from the command line

To control how ACT handles errors when running single or multi-line SQL queries from the command line, see “[Using the “on-error-stop” Option in ACT](#)” on page 66.

To run a single-line SQL query from the command line:

- 1 Type the query at the ACT prompt.

Note! The query must end with a semicolon.

- 2 Press Enter.

For example, to list all the rows of the `customer_dim` table of the `retail_sales` database and order the results based on gender, enter this command at the ACT command-line prompt:

```
retail_sales=> select customer_id, gender, city_id from customer_dim
order by gender;
      customer_id | gender | city_id
-----+-----+-----+
        743 | F      |      46
      2711 | F      |     124
        744 | F      |      66
```

To run a multi-line SQL query from the command line:

- 1 Type the first line of the query, then press Enter.

The ACT prompt changes from => to -> (for example, `retail_sales->`).

- 2 Enter the remaining lines.

You can press Ctrl-C at any time to abandon this command mode without executing the query.

3 On the last line, type a semicolon at the end of the line, then press Enter to run the query.

For example:

```
retail_sales=> select customer_id, gender, city_id  
retail_sales->from customer_dim  
retail_sales->order by gender;
```

or

```
retail_sales=> select customer_id, gender, city_id  
retail_sales->from customer_dim  
retail_sales->order by gender  
retail_sales->;
```

## Run queries from files

To control how ACT handles errors when running SQL queries from a file, see “[Use a Configuration File to Pass ACT Startup Parameters](#)” on page 67.

To run a query stored in a file:

1 Make sure the filename ends with .sql.

The query does not have to end with a semicolon.

2 At the prompt, enter this command:

```
\i filename
```

For example, consider this query stored in the file myQuery.sql:

```
SELECT *  
FROM customer_dim  
ORDER BY gender
```

To run this query, enter this command at the ACT command prompt:

```
retail_sales=> \i myQuery.sql
```

You can also run this query using the -f flag of the act command. For example:

```
[root@localhost ~]# act -d retail_sales -w beehive -f myQuery.sql
```

## Workaround for Multibyte Characters

ACT does not accept multibyte characters from the SQL prompt. For example, the following statement will produce a syntax error:

```
beehive=> create database db 测试;
```

The workaround to input multibyte characters is as follows:

1 Input the multibyte characters from an SQL file instead of at the SQL prompt:

a Create a file containing the SQL to insert the multibyte data (for example, insert\_file.sql).

b Execute it by issuing \i insert\_file.sql.

2 Or, use the SQL command buffer in ACT:

a Input the data using the query buffer, by issuing the ACT \e command, which brings up the default text editor.

- b Type in the SQL, including the multibyte characters.
- c Execute the SQL by exiting the text editor (for example by typing :wq if your default text editor is vi.)

## Exit ACT

To quit ACT, type \q and hit <Enter>.

## Page Through Query Results

When ACT returns results to the screen, it (by default) prints one page of results at a time. Hit the space bar to display the next page of results, hit <Enter> to display just one more line of results, and type “q” to quit looking at results and return to the SQL prompt. When you get to the last page of results, ACT displays the text, “(END)”. Type “q” to return to the SQL prompt.

## Throttle Query Results in ACT and Aster Database

To reduce the memory footprint of ACT and other Aster Database clients, you can set a *fetch-count* that constrains the number of rows returned at one time when you select from a large table. You can limit the total number of rows returned by using *fetch-limit*. Additionally, you can make your query results stream more efficiently by using the *server-side cursors* feature. In ACT, you can set these parameters using \set, and in other clients you can typically set them in your data source definition or parameters file.

Let's look at fetch-count first.

### Reduce Memory Use With Fetch-Count

The purpose of using fetch-count is to reduce the memory footprint of ACT on the server. Therefore, this type of configuration would normally be done by an administrator, or at least the setting would be made under an administrator's guidance. To set a fetch-count, you specify a value for the *fetch-count* parameter in ACT. This causes ACT to fetch rows in sets, with each set containing only the user-specified number of rows (or fewer rows).

ACT uses a fetch count by default, i.e. even when *fetch-count* is not set explicitly. The *fetch-count* (number of rows per fetch) should always be set to greater than 0, The default value is 1024 rows.

To set the *fetch-count* in a running session of ACT, use the \set command to set the *fetch-count* parameter. To do this, type the following at the ACT prompt:

```
\set fetch-count n
```

where *n* is the maximum number of rows ACT should return at a time.

To enforce the *fetch-count*, ACT uses server side cursors to fetch results, which can help prevent the memory footprint of ACT from growing too large.

Note that to the user, the results returned will not be different when using *fetch-count*. The purpose is simply to reduce the memory footprint of ACT on the server.

## Limit Caching With Server-Side Cursors

Setting `use-server-cursors` has the same effect as declaring a server-side cursor with the `DECLARE CURSOR` syntax in SQL. When server-side cursors are activated, one batch of data is retrieved at a time. This lets the queen avoid caching the entire result set and lets the workers continue performing computations while the queen is retrieving rows. As the result set becomes available from a worker, the queen sends it to the client.

Depending on the type of query, results can start streaming back to the client immediately (e.g. `SELECT * WHERE...`) or the results may have to be computed in their entirety (e.g. `GROUP BY`) before streaming to the client. Regardless of whether the results begin to stream to the client before the query has finished executing, server-side cursors can improve performance significantly for large result sets. If you want to start streaming results of a query as soon as possible, you can benefit from setting `use-server-cursors` and specifying a `fetch-count`. By default, `use-server-cursors` is set to 0 (off).

## Example of Fetch-Count With Server-Side Cursors

The following steps are an example of how to use server-side cursors and fetch-count together:

- 1 Enable server cursors by issuing the following command inside ACT (a setting of 1 turns cursors “on” and a setting of 0 means “off”):

```
beehive=> \set use-server-cursors 1
```

- 2 Set a `fetch-count` of 100:

```
beehive=> \set fetch-count 100
```

- 3 Issue the following SQL query:

```
SELECT A,B,COUNT(*) FROM lineitem GROUP BY a, b;
```

This will actually cause the following statements to be run:

```
BEGIN;
DECLARE x CURSOR FOR SELECT A,B,COUNT(*) FROM LINEITEM GROUP BY A,B;
FETCH 100 FROM x; // 100
FETCH 100 FROM x; // 200
...
...
FETCH 100 FROM x; // until all the rows have been fetched
CLOSE x;
END;
```

Aster Database reports all the fetched rows as the row count for this query.

## Set the Maximum Number of Rows Returned With Fetch-Limit

To limit the *total number of rows returned by each query*, set the `fetch-limit` in ACT. `fetch-limit` (the total number of rows fetched for a query) can be set to any value. A value less than

0 implies fetch all rows. A value greater than 0 implies `fetch-limit` rows in total. The default value is -1 (all rows).



**Tip!** When `fetch-limit` is used, the total row count returned for the query will be the total row count returned by the query or the row count specified by `fetch-limit`, whichever is smaller. For example, if a query normally returns 35,453 rows, but you have specified a fetch limit of 1000, the query will return 1000 rows (and it will display "1000 rows returned"). There will be no indication that there were in fact 35,453 rows that would have been returned had you not had a fetch limit set.

In ACT, you set the `fetch-limit` by typing:

```
\set fetch-limit n
```

where *n* is the maximum number of rows a single query will be allowed to return.

### The SQL LIMIT Clause vs. Fetch-Limit and Fetch-Count

Setting a `fetch-limit` is an alternative to specifying a `LIMIT` clause explicitly in your `SELECT` query, with an important difference in how the query will be executed, which can affect performance. `fetch-limit` is an ACT feature which controls how many rows are fetched by the queen from the workers and by the client from the queen. ACT stops fetching results from the cursor after the specified limit is reached. `LIMIT` is a SQL clause which controls how many rows are computed on the worker(s) or the queen.

The following examples illustrate the difference between using `fetch-limit` and using a `LIMIT` clause in the SQL query:

Consider the statement:

```
SELECT * FROM FOO, BAR where FOO.A = BAR.A;
```

#### **Example 1: Using `fetch-limit` and `fetch-count`**

To issue the query in ACT with a `fetch-count` of 100 and a `fetch-limit` of 1000:

```
\set fetch-count 100
\set fetch-limit 1000
SELECT * FROM FOO, BAR where FOO.A = BAR.A;
```

Here is how the query will execute:

- 1 ACT opens a cursor on the query.
- 2 The queen activates Parallel Cursors and passes the query

```
SELECT * FROM FOO, BAR where FOO.A = BAR.A;
```

to the workers.
- 3 Each worker then computes the entire equi-join as dictated by the `FOO.A = BAR.A` constraint.
- 4 The queen fetches results from workers in batches of 100 rows, until a limit of 1000 is reached.
- 5 ACT fetches results from the queen in batches of 100 rows, until a limit of 1000 is reached.

### **Example 2: Using a LIMIT clause in the SQL query**

To issue the same query in ACT using the SQL LIMIT clause to limit results to 1000 rows:

```
SELECT * FROM FOO, BAR where FOO.A = BAR.A LIMIT 1000;
```

Here is how the query will execute:

- 1 ACT issues the SQL statement to the queen.
- 2 The queen planner identifies `LIMIT 1000` and pushes down the following query to each of the workers:

```
SELECT * FROM FOO, BAR where FOO.A = BAR.A LIMIT 1000;
```

- 3 Each worker computes the equi-join as dictated by the `FOO.A = BAR.A LIMIT 1000` constraint. This allows for optimizations where the entire equi-join computation need not be done at each worker.

- 4 The queen fetches results from the workers.
- 5 ACT fetches results from the queen.

The difference between the two approaches occurs in step 3 of both examples. Using the SQL `LIMIT` clause (Example 2), the workers are allowed to compute with the constraint of `LIMIT 1000`, whereas in Example 1 they have to compute the entire equi-join.

So, as you can see, `fetch-limit` via server-side cursors does not translate into the workers doing a `LIMIT 1000` on their individual slice of data. Therefore, if the use case calls for it, an Aster Database power-user should be aware that using the SQL `LIMIT` clause can speed up query execution dramatically in Aster Database.

## **ACT Utility Commands**

ACT also offers many utility commands that provide assistance and carry out useful utility operations such as uploading files and changing output options. These commands start with a backslash character, and we describe them in the section, “[ACT Commands \(at the SQL Prompt\)](#)” on page 75.

## **Repeat Previously Typed Commands**

Hit the up arrow on your keyboard to toggle through your history of previously typed commands. To edit a previously typed command, just use the usual combination left arrow, right arrow, delete, and backspace keys to make your edits, and type <Enter> to finish editing the line. If needed, type “;” to issue the command. To list your command history on Linux, type `\s`.



**Tip!** If you receive an “Error writing history to file.” error on Linux when attempting to view command history with `\s`, check that the current Linux user has permissions to write to the current working directory.

## **Tab Completion**

The UNIX/Linux version of ACT can tab-complete SQL commands and table names that you type. Tab-completion is *not* available in the Windows version of ACT.

To use this feature, type the first couple of letters of a command and hit the <Tab> key. If the completion is unambiguous, ACT completes the command. If ACT doesn't complete the command, hit <Tab> again and ACT prints all the possible completions. Using the list as a reference, type enough additional characters to unambiguously identify the desired command or table, and hit <Tab> again to complete it. Here are a few common uses of tab completion:

- To complete common SQL commands. For example, type “se” and hit <Tab> to type SELECT.
- To list various ACT utility commands. For example, type “\” and hit <Tab> to show all the commands, or type “\d” and hit <Tab> to show all the commands that start with “d”.
- To complete a table name. For example, type “SELECT \* FROM sa” and hit <Tab> to complete the table name or hit <Tab> twice show all the table names that start with “sa”. You can also list the names of all tables in the database by typing “SELECT \* FROM ” (note the trailing space) and hitting <Tab> twice.

## ACT Commands (at the SQL Prompt)

The ACT SQL prompt accepts a number of ACT-specific commands that you issue by typing a backslash followed by a character or combination of characters and arguments. Do *not* type a semicolon to conclude these commands! Hitting <Enter> executes the command.

Note! Don't confuse these with the ACT command-line startup parameters (also known as “command line flags”), which you type at the shell command line when you launch ACT. For a list of those, see [“Startup Parameters for ACT” on page 63](#).

We describe the SQL-prompt commands in the tables that follow:

- [General-purpose utility commands in ACT](#)
- [Environment settings in ACT](#)
- [Query Buffer Commands in ACT](#)
- [Input/output-related commands in ACT](#)
- [Installed-function and installed-file management commands in ACT](#)
- [Informational commands in ACT](#)
- [Formatting-related commands in ACT](#)

Table 2 - 7: General-purpose utility commands in ACT

| Command | Description                   |
|---------|-------------------------------|
| \?      | prints help for ACT commands. |

Table 2 - 7: General-purpose utility commands in ACT (continued)

| Command                                                                                                                  | Description                                                                                                                                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \c [onnect] DBNAME USER<br>HOST PORT<br>\c [onnect] DBNAME USER<br>HOST<br>\c [onnect] DBNAME USER<br>\c [onnect] DBNAME | change login credentials and/or connect to a new database. The parameters must be specified in the order shown, with a space before each, and parameters may not be skipped. In other words, if only one parameter is specified, it is understood to be DBNAME; if a second parameter is also specified, it is understood to be USER; and so on. |
| \cd [DTR]                                                                                                                | change the current working directory.                                                                                                                                                                                                                                                                                                            |
| \copyright                                                                                                               | show ACT usage and distribution terms.                                                                                                                                                                                                                                                                                                           |
| \h                                                                                                                       | help with SQL commands.                                                                                                                                                                                                                                                                                                                          |
| \h [SQL command name]                                                                                                    | help with syntax of the specified SQL command, * for all commands.                                                                                                                                                                                                                                                                               |
| \g                                                                                                                       | or terminate with a semicolon (;) to execute query.                                                                                                                                                                                                                                                                                              |
| \q                                                                                                                       | quit ACT.                                                                                                                                                                                                                                                                                                                                        |
| \! [command]                                                                                                             | execute command in shell or start interactive shell.                                                                                                                                                                                                                                                                                             |
| \password                                                                                                                | change the password for the current user.                                                                                                                                                                                                                                                                                                        |

Table 2 - 8: Environment settings in ACT

| Command                          | Description                                                                                                                                                                                                                                                                     |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \info                            | display current environment settings.                                                                                                                                                                                                                                           |
| \set                             | display current ACT parameter settings.                                                                                                                                                                                                                                         |
| \set param-name<br>[param-value] | set ACT parameter setting <i>param-name</i> to value <i>param-value</i> . (For example, “\set fetch-count 500” tells ACT to fetch no more than 500 rows at a time when selecting.) If no parameter value is supplied, displays the current setting for the specified parameter. |
| \timing [on off]                 | toggle or set timing of commands.                                                                                                                                                                                                                                               |
| \pager [on off]                  | toggle or set to use pager to enable paging through large result sets.                                                                                                                                                                                                          |

Table 2 - 9: Query Buffer Commands in ACT

| Command   | Description                                                                                                                                                                                               |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \e [FILE] | edit the query buffer (or file) with external editor. On most systems, this launches your default text editor. When you save and exit the editor, the edited statement is passed back to ACT for running. |
| \g [FILE] | send query buffer to server (and results to file or   (pipe character)).                                                                                                                                  |
| \p        | show the contents of the query buffer.                                                                                                                                                                    |

Table 2 - 9: Query Buffer Commands in ACT

| Command | Description                     |
|---------|---------------------------------|
| \r      | reset (clear) the query buffer. |
| \w FILE | write query buffer to file.     |

Table 2 - 10: Input/output-related commands in ACT

| Command           | Description                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \echo<br>[STRING] | write string to query output stream (see \o below).                                                                                                                         |
| \i [FILE]         | execute SQL commands from SQL script file. (Run an SQL script.)                                                                                                             |
| \o [FILE]         | redirect all query results to file or   (pipe character).                                                                                                                   |
| \o                | Type \o with no argument to stop sending results to a file and resume sending them to the ACT shell.                                                                        |
| \s<br>[FILENAME]  | display command history in Linux (optionally, print history to a file specified by FILENAME) Note that query history includes only the first 2048 characters of each query. |

Table 2 - 11: Installed-function and installed-file management commands in ACT

| Command | Description                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \dF     | list installed files, SQL-MapReduce functions, and other functions in the current schema. Use a regular expression as an argument to display a subset of the available functions. For example, to view all installed functions in the database, issue :<br><br>\dF *.*<br>where the first asterisk means "all schemas" and the second means "all functions and files."                                                          |
| \dF+    | show details for all installed files, SQL-MapReduce functions, and other functions in the current schema. For each function, the output shows the name, schema, owner, upload time, and MD5 Hash fingerprint of the function.<br><br>Use a regular expression as an argument to display a subset of the available functions. For Example, type \dF+ *.* to show details for functions and files in all schemas in the database. |
| \dE     | show all the installed SQL-MR functions for which the current user has privileges. Use a regular expression as an argument to display a subset of the available functions. Shows function name, schema, owner, function version and creation time.                                                                                                                                                                              |

Table 2 - 11: Installed-function and installed-file management commands in ACT (continued)

| Command                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \install<br>[<FILE><br>[ [<SCHEMA>/<br>] <FILE_ALIAS><br>] | install the file or SQL-MapReduce function in Aster Database. The file must be available on the file system where ACT is running. Note that the database user running this command must have permission to install files and functions in the specified schema.<br><br>You cannot install two files or functions with the same name. If attempting to do this, you must follow these steps: <ul style="list-style-type: none"><li>• remove the existing file or function</li><li>• install the new file or function</li><li>• grant the appropriate privileges on the file or function.</li></ul> There is a limit of 238MB on the size of the file to be installed. If you try to install a larger file, you will see an error like:<br><br>ERROR: row text exceeds limit of 238MB ...<br><br>Note that when installing larger files, the queen may run out of memory. The queen needs available memory of approximately eight times the size of the file to be installed, in order to encode, buffer and copy the file. |
| \download<br>[ [<SCHEMA>/<br>] <FILE_ALIAS><br>] <FILE>    | download the specified installed file or function (identified by its <i>FILE</i> or <i>FILE_ALIAS</i> ) to the machine where ACT is running. Note that the database user running this command must have permission to download files and functions from the specified schema.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \remove<br>[ [<SCHEMA>/<br>] <FILE_ALIAS><br>]             | remove from the cluster the file or SQL-MapReduce function specified by its <i>FILE_ALIAS</i> . Note that the database user running this command must have permission to remove files and functions from the specified schema.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

Table 2 - 12: Informational commands in ACT

| Command       | Description                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------|
| \d            | list all tables, indexes and views in the current schema.                                                         |
| \d [PATTERN]  | describe table or index.                                                                                          |
| \dt           | list all tables in the current schema.                                                                            |
| \dt [PATTERN] | print schema, name, type, and owner of a table or tables. To see tables in a custom schema, type \dt schemaname.* |
| \dv           | list views.                                                                                                       |
| \dv [PATTERN] | describe view.                                                                                                    |
| \di           | list indexes.                                                                                                     |
| \di [PATTERN] | describe index.                                                                                                   |
| \dg           | list groups and roles.                                                                                            |
| \dg [PATTERN] | describe group.                                                                                                   |

Table 2 - 12: Informational commands in ACT (continued)

| Command                                                                            | Description                                                                      |
|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| \du                                                                                | list users.                                                                      |
| \du [PATTERN]                                                                      | describe user.                                                                   |
| \dn                                                                                | list schemas.                                                                    |
| \dn [PATTERN]                                                                      | describe schema.                                                                 |
| \l                                                                                 | list all databases.                                                              |
| \extl host=hostname_or_IP<br>[option_name=option_value,<br>...]                    | lists all databases on an external Hadoop systems.                               |
| \extd host=hostname_or_IP<br>database=dbname<br>[option_name=option_value,<br>...] | lists all tables in a database or describe a table on an external Hadoop system. |
| show sessionid                                                                     | shows the session identifier for the current session.                            |



**Tip!** In Aster Database 5.0, for the \extd command in ACT, if the optional user argument is not specified, the command will fail on any but the default database. The error message is not specific about what caused the command to fail. The workaround is to always specify the argument user when issuing \extd.



**Tip!** ACT uses the schema search path (search\_path) for the database user when displaying lists of tables, views and indexes. The schema search path defaults to the schema search path for the current user in the database. To set the search\_path from ACT, issue the following command:

```
beehive=> SET session search_path TO <schema>;
```

Note that multiple schemas are not supported. If multiple schemas are listed in the search\_path, the first schema listed will be used.

To display the current search\_path type:

```
beehive=> SHOW search_path;
```

Note that you may also set the search\_path on the server.

Alternatively, you can specify the schema to use when issuing commands by following the command with a schema qualified reference. This example shows how to display information on all tables in the schema "myschema":

```
\dt myschema.*
```

Table 2 - 13: Formatting-related commands in ACT

| Command     | Description                                             |
|-------------|---------------------------------------------------------|
| \a          | toggle between unaligned and aligned output mode.       |
| \f [STRING] | show or set field separator for unaligned query output. |
| \t [on off] | show only rows (off by default).                        |

Table 2 - 13: Formatting-related commands in ACT (continued)

| Command     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \x [on off] | set or toggle <i>expanded output</i> mode ON and OFF. With <i>expanded output</i> mode turned on, each record is split into rows, with one row for each value, and each new record is introduced with a text label in the form, -- [ RECORD 37 ] --. This can help make wide tables readable on a small screen, and is very useful if you're trying to read EXPLAIN output. Note that in expanded mode, the number of rows is not returned at the end of the table. Because of this, when querying a table with no rows, you will simply see the ACT prompt again. |

## Database Parameters Set with \set

The syntax to use the \set command is:

```
\set [ name [ value [ ... ] ] ]
```

Table 2 - 14: Database parameters

| Parameter                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| auto-commit [1 0]        | When set to 1 (the default, on), each SQL command is automatically committed upon successful completion. When set to 0 (off), you may manually commit your changes after each transaction or series of transactions by issuing the COMMIT command, or undo changes by issuing ROLLBACK. If you do not issue the COMMIT command, all transactions that occurred since the last COMMIT will rollback automatically.                                                                                                              |
| fetch-count [int]        | To limit the number of rows returned at a time. ACT uses a fetch count by default (i.e. even when <code>fetch-count</code> is not set explicitly.) The <code>fetch-count</code> (number of rows per fetch) should always be set to greater than 0, The default value is 1024 (1024 rows).                                                                                                                                                                                                                                      |
| fetch-limit [int]        | To set the maximum number of rows returned per query. A value less than 0 implies fetch all rows. A value greater than 0 implies <code>fetch-limit</code> rows in total. The default value is -1 (all rows).                                                                                                                                                                                                                                                                                                                   |
| use-server-cursors [1 0] | When set to 1, sets the server to use cursors (useful when the result set is very large). When set to 0 (the default, off), sets the server to not use cursors.                                                                                                                                                                                                                                                                                                                                                                |
| on-error-stop [1 0]      | By default, this feature is disabled or set to off = 0. When set to 1 (or "on") ACT will stop and exit if it meets an error during SQL query processing.<br>The following are ACT exit messages: <ul style="list-style-type: none"><li>• EXIT_SUCCESS = 0 means ACT finished processing normally.</li><li>• EXIT_FAILURE = 1 means an error occurred, such as "file not found" in the "-f" option.</li><li>• EXIT_USER = 3 means an error occurred in a sql script and the option "on-error-stop" was on or enabled.</li></ul> |

## Troubleshooting ACT

### ACT Connection Hangs When Using SSL

When ACT is configured to use SSL, the queen must also be set up to use SSL. If the queen is not setup for SSL and ACT tries to connect using SSL, the connection will fail. An error is not returned; the ACT client just hangs. If you experience this, check to ensure that SSL settings on the queen and in ACT match.

### Invalid User Name Error in ACT After Password Change

When changing the user's password using `ALTER USER <username> PASSWORD '<new_password>' ;` in ACT or in a SQL script running through ACT, you will see an ALTER USER successful message returned. However, ACT does not change the cached password for the user. Because of this, if you then connect as the user and issue another query, you may see an error related to invalid username and password. This is because ACT is using the cached username and password when attempting to establish a new session.

If you encounter this, change the password using the `\password` command in ACT instead. If you change the password using `\password`, your ACT connection will continue to work.

### Misleading Error Message Reports Problem With a Role Instead of With a User

When attempting to connect to a database as a user who does not have connect privileges on that database, the error incorrectly reports that the problem is with a role instead of with the user. This example illustrates the error message:

```
beehive=> create user kris with password 'beehive';
CREATE USER
beehive=> create database foo;
CREATE DATABASE
beehive=> \c foo kris;
Password for "kris":
act: ERROR:  role "kris" does not have connect privileges on this
database.
Previous connection kept
```

## CHAPTER 3 **SQL-MapReduce and Stream**

The Aster Database in-database MapReduce framework lets you develop functions and run them in close proximity to their input data sources, enabling you to analyze large data sets efficiently. Aster Database in-database MapReduce provides these main features:

- SQL-MapReduce: The Aster Database In-Database MapReduce framework, known as SQL-MapReduce (or SQL-MR), allows you to write and upload your own functions and run them in parallel on Aster Database for efficient data analysis.
- Stream API: The Aster Database Stream API allows you to run scripts and functions written in other languages including Python, Ruby, Perl, and C#.

This chapter is divided into the following sections:

- [Introduction to SQL-MapReduce](#)
- [SQL-MapReduce Query Syntax](#)
- [SQL-MR with Multiple Inputs](#)
- [SQL-MapReduce Java API](#)
- [SQL-MapReduce Examples in Java](#)
- [SQL-MapReduce C API](#)
- [Install and Use a Sample Function](#)
- [Manage SQL-MapReduce Execution](#)
- [Troubleshooting and SQL-MR Errors](#)
- [Stream API for Python, Perl, and Other Types of Scripts](#)
- [Using the R Programming Language and Environment](#)
- [SQL-MapReduce Security](#)
- [Manage Functions and Files in Aster Database](#)
- [FAQs About SQL-MapReduce and Stream](#)

# Introduction to SQL-MapReduce

## What is MapReduce?

*MapReduce* is a framework for operating on large sets of data using MPP (massively parallel processing) systems. The basic ideas behind MapReduce originated with the “map” and “reduce” functions common to many programming languages, though the implementation and application are somewhat different on multi-node systems.

MapReduce enables complex analysis to be performed efficiently on extremely large sets of data, such as those obtained from weblogs, clickstreams, etc. It has applications in areas such as machine learning, scientific data analysis and document classification.

In computer programming languages, a “map” function applies the same operation to every “tuple” (member of a list, element of an array, row of a table, etc.) and produces one output tuple for each input tuple it operates on. It is sometimes called a “transformation” operation. On an MPP database such as Aster Database, the “map” step of a MapReduce function has a special meaning. In this case, the input data is broken up into smaller sets of data, which are distributed to the worker nodes in a cluster, where an instance of the function operates on them. Note that if the data is already distributed as specified in the function call, the distribution step will not occur, because the function can operate on the data where it is already stored. The outputs from these smaller groups of data may be redirected back into the function for further processing, input into another function, or otherwise processed further. Finally, all outputs are consolidated again on the queen to produce the final output result, with one output tuple for each input tuple.

A “reduce” function in computer programming combines the input tuples to produce a single result by using a mathematical operator (like sum, multiply or average). Reduce functions are used to consolidate data or aggregate it into smaller groups of data. They can accept the output of a map function, a reduce function, or operate recursively on their own output.

In Aster Database, the “reduce” step in MapReduce works a little differently, as follows:

- 1 The input data is first be partitioned by the given partitioning attribute.
- 2 The tuples are then distributed to the worker nodes, if required by the function call, with all the tuples that share a partitioning key assigned to the same node for processing.
- 3 On each node, the function operates on these tuples, and returns its output to the queen. Note that the function itself may output the same, a larger or a smaller number of tuples than contained in the input dataset it received.
- 4 The output from each node is consolidated on the queen. Additional operations may be performed on the queen at this time. For example, if the function performs an average, the average results from all the nodes must be averaged again on the queen to obtain the final output.
- 5 The final output is then returned by the SQL-MR function.

## Aster Database SQL MapReduce

The Aster Database In-Database MapReduce framework, known as SQL-MapReduce or SQL-MR for short, lets you write functions in Java or C, save these functions in the cluster, and allow analysts to run them in a parallel fashion on Aster Database for efficient data analysis. Analysts invoke a SQL-MR function in a SELECT query and receive the function's output as if the function were a table. A SQL-MR function takes as input one or more sets of rows from tables or views (for example, the contents of a table in the database, the output of a SQL SELECT statement, or the output of another SQL-MR function) and produces a set of rows as output. Beginning in Aster Database 5.0, SQL-MR functions can now accept multiple inputs. For more on this, see "[SQL-MR with Multiple Inputs](#)" on page 87.

Because a call to a SQL-MR function results in a set of parallel tasks being run across the cluster, the input data provided to a SQL-MR function must be divided across the parallel tasks. SQL-MapReduce supports three kinds of inputs:

- 1 We call a SQL-MapReduce function a single input row function if it takes a single row-wise input. When you invoke a row function, you provide it rows in any order. In your SQL statement that calls the row function, you write "ON `my_input`", where `my_input` is your input table. The row function operates at the granularity of individual rows of the `my_input` table. This corresponds to a "map" function in traditional map-reduce systems. The Aster Database SQL-MR API for a function that accepts row-wise input is the *RowFunction* interface.
- 2 We call a SQL-MapReduce function a single input partition function if it takes a single partition-wise input, in which rows are clustered/grouped together by a specified key of one or more columns. In your SQL statement that calls the partition function, you write "ON `my_input` PARTITION BY `partitioning_attributes`" to specify that the function operates on rows sharing a common value of column(s) `partitioning_attributes` of `my_input`; the function has access to all such rows at once, enabling more complex processing than possible with row-wise inputs. Within each partition, you can sort rows using an ORDER BY clause. An Aster Database partition function corresponds to a "reduce" function in traditional map-reduce systems. The Aster Database SQL-MapReduce API for a function that accepts partition-wise input is the *PartitionFunction* interface.
- 3 A SQL-MR function that accepts multiple inputs is called a multiple input function. It can include a cogroup operation in which inputs from multiple sources are partitioned and combined before being processed, a dimension operation where all rows of one or more inputs are replicated to each vworker, or a combination of both. In your SQL statement that calls the multiple inputs function, you write a combination of the following, to specify each input and how its rows are to be distributed:
  - "ON `my_input` PARTITION BY `partitioning_attributes`" for each input where rows are to be partitioned among vworkers using the specified columns,
  - "ON `my_input` PARTITION BY ANY" for an input where rows can be processed wherever they were stored when the function was called, and/or
  - "ON `my_input` DIMENSION" for each input where all rows are to be replicated to all vworkers.

There are rules governing which types of inputs and how many of each type can be specified in the same multiple input function call. See [Rules for number of inputs by type \(page 89\)](#). The Aster Database SQL-MapReduce API for a function that accepts multiple inputs is the *MultipleInputFunction* interface.

In summary, a SQL-MapReduce function:

- is a function that uses the Aster Database API ( Java and C are the supported languages);
- is compiled outside the database, installed (uploaded to the cluster) using Aster Database ACT, and invoked in SQL;
- receives as *input* (from the ON clause(s)) some rows of one or more database tables or views, pre-existing trained models and/or the results of another SQL-MR function;
- receives as *arguments* zero or more argument clauses (parameters), which can modify the function's behavior;
- returns output rows back into the database;
- is polymorphic. During initialization, a function is told the schema of its input (for example, (key, value)) and how it needs to return its output schema;
- is designed to run on a massively parallel system by allowing the user to specify which slice of the data (partition) a particular instance of the function sees.

## SQL-MapReduce Query Syntax

Beginning in Aster Database version 5.0, the SQL-MapReduce function syntax is extended to allow one or more partitioned inputs and zero or more dimensional inputs. This has introduced some important changes to the syntax for SQL-MR functions. For more information, see [“SQL-MR with Multiple Inputs” on page 87](#)

Invoking a SQL-MR function has the following syntax in SQL:

```
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
      * | expression [ [ AS ] output_name ] [, ...]
FROM sqlmr_function_name
      ( on_clause
        function_argument
      ) [ [ AS ] alias ]
[, ...]
[ WHERE condition ]
[ GROUP BY expression [, ...] ]
[ HAVING condition [, ...] ]
[ ORDER BY expression [ ASC | DESC ][ NULLS { FIRST | LAST } ] [, ...] ]
[ LIMIT { count | ALL } ]
[ OFFSET start ];

```

where *on\_clause* is:

*partition\_any\_input* | *partition\_attributes\_input* | *dimensional\_input*

where *partition\_any\_input* is:

`table_input PARTITION BY ANY [ORDER BY expression] | table_input [order_by]`

where *partition\_attributes\_input* is:

`table_input PARTITION BY partitioning_attributes [order_by]`

where *dimensional\_input* is:

`table_input DIMENSION [order_by]`

where *table\_input* is:

`ON table_expression [ AS alias ]`

where *table\_expression* is:

`{ table_name | view_name | ( query ) }`

The synopsis above focuses only on the use of SQL-MapReduce. See SELECT in the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference* for a complete synopsis of SELECT, including the WHERE, GROUP BY, HAVING, ORDER BY, LIMIT, and OFFSET clauses.

Notes:

- `sqlmr_function_name` is the name of a SQL-MapReduce function you have installed in Aster Database. If the `sqlmr_function_name` contains uppercase letters, you *must* enclose it in double quotes! To see if the function is installed, see the instructions in “[Manage Functions and Files in Aster Database](#)” on page 140.
- The `on_clause` provides the *input data* the function operates on. This data is comprised of one or more partitioned inputs and zero or more dimensional inputs. The partitioned inputs can be a single `partition_any_input` clause and/or one or more `partition_attributes_input` clauses. The dimensional inputs can be zero or more `dimensional_input` clauses.
- `partition_any_input` and `partition_attributes_input` introduce expressions that partition the inputs before the function operates on them.
- `partitioning_attributes` specifies the partition key(s) to use to partition the input data before the function operates on it.
- `dimensional_input` introduces an expression that replicates the input to all nodes before the function operates on them.
- `order_by` introduces an expression that sorts the input data after partitioning, but before the function operates on it. It is always optional.
- The `table_input` clause includes an `alias` for the `table_expression`. For rules about when an alias is required, see [Rules for table aliases \(page 90\)](#). When declaring an alias, the `AS` keyword is optional.

- `function_argument` optionally introduces an argument clause that typically modifies the behavior of the SQL-MapReduce function. Don't confuse *argument clauses* with *input data*: Input data is the data the function operates on (see "ON," above), while argument clauses usually just provide runtime parameters. You pass an argument clause in the form `argument_name (literal [, ...])` where `argument_name` is the name of the argument clause (as defined in the function) and `literal` is the value to be assigned to that argument. If an argument clause is a multi-value argument, you can supply a comma-separated list of values. You can pass multiple *argument clause blocks*, each consisting of an `argument_name` followed by its value(s) encased in a single pair of parentheses, separated from the next argument clause block with whitespace (not commas). See "[Argument Clauses for Java Functions](#)" on page 100 for details.
- `AS` provides an alias for the SQL-MR function in this query. Using an alias is optional, and, when declaring an alias, the `AS` keyword is optional.

## SQL-MR with Multiple Inputs

Beginning in Aster Database version 5.0, Teradata has extended the capabilities of SQL-MR with support for multiple inputs. This allows SQL-MR functions to be applied to related groups of information derived from different data sets. Data from these multiple data sources can be processed within a single SQL-MR function. This changes SQL-MR in two important ways:

- 1 Extending the SQL-MR API to accept multiple inputs allows users to write or modify their own SQL-MR function to do more advanced analysis within the function itself. It essentially mimics the JOIN in SQL, but with better performance.
- 2 The Aster Database Analytics Foundation includes functions that take advantage of the new multiple input capabilities. See the *Teradata Aster Big Analytics Appliance 3H Analytics Foundation User Guide* for more information on individual functions.

Additional changes made to SQL-MR to support this feature are aliasing for table expressions and the addition of syntax to allow specifying explicit partitioning requirements when calling SQL-MR functions.

### Benefits of Multiple Inputs

Some benefits of extending SQL-MR to accept multiple inputs are:

- Prediction SQL-MR functions that use a trained model now have better performance and security. The function takes the model itself as a dimensional input, along with one or more data inputs to which it may be applied.
- There is no requirement that all inputs to a SQL-MR function share a common schema.
- The new capabilities avoid JOINs, UNIONs, and the creation of temporary tables, which were often used to work around the older ability to support only a single input.
- New types of analytic functions may now be created more easily (e.g. multichannel attribution).

- Memory is better utilized, because the partitioning and grouping of tuples that occurs before the function operates on them means that less data is actually processed by the function. In addition, the ability to hold one copy of a dimensional input in memory and use it to operate on all tuples from other inputs uses memory more efficiently.

## How Multiple Inputs are Processed

When multiple inputs are supplied, SQL-MR performs a grouping on the partitioned inputs, with optional support for dimensional inputs. For functions containing partitioned inputs (PARTITION BY *partitioning\_attributes*), the following steps occur:

- 1 The *partitioning\_attributes* in all partitioned inputs are examined. A new cogroup tuple is formed for every distinct *partitioning\_attributes* that is found. The cogroup tuple's first attribute will be this *partitioning\_attributes*.
- 2 For each partitioned input, a new attribute is added to the cogroup tuple. This attribute will hold *all* the attributes of each tuple in that input whose *partitioning\_attributes* match the cogroup tuple's *partitioning\_attributes*.
- 3 For each dimensional input, a new attribute is added to the cogroup tuple. This attribute contains *all* of that dimensional input's tuples.
- 4 After the above steps occur, we have one cogroup tuple for each distinct *partitioning\_attributes* with:
  - one attribute being the *partitioning\_attributes*,
  - plus one attribute for each partitioned input that contains a nested array of all of that input's matching tuples,
  - plus one attribute for each dimensional input that includes an array of all of that input's tuples.
- 5 The SQL-MR function then gets invoked on each cogroup tuple.

Comparison semantics are used in this grouping operation, so NULL values are treated as equivalent. Grouped tuples that have empty groups for certain attributes (that is, inputs with no tuples for a particular group) are included in the grouped output by default.

## Types of SQL-MR Inputs

From a semantic perspective, there are two possible types of inputs in SQL-MR:

- 1 Partitioned inputs are split up (partitioned) among vworkers as specified in the PARTITION BY clause. These inputs can specify one of:
  - PARTITION BY ANY - random. Note that in practice, PARTITION BY ANY simply preserves any existing partitioning of the data for that input. There may only be one PARTITION BY ANY input in a function.

- PARTITION BY *partitioning\_attributes* - sorted and partitioned on the specified column(s).



**Note!** All PARTITION BY *partitioning\_attributes* clauses in a function must specify the same number of attributes and corresponding attributes must be equijoin compatible (i.e. of the same datatype or datatypes that can be implicitly cast to match). Note that this casting is "partition safe", meaning that it will not cause any redistribution of data on the vworkers.

- 2 Dimensional inputs use the DIMENSION keyword, and the entire input is distributed to each vworker. This is done because the entire set of data is required on each vworker for the SQL-MR function to run. The most common use cases for dimensional inputs are lookup tables and trained models.

Here's how it works. A multiple-input SQL-MR function takes as input sets of rows from multiple relations or queries. In addition to the input rows, the function can accept arguments passed by the calling query. The function then effectively combines the partitioned and dimensional inputs into a single nested relation for each unique set of partitioning attributes. The SQL-MR function is then invoked once on each record of the nested relation. It produces a single set of rows as output.

## Semantic Requirements for SQL-MR Functions

Keep in mind the following semantic requirements when designing, writing, and calling SQL-MR functions. Note that before applying these rules, Aster Database will assume PARTITION BY ANY for legacy SQL-MR functions whose referencing queries omit the PARTITION BY clause.

### What multiple input structures are allowed?

Your multiple-input function always operates on at least two input sets. There are two alternatives for organizing the input data sets:

- You provide the first input set in a row-wise manner and make the other input set(s) available in their entirety. In Aster terminology, we say that the first set is a PARTITION BY ANY set, and the other sets are DIMENSION sets.
- You provide the first input set partitioned on a key you have chosen, and you provide the other input set(s) partitioned on key(s) and/or as DIMENSION set(s).

That's all you need to know in general about what types of inputs are allowed to work together. The following section lists the specific rules that govern the number and types of inputs that can be used.

### Rules for number of inputs by type

In general, any number of input data sets as specified by ON clause constructs can be provided to a SQL-MR function, but the allowed combinations are governed by the rules listed below.

- 1 A function may have at most one PARTITION BY ANY input.
- 2 If a function specifies PARTITION BY ANY, all other inputs must be DIMENSION.

- 3 Any number of grouping attributes as specified by the PARTITION BY *partitioning\_attributes* clause can be provided, as long as there are no PARTITION BY ANY inputs.
- 4 All PARTITION BY *partitioning\_attributes* clauses must specify the same number of attributes and corresponding attributes must be equijoin compatible (i.e. of the same datatype or of datatypes that can be implicitly cast to match).
- 5 The function will not be invoked if all of the PARTITION BY ANY and PARTITION BY *partitioning\_attributes* inputs are empty. Simply having some data in the DIMENSION inputs is not sufficient to invoke the function in itself. DIMENSION inputs are not first class inputs in this sense; they simply come along for the ride as function arguments might.
- 6 The order of the inputs does not matter. For example, your function could have:

```
SELECT ...
ON store_locations DIMENSION,
ON purchases PARTITION BY purchase_date,
ON products DIMENSION ORDER BY prod_name
...
```

### Rules for table aliases

- 1 An alias is required for subselects.
- 2 If you are referring to a base table or view, an alias is not required. Aster Database will use the table/view name as the default alias.
- 3 If your SQL-MR function refers to a table or view more than once, then a conflict will be reported, as in this example:

```
SELECT * FROM union_inputs(ON t PARTITION BY ANY ON t DIMENSION mode ('roundrobin'));
```

ERROR: input alias T in SQL-MR function UNION\_INPUTS appears more than once

You must give a different alias to each reference to the table or view.

### Number of inputs

A SQL-MR function invocation triggers the following validations:

- The multiple input SQL-MR function expects more than one input.
- The single input SQL-MR function expects exactly one input.

## Use Cases and Examples for Multiple Inputs

There are many types of SQL-MR functions that can benefit from having multiple data inputs. These generally fall into two classifications: cogroup functions and dimensional input functions. Note that these are not mutually exclusive; a single function could include both cogroup and dimensional operations.

## Cogroup Use Case

Cogroup allows SQL-MR functions to be applied to related groups of information derived from different data sets. The different data inputs are grouped into a nested structure using a cogroup function before the SQL-MR function operates on them.

The following use case is a simplified sales attribution example for purchases made from a web store. We want to find out how much sales revenue to attribute to advertisements, based on impressions (views) and clicks leading to a purchase. As inputs, we have the logs from the web store and the logs from the ad server.

This type of result cannot easily be computed using SQL or SQL-MR capabilities without multiple data inputs.

## Cogroup Example

This example uses a fictional SQL-MR function named `attribute_sales` to illustrate how cogroup works. The function accepts two partitioned inputs, as specified in the two ON clauses, and two arguments.

As inputs to the SQL-MR function, we have a `weblog` data set that contains the store's web logs, where we get purchase information. We also have a `adlog` data set that contains the logs from the ad server. We will partition both inputs on the user's browser cookie.

Figure 5: Cogroup Example Tables

Cogroup Example Tables

| weblog |          |          |
|--------|----------|----------|
| cookie | cart_amt | page     |
| AAAA   | \$60     | thankyou |
| AAAA   | \$140    | thankyou |
| BBBB   | \$100    | thankyou |
| CCCC   |          | intro    |
| CCCCC  | \$200    | thankyou |
| DDDD   | \$100    | thankyou |

| adlog  |          |            |
|--------|----------|------------|
| cookie | adname   | action     |
| AAAA   | champs   | impression |
| AAAA   | puppies  | click      |
| BBBB   | apples   | click      |
| CCCC   | baseball | impression |
| CCCC   | apples   | click      |

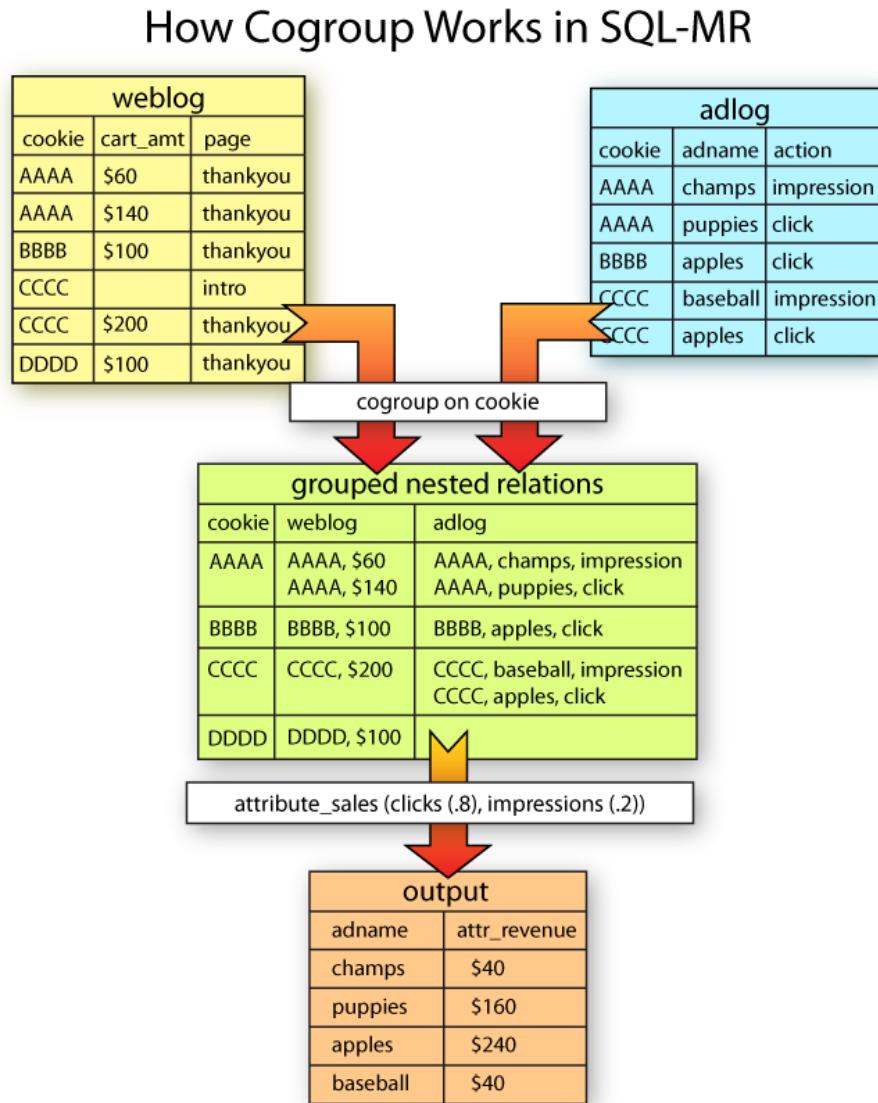
The arguments to the `attribute_sales` function are `clicks` and `impressions`, which supply the percentages of sales to attribute for ad clickthroughs and views (impressions) leading up to a purchase.

We will use the following SQL-MR to call the `attribute_sales` function:

```
SELECT adname, attr_revenue
  FROM attribute_sales (
    ON (SELECT cookie, cart_amt, adname, action
        FROM weblog
       WHERE page = 'thankyou') as W PARTITION BY cookie
    ON adlog as S PARTITION BY cookie
    clicks(.8) impressions(.2))
;
```

The following diagram shows how SQL-MR will execute this function:

Figure 6: How a SQL-MR function performs a cogroup



The two inputs are cogrouped before the function operates on them. The cogroup operation is conceptually performed in two steps.

- 1 Each input data set is first grouped according to the `cookie` attribute specified in the `PARTITION BY` clauses. A “cogroup tuple” is formed for each unique resulting group. The tuple is comprised of the `cookie` value identifying the group, along with a nested relation that contains all values from both the `weblog` and `adlog` inputs which belong to the group. The middle box in Figure 6 shows the output of the cogroup operation.
- 2 The `attribute_sales` function is then invoked once per cogroup tuple. At each invocation it processes the nested relation, treating it essentially as a single row. The function then attributes the sales revenue to the appropriate advertisements as previously described. The bottom box in the diagram shows the output of the SQL-MR function.

Note that the cogroup result includes a tuple for the "DDDD" cookie, even though there is no corresponding group in the adlog data set. This is because Aster Database grouping performs an OUTER JOIN, meaning that cogroup tuples that have empty groups for certain attributes are included in the cogroup output.

### Dimensional Input Use Case: Lookup Tables

A typical scenario for multiple inputs with dimensional data is when a function needs to access an entire lookup table of values for all rows of a given input. To see how this was accomplished prior to multiple input SQL-MR, let's consider the following example. Suppose a query needed to reference a lookup table of values. Prior to the introduction of multiple inputs, if the lookup table was small enough, one of the following strategies could be used:

- Add the lookup table as an additional row in the query table during the query
- Hold the lookup table in memory for the duration of the query.

There are many scenarios like the following, however, for which the above solutions do not work because of the size, complexity or format of the data:

- The data is too big to fit in memory and/or would make the main query table too large and unwieldy if added to it.
- The data input consists of multi-dimensional data, such as geospatial data.
- The data consist of a model, usually in JSON format, and a set of data to be analyzed against, using the model. For a discussion of this scenario, see [Dimensional Input Use Case: Machine Learning \(page 96\)](#).

In these cases, analysts will find it helpful to use SQL-MR with multiple inputs.

The SQL-MR function can loop over the input data - holding one of the inputs in memory and repeatedly performing the same function on each row of another input. Only one single instance of the dimensional input is held in memory on each of the worker nodes, and it is used in processing each incoming row of partitioned data. Prior to this functionality in SQL-MR, this type of data could not easily be processed. That is because an instance of one of the data inputs had to be held in memory for use by each row of data from any additional input. This could cause slow performance if one or both datasets were very large or of a structure not easily represented in a relational form.

### Dimensional Input Example

In this example, we want to create a SQL-MR function to illustrate how dimensional inputs are processed. We'll create the SQL-MR function for a retailer of mobile phone accessories. The function will take data from accessory purchases made by mobile phone and find the closest retail store at the time of purchase. We have two data sets:

- 1 a phone\_purchases data set that contains entries for mobile phone accessory purchases along with normalized spatial coordinates of the mobile phone from the time when an online purchase was made, and
- 2 a stores data set that contains the location of all the retail stores and their associated normalized spatial coordinates.

The following diagram shows the two data sets:

Figure 7: Dimensional Example Tables

| Dimensional Example Tables |        |        |        |        |        |
|----------------------------|--------|--------|--------|--------|--------|
| phone_purchases            |        |        | stores |        |        |
| pid                        | xcoord | ycoord | sid    | xcoord | ycoord |
| p0                         | 2      | 1      | s0     | 1      | 4      |
| p1                         | 1      | 5      | s1     | 2      | 3      |
| p2                         | 3      | 2      |        |        |        |
| p3                         | 0      | 4      |        |        |        |

This type of result cannot easily be computed using basic cogroup capabilities, because the data sets need to be related using a proximity join as opposed to an equijoin. [Figure 8](#) illustrates how this is expressed and executed using cogroup extended with dimensional inputs.

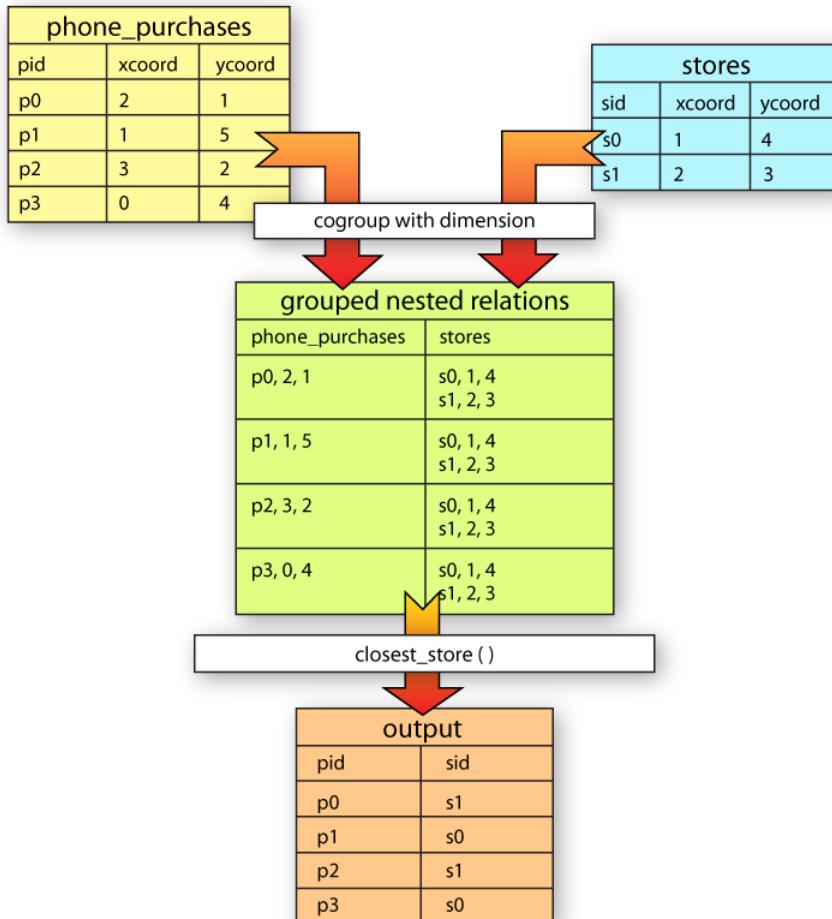
We will create a SQL-MR function named `closest_store`, which accepts two partitioned inputs as specified in two ON clauses. We will use the following SQL-MR to call the `closest_store` function:

```
SELECT pid, sid
  FROM closest_store (
    ON phone_purchases PARTITION BY ANY,
    ON stores DIMENSION)
;
```

The following diagram shows how SQL-MR will execute this function:

Figure 8: How dimensional inputs work in SQL-MR

## How Dimensional Inputs Work in SQL-MR



The `closest_store` SQL-MR function receives the result of a cogroup operation on the `phone_purchases` data set and the dimensional input data set `stores`. The two boxes at the top of the diagram show sample `phone_purchases` input and `stores` input respectively. The operation is conceptually performed in three steps:

- 1 The `phone_purchases` input remains grouped in whatever way it is currently stored in the database, as specified by the `PARTITION BY ANY` clause and the `stores` input is grouped into a single group as specified by the `DIMENSION` clause.
- 2 The groups are combined using what is essentially a Cartesian join. The result of the cogroup operation is a nested relation. Each tuple of the nested relation contains (conceptually) an arbitrary group of phone purchases concatenated with the single group comprised all retail stores. The middle box in the diagram shows the result of the cogroup operation.
- 3 The `closest_store` function is subsequently invoked once per cogroup tuple. At each invocation it receives a cursor over an arbitrary group of purchases along with a cursor over the entire set of stores. The function then performs a proximity join using the

normalized spatial coordinates to find the closest store to each purchase. The bottom box in the diagram shows the output of the SQL-MR function.

## Dimensional Input Use Case: Machine Learning

Machine learning is another common use case for SQL-MR with multiple inputs. In machine learning, you create or choose a “model” that predicts some outcome given a set of data. You typically test the model to determine its accuracy, fine tuning it until its predictions fall within the desired margin of error. The model itself is comprised of mathematical and statistical algorithms created through observations of patterns found within a given dataset.

Let’s assume you want to generate a trained predictive model, fine tune it, and apply it to a large set of data. Imagine that you have ten million emails, and you need to bucket them by subject matter. You would use a function (such as a “decision tree” function) to generate an algorithm that parses emails and places them in the appropriate bucket. The function might do some statistical analysis to determine where clusters of data appear, and create subject matter “buckets” based on these. These emails might be placed into buckets based on frequency of occurrence of certain words, work proximity and/or grammatical analysis.

To test the accuracy of your model, you might have a human being do this same classification work manually for a subset of the emails (called the "sample dataset"). The sample dataset might be one thousand emails. The person would read and classify each email according to the desired criteria. The model is then applied to the sample data set and the results compared to the known outcome (the results generated by the human being). You can then gauge the reliability of the predictive model and fine tune it. Finally, you can do the analysis on future emails using the predictive model, with a known margin of error.

Some functions that can generate these models include Naive Bayes, K-means nearest neighbor, decision trees, and logistic regression. The model generated by this type of function does not generally follow a relational structure. It is more likely to be in a JSON format. It can be stored in the file system, but it is more commonly stored in a database.

So in the machine learning scenario, the data inputs to the SQL-MR function will consist of 1) a model, usually in JSON format, and 2) one or more sets of data to be analyzed against, using the model. Similar to the lookup table example, the predictive model must be applied to each row of input from the new data set. Thus, the predictive model will be input to the function using the DIMENSION keyword. The data to be analyzed could use either PARTITION BY ANY or PARTITION BY *partitioning\_attributes*.

## SQL-MR Multiple Input FAQ

### How are multiple inputs combined?

Multiple inputs are combined using what is essentially a cogroup operation, with the addition of support for dimensional inputs. Grouping is done using an OUTER JOIN where NULLs compare equal. The SQL-MR function is effectively invoked once for each unique partition of all the partitioned inputs. All dimensional inputs are provided at each invocation. The function can output one or more tuples at each invocation.

There are two mutually exclusive cases to consider in determining what the unique partitions of the partitioned inputs will be:

- One or more `partition_attributes_input` inputs are combined into partitions using a cogroup operation. The cogroup operation forms one partition for each unique combination of partitioning attributes present in any of the inputs. Each partition provides the values of the partitioning attributes and the tuples from each input that agree on those values. If a given input has no tuples for a particular combination of partitioning attributes, then an empty set of tuples is provided for that input.
- A single `partition_any_input` is processed wherever its data is stored. Each invocation provides the input tuples to the vworker where they currently reside in Aster Database.

Note that the function is not invoked if all of the inputs tables to partitioned inputs are empty, even if a dimensional input has been provided. Thus the dimensional inputs are not first class inputs in that they do not drive invocation of the function. They are simply provided as additional input to the function.

### **Can dimensional inputs include non-deterministic expressions?**

No. Dimensional inputs should not include non-deterministic expressions. These are expressions that are not guaranteed to evaluate to the same result every time (expressions with a volatility other than IMMUTABLE). Note that whenever changing Global User Configuration (GUC) settings can change the result of an expression, the volatility of that expression may be classified as STABLE, but it cannot be classified as IMMUTABLE. An example of this would be changing Locale and Formatting settings, such as datestyle or time zone.

### **Where will the output of the function be located?**

The output location for row and cogroup functions is determined by the function:

- If it is vworker specific, the output goes to that worker.
- If it is partitioned, the output will be partitioned in the same way.
- If it is replicated, the output will be replicated.

If the input is located on the queen:

- If the inputs are row or dimension inputs, the output will be replicated to the vworkers.
- If the input is partitioned, the output will also be partitioned.

## **SQL-MapReduce Java API**

### **Prerequisites for Working with SQL-MapReduce in Java**

The features presented in this document are realized via extensions to the SQL language and thus require little or no semantic change for existing users of Aster Database.

In order to develop SQL-MapReduce functions, you need:

- 1 The Aster Database SQL-MapReduce SDK Java package,  
`com.asterdata.ncluster.sqlmr`, which you can find in the `sqlmr-sdk-`

`java.tar.gz` archive. The examples in this section are based on sample code in the bundle.

- 2 The Aster Database ACT client – which has support for installing procedural code.
- 3 Java JDK 6, Update 39 or newer – for compiling Java functions on your workstation.
- 4 Python 2.5 – for developing procedural code written in Python.
- 5 HTTP access to the queen node and the AMC – for viewing and managing SQL-MapReduce queries with the Aster Database AMC, including queries that invoke procedures. To check SQL-MapReduce query execution, see “[Manage SQL-MapReduce Execution](#)” on page 114.



**Note!** SQL-MR functions written with the Aster Database SQL-MapReduce Java SDK version 4.x can still be run on Aster Database 5.0 without recompiling. However, functions written in the Aster Database SQL-MapReduce SDK version 5.0 and higher cannot be run in pre-5.0 versions of Aster Database.

## Write a SQL-MapReduce Function in Java

This section guides you through writing and invoking a simple SQL-MapReduce function. It assumes you have downloaded the Aster Database SQL-MapReduce SDK.

To write an SQL-MapReduce function in Java, you create a Java class that implements one of the following interfaces:

```
com.asterdata.ncluster.sqlmr.RowFunction  
com.asterdata.ncluster.sqlmr.PartitionFunction  
com.asterdata.ncluster.sqlmr.MultipleInputFunction
```

The class must implement a public constructor that takes a  
`com.asterdata.ncluster.sqlmr.RuntimeContract`.

The name of your SQL-MapReduce function is the name of the Java class, ignoring case differences. So, for example, a function named *splitintowords* might be implemented by a Java class `com.mycompany.SplitIntoWords`.

Teradata Aster’s SQL-MapReduce framework supports three types of functions. A SQL-MapReduce function must implement one of these three interfaces (either RowFunction, PartitionFunction, or MultipleInputFunction).

- 1 RowFunction. A RowFunction corresponds to a “map” function with a single input, and must be invoked without a PARTITION BY clause. From an interface perspective, the function will be passed an iterator to an arbitrary set of rows. A RowFunction consists of two functions: the constructor and the operate function, `operateOnSomeRows()`.
- 2 PartitionFunction. A PartitionFunction corresponds to a “reduce” function with a single input, and must be invoked with a PARTITION BY clause. Rows with the same values for the PARTITION BY expressions are brought together onto the same logical worker, and each invocation of the function is passed all the rows in that partition. A partition function consists of two methods: the constructor, and the operate function, `operateOnPartition()`.

- 3 **MultipleInputFunction.** A `MultipleInputFunction` is a new row-producing interface, provided for functions that require multiple inputs. It implements a row emitting method `operateOnMultipleInputs()` that is provided one or more partitioned inputs and zero or more dimensional inputs per invocation.

All of these function types are described below.

## Constructor for a SQL-MapReduce Function in Java

Each function must implement a constructor which takes only a `RuntimeContract`. The function class is instantiated during planning of the query, and it informs the system about its properties via the `RuntimeContract`. The system fills in various fields in the contract (such as the input schema and the argument clauses) and passes this incomplete contract to the constructor. The constructor must fill in the function's output schema and then complete the contract. Below is an example of a constructor for a simple SQL-MapReduce function that always returns (word varchar, count int).

```
public tokenize(RuntimeContract contract)
{
    ArrayList<ColumnDefinition> outputColumns = new ArrayList<ColumnDefinition>();
    outputColumns.add( new ColumnDefinition( "word", SqlType.varchar() ) );
    outputColumns.add( new ColumnDefinition( "count", SqlType.int() ) );
    contract.setOutputInfo( new OutputInfo(outputColumns) );
    contract.complete();
}
```

## Operate Function: `operateOnSomeRows()` or `operateOnPartition()`

A `RowFunction` must implement `operateOnSomeRows()`, while a `PartitionFunction` must implement `operateOnPartition()`. The only difference between the two is that `operateOnPartition()` includes an argument that describes the current partition. Both these functions are given a `RowIterator`, which allows iteration through all the rows that the function sees. Moreover, they are provided with a `RowEmitter`, which allows the function to return rows to the database.

Below is an example of a simple function that tokenizes its input into rows. This example and the rest of the examples in this section are part of the Aster Database SQL-MapReduce SDK bundle.

### Sample tokenize function:

```
public void operateOnSomeRows(
        RowIterator inputIterator,
        RowEmitter outputEmitter
)
{
    while ( inputIterator.advanceToNextRow() )
    {
        String[] parts = splitPattern_.split( inputIterator.getStringAt(0) );
        for (String part : parts)
        {
```

```
        outputEmitter.addString(part);
        outputEmitter.addInt(1);
        outputEmitter.emitRow();
    }
}
```

See the `examples/` directory in the SQL-MapReduce SDK for sample SQL-MapReduce functions, as well as a Makefile to build and package them. For reference information on the SQL-MapReduce API, see the Javadoc reference in the SDK bundle.

## Argument Clauses for Java Functions

Often it's useful to let the calling SQL query pass a runtime argument to the SQL-MapReduce function in order to modify the behavior of the function. To support this, the API provides the `useArgumentClause` method for declaring argument clauses. An argument clause can contain multiple values, and a function can accept multiple argument clauses. See “[SQL-MapReduce Query Syntax](#)” on page 85 for an explanation of how the SQL user passes multiple clauses and values.



**Note!** Don't confuse argument clauses with input data. *Input data*, provided in the ON clause, provides the data the function operates on, while an *argument clause* typically sets an operating parameter the analyst has chosen to use in this running of the function.

You declare argument clauses when you declare your `RuntimeContract`, as shown here:

```
public final class MyFunction implements RowFunction
{
    public MyFunction(RuntimeContract contract)
    {
        String mySingleValue = contract.useArgumentClause("mysingle").getSingleValue();
        List<String> myMultipleValues =
            contract.useArgumentClause("mymultiple").getValues();
        // ...
    }
    // ...
}
```

After you've defined the argument clauses shown in the example above, the user of your SQL-MapReduce function can pass arguments in his SQL query like this:

```
SELECT ...
FROM myfunction(
    ON mytable
    MYSINGLE ('some value')
    MYMULTIPLE ('a value', 'another value')
);
```

See “[SQL-MapReduce Query Syntax](#)” on page 85 to see this in the context of other select clauses.



**Notice!** You cannot use parameter markers (“?”) in SQL-MapReduce invocations. (A parameter marker, as explained in the Microsoft SQL Server documentation is “a question mark (?) placed in the location of an input or output expression” in a SQL statement.)

For example, you cannot write:

```
SELECT *
FROM GetUIDropDown(
ON (SELECT 1)
    producttag(?)
    maxdropdownsize('500')
    clustersonly('0')
);
```

**Workaround:** SSRS allows the use of report parameters. In SSRS, set the parameter so that it does not allow blank values (that is, uncheck the “Allow blank values” checkbox). This forces the sending of a string, which allows the SQL-MapReduce function to run with unnamed parameters.

## Test Java Functions Locally with TestRunner

For information on how to test Java functions locally, see the *Aster Database Development Environment User Guide*.

## Build and Package the SQL-MapReduce Function

To prepare a Java function for use in SQL-MapReduce, compile it into a class file (build it on your workstation) and install it in Aster Database using the ACT client’s \install command. Be sure to also install any other classes that your function depends on, as we’ll explain below.

The examples directory provided in SQL-MapReduce SDK bundle (ncluster-sqlmr-sdk.zip) contains a Makefile that builds and packages the Aster Database example functions when you run ‘make’.

You must package your SQL-MapReduce functions in one of the following ways:

- as a single .class file for the function,
- as a .jar file containing the function class and other classes, or
- as a .zip file containing the function’s .jar and other (possibly third-party) .jar files.

The name of the file must match the name of the function. For example, a function named *foo* must be compiled as *Foo.class*, and you can install the class on its own or as part of *foo.jar*, or, finally, as part of *foo.zip* which contains *foo.jar* with its *Foo.class*.

The advantage of a .zip-packaged function is that you can include *external* .jar files unmodified. For example, you might write a .zip-packaged function with the following structure:

Figure 9: Example structure of an SQL-MapReduce function bundle to be installed in Aster Database



In this example, the primary SQL-MapReduce function we care about is “foo,” so we package it in an archive called “foo.zip,” which in turn must contain foo.jar, which in turn must somewhere contain a file Foo.class (path and case is not significant). By using a .zip file, we can contain other classes that our foo function relies on. In this case, the package also contains another class in foo.jar (OtherClass.class) and also another, third-party .jar file (otherjar.jar) with a class that foo relies on, called “Useful.class.”

Once you have packaged your functions, install them in your database as shown in [“Install and Use a Sample Function” on page 111](#).

## SQL-MapReduce Examples in Java

Aster Database includes sample source code that provides examples of SQL-MapReduce functions that developers can create for use in Teradata Aster’s In-Database MapReduce framework. While these are not Aster Database-supported functions, these samples are useful for practice to accelerate development of custom SQL-MapReduce functions that can be used to solve a customer’s most pressing analytical questions.

### **SQL-MapReduce Example 1: Word Count**

Consider a hypothetical SQL-MapReduce function that splits strings into individual words:

```

SELECT word
FROM SplitIntoWords (
    ON documents
)
  
```

In this example, the `SplitIntoWords` function is invoked once for every row in the `documents` table. It is Java procedural code that takes each document and emits a row for each word. The function itself defines the columns that appear in its output rows; in this case, `SplitIntoWords` emits rows with a single column named `word`.

We might want to use such a function to compute the 10 most-frequently occurring words in a body of text. One approach would be to write another SQL-MapReduce function that counts the number of times a given word appears. We might have a `CountInput` function for this purpose:

```

SELECT word, count AS frequency
FROM CountInput (
  
```

```
    ON SplitIntoWords( ON documents )
    PARTITION BY word
    )
ORDER BY frequency DESC
LIMIT 10
```

In this example, the rows that are output by `SplitIntoWords` are formed into groups of distinct words using the `PARTITION BY` clause. Then, the `CountInput` function counts the number of words in each partition.

Of course, we don't need a special `CountInput` function at all! We can just use normal SQL:

```
SELECT word, COUNT(*) as frequency
FROM SplitIntoWords(
    ON documents
)
GROUP BY word
ORDER BY frequency DESC
LIMIT 10
```

This is a simple example of the flexibility provided by the integration of SQL and MapReduce-style computation in SQL-MapReduce functions.

## SQL-MapReduce Example 2: Sessionization

Sessionization is the process of mapping each click in a clickstream to a unique session identifier. We define a session as a sequence of clicks by a particular user where no more than n seconds pass between successive clicks (that is, if we don't see a click from a user for n seconds, we start a new session). Sessionization can be easily done with the `Sessionize` SQL-MapReduce function. Sample code is included with the Aster Database SQL-MapReduce Java API.

We can sessionize a table called `clickstream`, which consists of a `userid` and `xtimestamp` attribute, like this:

```
SELECT xtimestamp, userid, sessionid
FROM   Sessionize  (
    ON      clickstream
    PARTITION BY userid
    ORDER BY    xtimestamp
    TIMECOLUMN ('xtimestamp')
    TIMEOUT     (60)
) ;
```

The first parameter to the `Sessionize` SQL-MapReduce function is the name of the `xtimestamp` attribute, while the second is n, the number of seconds between clicks that results in the starting of a new session. The `clickstream` table is partitioned by `userid`, and within each partition, rows are sequenced by `xtimestamp`. The `sessionize` SQL-MapReduce function is then invoked against each of these ordered partitions, emitting the input rows with the appropriate `sessionid` added.

# SQL-MapReduce C API

Aster Database allows you to write custom functions in C that you install and run on the cluster. SQL-MapReduce functions written in C are invoked using a SQL SELECT, just like SQL-MapReduce functions written in Java.

Aster Database provides a C SDK for developing and testing your C functions locally on your development workstation, before you install and run them on the cluster. To deploy an SQL-MR function, you write the function, compile it into a shared library (.so file), test it, and use the ACT \install command to install it on the cluster. The function is usable by all Aster Database SQL users.

This section lists the resources provided in the SDK and explains how to write and test SQL-MapReduce functions in C.



**Note** SQL-MR functions written with the Aster Database SQL-MapReduce C SDK version 4.x can still be run on Aster Database 5.0 without recompiling. However, functions written in the Aster Database SQL-MapReduce SDK version 5.0 and higher cannot be run in pre-5.0 versions of Aster Database.

## Types of C Functions

Teradata Aster's SQL-MapReduce C API supports three types of functions:

- 1 Row Function. A Row Function corresponds to a “map” function and must be invoked without a PARTITION BY clause. From an interface perspective, the function will be passed an iterator to an arbitrary set of rows. A Row Function consists of two methods: the *contract method*, with a name like echo\_input\_newRowFunction (where “echo\_input” is the name of our example row function), and the *operate method* with a name like, for example, echo\_input\_operateOnSomeRows.
- 2 Partition Function. A Partition Function corresponds to a “reduce” function and must be invoked with a PARTITION BY clause. Rows with the same values for the PARTITION BY expressions are brought together onto the same logical worker, and each invocation of the function is passed all the rows in that partition. A Partition Function consists of two methods: the *contract method*, with a name like geog\_filter\_newPartitionFunction (using “geog\_filter” as an example function name), and the *operate method* with a name like, for example, geog\_filter\_operateOnPartition.
- 3 Multiple Input Function. A Multiple Input Function takes one or more partitioned inputs and zero or more dimensional inputs. As such, it is invoked with one or more PARTITION BY clauses (for the partitioned inputs) and zero or more DIMENSION clauses (for the dimensional inputs). A Multiple Input Function consists of two methods: the *contract method*, with a name like lookup\_filter\_newMultipleInputFunction (using “lookup\_filter” as an example function name), and the *operate method* with a name like, for example, lookup\_filter\_operateOnMultipleInputs.

## Get and Unpack the SQL-MapReduce C SDK

In these instructions, we'll be working in an example directory called `~/dev/stage/sqlmr-sdk`.

- 1 Obtain the C SDK bundle, `sqlmr-sdk-c.tar.gz`.
- 2 Unpack the C SDK bundle.

```
cd ~/dev
tar xf sqlmr-sdk-c.tar.gz
```

Change the working directory to `~/dev/stage/sqlmr-sdk`. You will see these directories:

- `include/sqlmr/api/c` contains the SQL-MapReduce C API headers, such as `ArgumentClause.h`, `ByteArray.h`, `ByteStream.h`, `ColumnDefinition.h`, `Core.h`, and so on. Comments in the header files explain the methods you will implement when developing a row or partition function, and they discuss memory ownership, error conditions, and so on.
- `example/c` contains some sample C API functions (source code and makefiles) such as `echo_input` and `list_files`. You will also find source code here, for example function tests that use TestRunner for local testing.
- `include/sqlmr/testrunner/c` contains the header files of the TestRunner testing framework for testing API functions locally. Comments in the headers explain the tools provided for testing.
- `lib` contains the TestRunner library.

## Build the C API Examples and Tests

A makefile is provided to build the examples. To build the examples, do this:

- 1 Ensure your development workstation conforms to the guidelines stated in “[Build Tools and Dependencies](#)”, below.
- 2 Find the example you wish to build. Here, we'll show how to build `echo_input.c`. Change the working directory to the `echo_input` example directory. Here, we'll assume the path is:

```
cd ~/dev/stage/sqlmr-sdk/example/c/echo_input
```

- 3 Build the example. Type “make” followed by the setting of the `SQLMR_SDK` environment variable to your SQL-MapReduce SDK directory. In our example set-up, this looks like:

```
# make SQLMR_SDK=~/dev/stage/sqlmr-sdk
```

This builds your function (`echo_input.so`, in this case) and places it in the `build` directory. It also creates the `build` directory if needed. At this point, you could install and use the function in Aster Database if you like, but it's a good idea to test your code locally, first.

- 4 Build and run the test for the example:

```
make SQLMR_SDK=~/dev/stage/sqlmr-sdk test
```

This builds the function, builds its corresponding test program, and runs the test. You will see messages at the command line indicating the success or failure of each stage. For example, a successful test run shows lines similar to:

```
Loaded 'sqlmr_functionModuleInfo' symbol from build/echo_input.so: 0x2b2158d2ce30
Loaded function entries for module 'build/echo_input.so' (with matching API version 3)
Function completed operating without error
Task reported as completed
```

- 5 Inspect the test output. Change the working directory to the build/testoutput directory (the complete path will be like ~/dev/stage/sqlmr-sdk/example/c/echo\_input/build/testoutput) and view:
  - testrun.out, the *output data file* that contains the actual output the function produced based on the test input data.
  - contract.out, the *completed runtime contract file* that provides a list of the output columns of the SQL-MapReduce function, and the datatype of each
- As with most makefiles, you can type “`make clean`” to delete all built files and test output, so that you can revise your code and test again.
- 6 If you like, you can install the .so file on the cluster and use your function there. See “[Install and Use a Sample Function](#)” on page 111.

## Build C API Functions

### Build Tools and Dependencies

Before you begin SQL-MapReduce C API development, make sure your development machine meets the following requirements:

- 1 The operating system must be one of the supported, 64-bit versions of Linux listed in the *Aster Database 5.0 Server Platform Guide*.
- 2 You must have gcc for compiling your C functions.
- 3 If your SQL-MapReduce C function uses libraries other than the C standard libraries, you must statically link to those libraries when you build the function.

### Build and Deploy SQL-MapReduce C API Executables

To prepare a C or C++ function for use in SQL-MapReduce, compile it into a “.so” executable file (build it on your workstation) and install it in Aster Database using the ACT client’s \install command.

#### *Compiling*

To compile your function, use gcc with the following flags and environment settings:

- Required: The `SQLMR_SDK` environment variable must be set to the path of your SQL-MapReduce C API directory. This is the directory that includes `example`, `include` and `lib` directories. For example, if you unpacked the API to `~/dev/stage` on your machine, then the setting is “`SQLMR_SDK=~/dev/stage/sqlmr-sdk`”. We recommend that you make this setting part of your standard environment settings.
- Required: Use the `-fPIC` flag to ensure your compiled function code is relocatable.

- Optional: For help in debugging, use the `-Wall` flag to show all compiler warnings.
- Optional: To create a debuggable executable, use the `-g` flag.

For example, to compile the `echo_input.c` example, you type:

```
gcc -fPIC -c echo_input.c
```

### ***Linking***

When you link your libraries, use the `-shared` flag to create a shared library (`.so`). For example, to link the `echo_input.c` example, you type:

```
gcc -shared -o echo_input.so echo_input.o
```

Remember! If your function depends on third-party or other libraries, you must statically link them.

To see examples of the linking flags in use, review the Makefiles provided in the API directory, `sqlmr-sdk/example`.

## **Prepare C API Executables for Installation**

Follow these rules with respect to packaging your function executables:

- 1 The file must be a “`.so`” file compiled and linked as described above. The executable cannot be packaged in a `.zip` archive or any other type of archive, and you cannot install a “`.c`” source code file.
- 2 The name of the file must match the name of the function. For example, a function named “`foo`” must be compiled as `foo.so` and *not* as `libfoo.so` or any other name.

Once you have built your functions, install them in Aster Database using the ACT client’s `\install` command as shown in “[Install and Use a Sample Function](#)” on page 111.

## **Write an SQL-MapReduce Function in C**

You define a C-based SQL-MapReduce function by writing a function that satisfies the following requirements and uses the following API resources. (For examples, see the `sqlmr-sdk/example` directory.)

- 1 Headers to include: Import the needed SQL-MapReduce headers from `sqlmr/api/c/` (in the `sqlmr-sdk/include` directory). Most SQL-MapReduce C functions will include `Core.h`, `FunctionModule.h`, `NativeValue.h`, `RowBuilder.h`, `RowIterator.h`, `RowView.h` and `RuntimeContract.h`. Because it is a row function, the `echo_input` example must include `RowFunction.h`.
- 2 Making the function known to Aster Database: In an `SQLMR_FUNCTION_MODULE_BEGIN` / `END` block, declare the function as a `newRowFunction` or `newPartitionFunction`. This looks similar to

```
SQLMR_FUNCTION_MODULE_BEGIN()
{
    SQLMR_FUNCTION_ENTRY("echo_input")
    {
        entry->newRowFunction = &echo_input_newRowFunction;
    }
}
```

```
SQLMR_FUNCTION_MODULE_END()
```

- 3 Row/partition implementation requirements: Implement your function by doing one of the following:
  - If the function is a *row function*, write the prototype and implementation of the `<my_function_name>_newRowFunction()` method and the `<my_function_name>_operateOnSomeRows()` method. The row function must complete the runtime contract, as explained in `RuntimeContract.h`.
  - If the function is a *partition function*, write the prototype and implementation of the `<my_function_name>_newPartitionFunction()` method and the `<my_function_name>_operateOnPartition()` method. The partition function must complete the runtime contract, as explained in `RuntimeContract.h`.
- 4 Handling arguments: If your SQL-MapReduce function will take arguments in the SQL command, use the facilities provided in `ArgumentClause.h` (`SqlmrArgumentClauseH`) to add argument clauses to your function. For an example of this, see `sqlmr-sdk/example/c/repeat_input/repeat_input.c`.
- 5 Helper files: Aster Database allows you to install files on the cluster to act as SQL-MapReduce helper files or to hold data that you do not wish to store in the database. If your SQL-MapReduce function will use or operate on an installed file, use the functions of `InstalledFile.h` (like `sqlmr_if_getInstalledFiles` and `sqlmr_if_openForRead`). For an example of this, see `sqlmr-sdk/example/c/list_files/list_files.c`.
- 6 Naming conventions for API methods: The user-facing methods provided by the SQL-MapReduce C API have names starting with “`sqlmr_`”. Often, there’s also a two-letter code in the name indicating which module provides the function. These include:
  - `sqlmr_rc_*` for functions provided by `RuntimeContract`
  - `sqlmr_ri_*` for functions provided by `RowIterator`
  - `sqlmr_rb_*` for functions provided by `RowBuilder`
  - `sqlmr_rv_*` for functions provided by `RowView`
- 7 Naming conventions for datatypes: Datatype names in the SQL-MapReduce C API follow these rules:
  - All types in the API start with `Sqlmr*`
  - Some types also end with `*H` (like, for example, `SqlmrRowViewH`). The “H” stands for “handle,” meaning that such types are actually pointers to some opaque type.
  - By contrast, types that do *not* end with `*H` (like, for example, `SqlmrNativeValue`) are value types, and are structs with a non-opaque representation.
- 8 Error handling: Handle errors and return error information using the methods of `sqlmr-sdk/include/sqlmr/api/c/core/Error.h`.
- 9 Memory Management: See “[Memory Management in C API Functions](#)”, below.

## Memory Management in C API Functions

### Memory Management Utility Methods in the C API

To allocate and release memory in your C API functions, the API provides utility methods, such as the `sqlmr_malloc`, `sqlmr_release` and other methods of `Memory.h`. In addition, many of the datatype objects such as `String` provide more specialized memory utility functions. See “[Datatypes in the C API](#)” on page 109.

### Memory Ownership in C API Functions

Managing memory correctly is an important part of programming against the SQL-MapReduce API in C and C++. One central concept of memory management is *ownership*. In general, if your code owns some memory, that memory is your responsibility, and you must release it appropriately (sometimes with the `sqlmr_release` function and sometimes with a type-specific `sqlmr_*_releaseOwned` function, as described in the API reference documentation). For memory that your code does *not* own, the SQL-MapReduce framework will release the memory, so your code must not release it or modify it.

Functions in the SQL-MapReduce API never take ownership of data given to them, but they may transfer ownership to the caller. The reference documentation for the SQL-MapReduce API describes the rules for memory ownership in detail on a function-by-function basis.



**Notice!** Memory must be released with the appropriate function. Do not use the C library `free` function or the C++ `delete` operator to release objects that have been provided by the SQL-MapReduce framework. (Of course, if you have allocated some memory yourself, then you must free that memory as usual when it is no longer needed.)

## Datatypes in the C API

Datatype objects in the SQL-MapReduce C API provide utility methods for memory allocation and operations on data objects. The datatypes have the following common set of utility functions, where “\*” is replaced with the name of the datatype:

- `sqlmr_*_newFromOther`
- `sqlmr_*_newFromCstring`
- `sqlmr_*_releaseOwned`
- `sqlmr_*_allocCstring`
- `sqlmr_*_printToCstring`
- `sqlmr_*_cmp`
- `sqlmr_*_equals`

In addition to the common methods listed above, some datatypes provide more specialized functions. For example, the `Sq1mrDate` datatype provides the utility functions `sqlmr_date_newFromMDY` and `sqlmr_date_toMDY` for working with month, day, and year values directly.

For details, read the individual header file comments. For example, see `String.h` for information about `sqlmr_string_newFromOther`, `sqlmr_string_newFromCstring`, `sqlmr_string_releaseOwned`, and so on.

## Test C API Functions Locally with TestRunner

Because SQL-MapReduce functions are built to run in Aster Database, on data in Aster Database, testing them locally requires a test program that can simulate the data and services of the cluster. The SDK provides this in the form of the TestRunner (the actual library path and name is `sqlmr-sdk/lib/libsqlmr-testrunner.so`).

Detailed reference documentation for TestRunner is provided in the header files in `sqlmr-sdk/include/sqlmr/testrunner/c/SimpleTest.h` and in the supporting headers `TestArgumentClause.h` and `TestColumnDefinition.h`.

TestRunner allows you to write a test that:

- Runs an SQL-MapReduce function contained in a specified SQL-MapReduce C executable file (an .so library).
- Provides test input data to the function, using a data input file you specify.
- Simulates other SQL-MapReduce services, such as installed files or temporary storage.
- Writes the test output to files you specify:

The *completed runtime contract file* provides a list of the output columns of your SQL-MapReduce function, and the datatype of each. For example, if you have built the “test” target in the `echo_input` example, you will see `sqlmr-sdk/example/c/echo_input/build/testoutput/contract.out`

The *output data file* contains the actual output your function produced based on the test input data. For example, (assuming you have built the “test” target), you will see, `sqlmr-sdk/example/c/echo_input/build/testoutput/testrun.out`

After running the function, the test can then verify the contents of the created output files (for the completed contract and output data).



Note! Only row functions are supported for the C test runner. Thus it is not possible to test partition or multiple input functions using this test runner.

## Write the Test Code

For example code that shows how to write a test program, read the samples such as `testrunner_echo_input.c` that are provided in the `sqlmr-sdk/example/c` directory of the SDK. In a nutshell, when writing your test program for the TestRunner framework, you can do the following:

- Use the `SqlmrSimpleTestParams` fields `functionFile` and `functionName` to store the path and name of the executable (.so file) to be tested.
- Use the column utilities of `SqlmrSimpleTestParams` to build a data input table, which is an array of `SqlmrTestColumnDefinition` objects. This simulates an Aster Database table.
- Specify the name of the input file that will fill the simulated table with data, as well as sources for any arguments the function takes. The test data in the input file is assumed to be tab-delimited. For an example input file, see `sqlmr-sdk/example/c/echo_input/testrun.in`.

- If your SQL-MapReduce function uses installed files (for example to feed additional data to the function) use the `SqlmrSimpleTestParams` field `installedFileDirectoryOrNull` to specify the location of the simulated installed file. This allows TestRunner to simulate the installed files feature of Aster Database. (See “[Manage Functions and Files in Aster Database](#)” on page 140.)
- Specify where the test results will be saved (using the `completedContractFileName` field for the completed SQL-MapReduce runtime contract results and the `outputFileName` field for the function’s output).
- Run the test using the `sqlmr_runSimpleTest` function.

It can be convenient to mimic the naming conventions of the examples. To do this, name your test program “`testrun-<function_name>.c`,” as in “`testrun-echo_input.c`.”

### Build and Run Tests

To run the test, build it as demonstrated in the sample Makefiles (for example, see the “test” target in `sqlmr-sdk/example/c/echo_input/Makefile`) and then run the test executable. You must add the SQL-MapReduce SDK `lib` directory to your `LD_LIBRARY_PATH` in order to run the test. For example:

```
LD_LIBRARY_PATH=~/.dev/code/stage/sqlmr-sdk/lib:$LD_LIBRARY_PATH
```

## Install and Use a Sample Function

This section shows you how to install your SQL-MapReduce function and how to invoke it in an SQL query. (For a complete explanation of how to install and manage functions, see instead “[Manage Functions and Files in Aster Database](#)” on page 140.)

### Prerequisites

- Make sure you have followed the instructions in “[Build and Package the SQL-MapReduce Function](#)” on page 101 or “[Prepare C API Executables for Installation](#)” on page 107.
- Make sure the SQL user account *that you will use to install the function* has the `INSTALL FILE`, `CREATE FUNCTION`, and `EXECUTE` privileges in the schema where you will install the function. See “[User Permissions for Installed Files and Functions](#)” on page 139.
- Make sure your file is not too large to install. There is a limit of 238MB on the size of the file to be installed. If you try to install a larger file, you will see an error like:

```
ERROR: row text exceeds limit of 238MB ...
```

Note that when installing larger files, the queen may run out of memory. The queen needs available memory of approximately eight times the size of the file to be installed, in order to encode, buffer, and copy the file.

## Procedure

- 1 Connect to Aster Database with the Aster Database ACT client. Use a user account that has rights to install functions in the schema and rights to grant permissions to those functions. In this example we'll use a schema called, *textanalysis*.

```
$ act -h <queen-ip> -d <databasename> -U <username> -w <password>
```

- 2 Example only: For this example, we create some test data. At the ACT SQL command line, we type:

```
BEGIN;
CREATE FACT TABLE documents (document varchar)
    DISTRIBUTE BY HASH(document) ;
INSERT INTO documents VALUES
    ('this is a test document'),
    ('this is another test document') ;
END;
```

- 3 Install the SQL-MapReduce function using ACT's \install command. (For a list of such commands, see [“Manage Functions and Files in Aster Database” on page 140](#).)

Here, we assume:

- the files counttokens.jar and tokenize.jar are local to the directory where you invoked ACT,
- you are working in a schema called “textanalysis” (*Tip*: If you don't have a schema set up for your use, try installing in the schema, “public”, instead. Just replace “textanalysis” with “public” below.), and
- you have INSTALL FILE and CREATE FUNCTION privileges in the *textanalysis* schema where you will install the functions.

```
\install counttokens.jar textanalysis/counttokens.jar
\install tokenize.jar textanalysis/tokenize.jar
```

Optional: If you like, you can enclose the \install command and subsequent calls to it in a transaction so that you can roll it back later to remove the SQL-MapReduce function from your cluster. (In this example we do an ABORT to remove our test function from the system.)

Optional usage:

```
BEGIN;
\install counttokens.jar textanalysis/counttokens.jar
\install tokenize.jar textanalysis/tokenize.jar
SELECT ... -- queries that use the installed functions
ABORT;
```

*Tip*: You can always type “\dF+” or “\dF \*.\*” to check which schema a function belongs to. When you call the function, Teradata Aster recommends that you include its schema name. Type \dF+ <function name> to check which schema the function belongs to.

- 4 Use the GRANT command to give the EXECUTE privilege to users who will run the function. For this example, let's assume we want user “beehive” to be able to run the function.

Note that in most ACT commands for managing functions, when you type the function name, *you do not type its suffix* (like “.jar” in this example). Thus the syntax is:

```
GRANT EXECUTE
    ON FUNCTION textanalysis.counttokens
    TO mjones;
```

Repeat the above step for the rest of your users and functions. Alternatively, you can grant EXECUTE rights at the group level by replacing the user name with a group name in the GRANT EXECUTE statement.

## Run the Function

Now the function is installed and usable by all users to whom you've granted EXECUTE rights. Test your function by following the steps below to run it:

- 1 Run Aster Database ACT and log in as an SQL user who has the EXECUTE privilege on the function.
- 2 Invoke the function in a statement such as a SELECT or other data-retrieval statement. Make sure you schema-qualify the function's name, or have its schema in your schema search path.

This example nests a call to the function `tokenize` inside a call to the function `counttokens`:

```
SELECT token, count
FROM textanalysis.counttokens
(
    ON (SELECT token, count
        FROM textanalysis.tokenize
        (
            ON documents)
    )
    PARTITION BY token
)
;
```

*Note!* Here, the function name is an all-lowercase name, so we did not have to surround the name in double quotes. If your function name contains uppercase letters, your SELECT statement must enclose it in double quotes.

- 3 Invoke another SQL-MapReduce function. We add this example to show that the following code is equivalent to the example you just typed:

```
SELECT token,
       sum(count)
FROM textanalysis.tokenize
(
    ON documents)
GROUP BY token
ORDER BY sum(count)
;
```

See “[Manage Functions and Files in Aster Database](#)” on page 140 for more information.

# Manage SQL-MapReduce Execution

Aster Database monitors the execution of the SQL-MapReduce function and provides statistics in the Processes tab of the AMC (Aster Database Management Console). Running functions and recently run functions are highlighted in green or grey on the left side of the screen. In the Processes tab, in the Processes list, click the function to open it for inspection in a new tab.

Figure 10: Examining a running SQL-MapReduce function in the AMC

*Note:* Since the code of an SQL-MapReduce function is foreign to Aster Database, this code is executed in a sandboxed environment.

## Start an SQL-MapReduce Job

An SQL-MapReduce job is automatically started when you issue a query that includes an SQL-MapReduce function.

## Cancel an SQL-MapReduce Job

You can stop a running SQL-MapReduce job by canceling its SQL query. You cancel a query by typing `Ctrl+C` in ACT while the query is running, or in the AMC, as explained here:

To cancel the job in the Aster Database Management Console (“AMC”):

- 1 Open the AMC in a browser window by typing *http:<IP address of the queen>*
  - 2 Go to the Processes tab.
  - 3 Find your running query in the Processes list. To do this, it may be helpful to sort based on status. Click the Status column *twice* to bring the running queries to the top of the list.

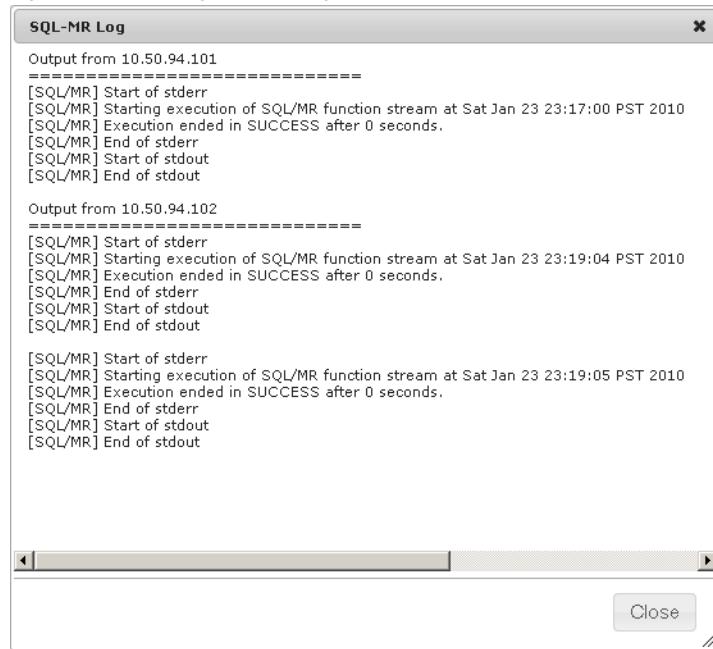
- 4 There are two ways to cancel the query.
  - a In the far right column of the Processes list, click the blue Cancel icon of the query you wish to cancel. If the Cancel icon does not appear, the process cannot be canceled at this time.
  - b Click the ID of the query you wish to cancel. In the Process Detail tab that appears, click the Cancel Process button.

## Debug SQL-MapReduce Job and Task Execution

Your SQL-MapReduce functions can emit debugging messages written to the standard output or the standard error. During or after execution, access to the standard output (stdout) and standard error (stderr) is provided through the AMC. To see this:

- 1 Open the AMC in a browser window by typing *http:<IP address of the queen>*
- 2 Go to the Processes tab.
- 3 Find your query in the Processes list. To do this, it may be helpful to sort based on Type or User. Click a column to sort based on that column.
- 4 Click the ID of the query you wish to view. In the Process Detail tab that appears, click the View Logs button.

Figure 11: Checking the error logs after an SQL-MapReduce function has run



## Troubleshooting and SQL-MR Errors

### Size Limit on Constant String

If you provide a constant string without a datatype in the ON clause of a SQL-MR function, you will see an error message like:

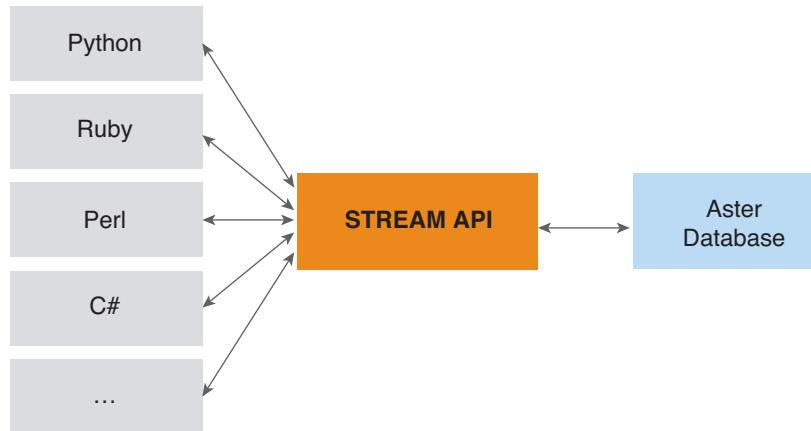
```
ERROR: row is too big: size 34080, maximum size 32736
```

You can avoid this error by doing one of these workarounds:

- Cast the content string to the VARCHAR datatype.
- Put the constant string into a table, and use a SELECT statement to retrieve it.

# Stream API for Python, Perl, and Other Types of Scripts

The Aster Database Stream API (STREAM) allows you to run scripts and functions written in various languages including Python, Ruby, Perl, and C# (the last is supported using the Mono runtime).



The Stream API works much like the SQL-MapReduce APIs, but does not provide the built-in datatype handling that SQL-MapReduce provides.

The Stream API is useful when you have existing scripts you want to run in Aster Database, or if you prefer to write your functions in a language other than those supported in SQL-MapReduce, such Java, C, and C++.

## Stream Function Usage

### Syntax

```
SELECT ... FROM STREAM (ON input_relation SCRIPT 'script_name') ;  
SELECT ...  
FROM STREAM  
    (ON {table_name | view_name | (query) }  
     [PARTITION BY expression [, ...]]  
     [ORDER BY expression [ASC | DESC] [, ...]]  
     [SCRIPT ('[runtime_name] scriptname')]  
     [MEM_LIMIT_MB ('int' | 'unlimited')]  
     [OUTPUTS ('column_name column_type' [, ...])]  
     [NULLSTRING ('null_string')]  
     [DELIMITER ('delimiter_character')])  
    ) [ [AS] alias ]  
    [, ...]  
    [WHERE ...]  
    [GROUP BY ...]  
    [HAVING ...]  
    [ORDER BY ...]  
    [LIMIT ...]  
    [OFFSET ...]  
);
```

## Arguments

|              |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ON           | Required | The table, view, or query whose contents or results the Stream function operates on. To get data from a table, you must use the <i>actual table name</i> ; you cannot use an alias.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| PARTITION BY | Optional | Introduces an <i>expression</i> that partitions the table's contents, view's contents, or query's results before the function operates on them.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| ORDER BY     | Optional | Introduces an <i>expression</i> that sorts the table's contents, view's contents, or query's results before the function operates on them.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| SCRIPT       | Optional | Introduces the name of the script or executable to be run. If the runtime needed to run the script is non-obvious, then you must pass the runtime executable name <i>before</i> the script name. The names are surrounded in a common set of single-quotes, in the form:<br><br><pre>SCRIPT ('python mapper.py')</pre> Aster Database chooses the runtime executable in the same manner as a typical command shell. That is, Aster Database follows the usual Unix convention of looking at the first line to see if it starts with an interpreter directive. For example, mapper.py has the following first line:<br><br><pre>#!/usr/bin/env python</pre> so the Python runtime will be used automatically. This means that you can write your SCRIPT clause like this, omitting the runtime name:<br><br><pre>SCRIPT ('mapper.py')</pre> If there is no such directive, then the program will run only if it is an executable binary (for example, a compiled C/C++ program).        |
| MEM_LIMIT_MB | Optional | Sets the virtual memory limit in MB of the script process to the specified value per Stream function invocation.<br><br>For example: <ul style="list-style-type: none"> <li>• To set the memory limit to 1000 MB:<br/><br/> <pre>SELECT * FROM stream (ON numbers_small     SCRIPT ('myscript.py') OUTPUTS('outputline varchar')     MEM_LIMIT_MB('1000'));</pre> </li> <li>• To remove the memory limitation:<br/><br/> <pre>SELECT * FROM stream (ON numbers_small     SCRIPT ('myscript.py') OUTPUTS('outputline varchar')     MEM_LIMIT_MB ('unlimited'));</pre> </li> </ul> If the Stream process exceeds the memory limit, the operating system stops the Stream process and this error message appears in ACT:<br><br><pre>ERROR: SQL-MR function STREAM failed: Stream process exited with non-zero exit value (1)</pre> You can set the default memory limit per process using the STREAM_MEM_LIMIT environment variable in asterenv.sh. The default value is set to 4000 MB. |

|            |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OUTPUTS    | Optional | <p>Specifies the names and types of output columns from this call to the Stream function. By default, the output from a Stream script consists of two columns: (key varchar, value varchar). To specify different column names and/or types, use the OUTPUTS clause. In the OUTPUTS clause, each '<i>column_name column_type</i>' pair must be surrounded with single-quotes. For example, you might add the following example clause if your function produces pairs of given names and counts:</p> <pre>OUTPUTS ('givenname varchar', 'count int')</pre> <p>You can specify the star character “*” in the OUTPUT clause to represent the column names and types of the input of the Stream function. For example, consider this:</p> <ul style="list-style-type: none"> <li>The table testtable has two columns: test_id int and test_number float.</li> <li>repeat.py outputs rows with column types int, float, and int and float, in that order.</li> </ul> <p>You could specify the schema of the output table as follows:</p> <pre>SELECT * FROM stream (     ON testtable SCRIPT('repeat.py')     OUTPUTS('id int', 'number float', '*') );</pre> <p>The output columns have this schema:</p> <pre>'id int', 'number float', 'test_id int', 'test_number float'</pre> <p>If the db setting enable_quoted_identifiers='off', the star can be enclosed in single or double quotes.</p> <p>If the db setting enable_quoted_identifiers='on' (default), the star must be enclosed in single quotes.</p> |
| NULLSTRING | Optional | <p>Changes NULL values in the input table to <i>null_string</i>, before sending the input data as an STDIN stream to the Stream script.</p> <p>Changes <i>null_string</i> values in the columns of the output table to NULL.</p> <p>If you do not use this argument, the default behavior is:</p> <ul style="list-style-type: none"> <li>INPUT: If there is a NULL value in the input row, the Stream function does not add any value between the two corresponding column delimiters sent as STDIN to the Stream script process.</li> <li>OUTPUT: Given a value (possibly empty) between two consecutive delimiters in the STDOUT of the Stream script process, the Stream function adds the value to the corresponding column in that row. If the script writes less columns than expected, the remaining columns are filled with NULL values.</li> </ul> <p>NOTES:</p> <ul style="list-style-type: none"> <li>To minimize overhead, it is best for the NULLSTRING to be as short as possible (for example, an empty string " ").</li> <li>The STDOUT of the script should be in UTF-8 to allow for NULLSTRING detection.</li> </ul> <p>EXAMPLES:</p> <pre>SELECT * FROM stream (ON mytable SCRIPT('myscript')     OUTPUTS('line varchar') NULLSTRING('MYNULL')) ORDER BY 1; SELECT * FROM stream (ON (select (1)) SCRIPT('myscript')     OUTPUTS('a int','b int','c int') NULLSTRING('')) ORDER BY 1;</pre>                                                                                              |
| DELIMITER  | Optional | <p>Specifies the column-delimiter character. You only need to use this clause if your Stream script does <i>not</i> follow the default behavior of using tabs to delimit output columns. In the DELIMITER clause, you must surround the character in single-quotes. In your Stream function implementation code, you must specify this character as the column delimiter. Here is an example DELIMITER clause that declares the pipe character to be the delimiter:</p> <pre>DELIMITER (' ')</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

|    |                                                                                                                                                              |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AS | Optional Provides an alias for the Stream function call in this query. Using an alias is optional, and, when declaring an alias, the AS keyword is optional. |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Script Invocation with the Column Name in Quotes

The column name in the OUTPUTS clause can be a quoted string (possibly multi-word and with spaces). The only illegal characters in the quoted string are single quotes, double quotes, backslashes, and the character with code zero.

The column name can also be enclosed in square brackets. Quoted identifiers surrounded by square brackets can contain any character except the following characters: single quotes, double quotes, square brackets, backslashes, and the character with code zero.

The double quotes or square brackets are not captured in the column name.

## Examples

### Running a Python Script

This example runs mapper.py, a Python script that reads a line of text from the input stream and generates a row for each word in the line.

#### *The mapper.py Script*

```
#!/usr/bin/env python
#
# This is a very basic mapper that splits a line of prose into
# individual words. It is written in Python. Its output can be
# used in counting the occurrences of common words in a text.
#
import sys

def generateWords(line, delims):
    startidx = 0
    curidx = 0
    while curidx < len(line):
        if line[curidx] in delims:
            yield line[startidx:curidx]
            while curidx < len(line) and line[curidx] in delims:
                curidx += 1
            startidx = curidx
            curidx += 1
        yield line[startidx:]

while True:
    line = sys.stdin.readline()

    # Break on EOF.
    if line == "":
        break

    for word in generateWords(line, delims= '\n\t'):
        if len(word) > 0:
```

```
print '%s\t%s' % (word, 1)  
sys.stdout.flush()
```

### ***Query to invoke mapper.py***

This example query uses the input table “input\_table” when invoking the mapper.py function.

#### **Example input table**

Table 3 - 15: Example input table: input\_table

| content                  |
|--------------------------|
| Old MacDonald had a farm |

#### **Example STREAM query**

Use ACT to run this SQL-MR query, which calls the Script API:

```
SELECT * FROM STREAM  
(ON (SELECT content FROM input_table)  
SCRIPT ('python mapper.py')  
OUTPUTS ('word varchar', 'count int'));
```

#### **Output**

Run on a small sample table, this generates:

Table 3 - 16: Example output table

| word      | count |
|-----------|-------|
| Old       | 1     |
| MacDonald | 1     |
| had       | 1     |
| a         | 1     |
| farm      | 1     |

#### **Running a UNIX Command**

This example runs a UNIX command:

```
select * from stream  
(on (select 1) SCRIPT ('ls -F')  
outputs ('line varchar'));
```

The query executes the command `ls -F`, which outputs the names of the files in the current working directory using the varchar type.

## STREAM Script Execution Environment

### Bash Unix Shell

SQL-MR runs the Stream script through the Bash Unix shell. As a result, all the functionality and built-in commands of Bash are available for Stream script execution.

### PATH Environment Variable

The \$PATH environment variable is used for Stream script command execution. To view the path, run this query:

```
select * from stream (on (select(1)) SCRIPT ('echo $PATH'));
```



---

The temporary SQL-MR execution directory can change across query executions. Therefore, do not use a specific SQL-MR directory path. Instead, to refer to the installed files placed in the directory, use a relative path starting with "./".

---

### Path and Names of Installed Files

Installed files are available within the current working directory of the Stream function. Installed files are stored in subdirectories corresponding to their schemas within the SQLMR execution directory.

In addition, files of schemas in the search\_path are placed in the top level SQLMR execution directory. This lets you refer to files without a schema-qualified name from the query using Stream. If two files with identical names exist in different schemas, only the file with higher priority will exist.

The files' priority is determined by the following rules:

- Files that appear in schemas in the search path have higher priority than those that do not.
- Files that appear in schemas in the search path have priorities ordered by their search path. The earlier the schema appears in the search\_path, the higher the priority.
- Files that appear in schemas not in the search path have priorities that correspond to the alphabetical order of the schemas.

### Valid Installed File and Schema Names

Because installed files are placed into schema-qualified subdirectories, the installed file names and schema names cannot contain the slash character ("/").

### Accessing Installed Files

To access the files in the current working directory from a script, use a relative path ("./"). The script runs with the user and group name "extensibility." The user "extensibility" is guaranteed to have execute permissions to those files. Also, by default, the user "extensibility" has read and write permissions to those files.

The script can create files in the current working directory or other directories, such as /tmp. If the script creates files, it sets the owner and associated group allowed to access the file to "extensibility," and gives the user and the group these permissions:

-rw-rw-r--

## Creating, Installing, and Invoking a Stream Function

To create and install a Stream function, do the following:

- 1 Write your script in the language of your choice.

To write the function, write a script or program that reads rows from its input stream (stdin) and writes rows to its output stream (stdout).

In the output, your program must delimit each row with a newline (\n) character and delimit each column with the character expected by your queries (read the description of the DELIMITER clause for an explanation). By default, the column delimiter is a tab (\t) character.

- 2 Ensure that the script exits with exit(0) after a successful execution.
- 3 Run ACT and connect to your Aster Database:  

```
$ act -U beehive -h <IP address of the queen>
```
- 4 Install your script as a file in Aster Database.

To do this, use the \install command at the ACT SQL prompt. (See “[Manage Functions and Files in Aster Database](#)” on page 140 for more details.)

For example, to install the mapper.py Python script, enter this command:

```
beehive=> \install mapper.py;
```



**Tip!** The script, `mapper.py`, is a sample shown in “[Running a Python Script](#)” on page 120. You can contact Teradata Global Technical Support (GTS) for instructions on obtaining it.

- 5 (Optional) Use the \install command to install any other file needed by your script.
- 6 To invoke a Stream function, enter the SQL-MR query that calls the STREAM API.

For example:

```
beehive=> SELECT * FROM STREAM
beehive->(ON (SELECT content FROM input_table)
beehive->SCRIPT ('python mapper.py')
beehive->OUTPUTS ('word varchar', 'count int'));
```

## Stream Behavior

### No Reading of All Input Rows

A Stream script does not have to read all of the input rows. Just like any other SQL-MR function, the script can exit before reading any or all of its input. If so, a log message is written to the SQL-MR logs. For example:

```
[STREAM]      Stream process exited before reading all of STDIN
```

The script does not throw any exception or error message.

### Location of STDERR

STDERR from a script is written to the STDERR of the SQL-MR execution framework and can be found in its logs.

## Error Detection

Any non-zero exit value of the Stream script is interpreted as an error and causes an exception to be thrown and the query to fail.

**Table 3 - 17** describes the exit values.

Table 3 - 17: Bash exit values

| Exit Value | Description                         |
|------------|-------------------------------------|
| 2          | Misuse of Bash shell built-ins.     |
| 126        | Invoked command cannot be executed. |
| 127        | Command not found.                  |
| 128        | Invalid exit argument.              |



We recommend that you do not override these exit values in your Stream scripts.

## Troubleshooting Script Crashing

If your Stream script crashes, try the following to resolve the issue:

- If you have access to the worker nodes, try to execute the script from the command line on the worker node. This is a sanity check that the script can be executed. Of course, one would have to provide the script's STDIN.
- Look at the Stream execution logs for possible crash causes. The logs are captured by the SQL-MR logging mechanism. You can view the logs in the AMC.
- To view the row output of the script, you could specify the information to generate in the arguments clause of Stream:  
`OUTPUTS ('line varchar')`
- The “catch-all” error exit code is 1. If your Stream query execution results in an exit code of 1, one possibility is that the script is exceeding Stream’s default memory limit of 4 GB. To test whether this memory limit is the cause of the error, remove the memory limit by specifying the “unlimited” option for the `MEM_LIMIT_MB` argument of Stream.

# Using the R Programming Language and Environment

- Overview of R
- Installing R on Aster Database
- Installing R from a Local Repository
- Installing Prerequisites for Optional R Packages
- Execution Model For R
- Writing an R Program to Run Inside the Aster Database
- Data Type Mapping Between R and Aster Database
- Troubleshooting

## Overview of R

R is an open source programming language and software environment for statistical data analysis and graphics. The R language is highly popular among data scientists for advanced data analysis and model development.

With a library of over 3,000 add-in packages developed by leading experts and made available in the Comprehensive R Archive Network (CRAN), R provides a wide range of analytic functions covering areas such as time series analysis, classification, clustering, data smoothing, linear and generalized linear models, nonlinear regression models, resampling methods, classical parametric, and nonparametric tests.

For a listing of all the R packages, go to <http://www.r-project.org> and click Packages.



Teradata does not ship the R environment with the Teradata Aster database because R is Open Source and under the GPL license. Also, Teradata does not provide support for R. As described in this document, Teradata provides mechanisms for facilitating the installation, administration, and integration of R, which you can download from the open source community, on the Aster Database cluster.

## Goal of R Integration with Aster Database

R is designed to operate in a single server (single-threaded) on data that is entirely in the main memory of the system. Hence, R fails when the data becomes too large to fit in memory. This limitation is exacerbated by the call by value semantics of R execution, which leads to many copies of data being created in memory as data flows from one function to another.

Data scientists and statisticians using R routinely analyze large data stored in relational databases. Currently, the only option available for data scientists to analyze data stored in a relational database is to read data out to the R environment. This leads to a number of problems, including time-consuming data extraction from (and export to) relational databases. This typically prohibits interactive data analysis, unnecessarily duplicates data storage in the organization, and requires a system with large amounts of memory and storage to run R and process large amounts of data.

The R integration in the Aster Database is aimed at addressing these challenges by enabling the in-database execution of R, both to avoid extraction of data from Aster Database and to scale R to large data sets through parallelized execution, which enables users to run multiple instances of their programs over partitioned data.

Specifically, the R integration enables simplified installation and administration of R and R optional packages, installation of user R programs, execution of user programs in such a way that each R instance directly accesses the data partition stored in each vworker of the Aster Database cluster.

## Terminology

- Base R—The set of standard packages that are automatically available in any R installation. These standard (core) packages include the basic functions that are required for R to work as well as standard statistical and graphical functions.
- R Packages (optional)—All statistical functions beyond the base statistical packages that can be downloaded from CRAN mirrors.
- Aster Package Manager (APM)—An Aster Database cluster service that manages all third-party software installed on your Aster Database cluster. Currently, only the R programming language is supported.
- Auxiliary root area (/opt/aster/third-party)—Apparent root directory for all third-party software installed on Aster Database.

## Supported R Functionality in Aster Database

- Support for R installation/uninstallation on the Aster Database cluster.
- Support for R package installation/uninstallation in the default location on all cluster nodes.
- Support for R and R package installation from an online repository or a local repository containing RPMs of R and all their dependent RPMs.
- Automatic cleanup after failed/incomplete installs or upgrades (for example, after successful R installation on some cluster nodes and failed installation on others nodes), as well as reporting the failure to the log.
- Support for transparent new cluster node addition (node synchronization). When a new node is added to the cluster, R and all installed R packages are deployed on that node automatically.
- Maintaining of consistency of R and its optional package on all the nodes in the cluster.
- SQL-driven running of R programs over partitioned data on each virtual worker via the SQL-MR Streams module.

## Unsupported R Functionality in Aster Database

- The installer installs/uninstalls the latest version of base R and there is no multi-version support in this release.
- The R installer does not guarantee proper R functionality after an Aster Database upgrade.
- R optional package installation/uninstallation from Rscript/R Shell is not allowed.

- R upgrade is not supported. To upgrade R, uninstall it, then install the new version. To upgrade R optional packages, just install the optional R package, which installs the latest R optional package.

## R Installation/Uninstallation

During R installation, a complete installation is first carried out on the queen and cluster-wide synchronization of the auxiliary root is used to install R on all the workers. Likewise, uninstallation is first performed on the queen and a cluster-wide synchronization of the auxiliary root is used to remove R from the cluster.

The base R installation requires a number of dependency packages. The R package installer internally calls the Yum (for Red Hat) or Zypper (for SLES) package installer. These package installers install the latest version of a package or group of packages while ensuring that all dependencies are satisfied. You need to provide a valid repository mirror (either official Red Hat/SLES mirrors or a suitable local repository mirror) to resolve the distribution package dependencies using the appropriate ncli `apm` command options. For more information about the ncli `apm` commands, see the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*.

## Installing R on Aster Database



Before you install R, back up these files because the R installer overwrites them:  
`/opt/aster`, `/opt/aster/third-party`, and `/opt/aster/third-party/R`.

This procedure uses some ncli commands. For more information on ncli, see the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*. To install R on Aster Database:

- 1 Log on to an Aster Database queen as root.
- 2 Set up a SUSE repository by running this ncli command:

```
# ncli apm administer R --setuprepo=<repoName>,<repoURL>
```

To set up a SLES repository, enter this command:

```
ncli apm administer R --setuprepo=sles,  
"http://mirror.example.com/sles/11.sp1/x86_64"
```

- 3 Install R on the queen and all the worker nodes of your Aster Database cluster:

To install base R:

```
# ncli apm install R
```

To install a specific R package, use this command:

```
# ncli apm install R --packages=<packages>
```

For example, to install the getopt package, which is commonly used in Rscripts on Aster Database, enter this command:

```
# ncli apm install R --packages=getopt
```

To install multiple packages, provide a comma-separated list of the packages as in this example:

```
# ncli apm install R --packages=dave,forensic,genlasso  
For a listing of all the R packages, go to http://cran.r-project.org and click Packages.
```

## Installing R from a Local Repository

You can also create a local repository to install R. This local directory should contain all the R and associated dependency packages. Note that there are some packages, especially R packages, that are not found on the Redhat and SLES mirrors.

### Required R Package Locations

- R-java-devel
- R-javatexinfo-tex

For SLES, the required packages, available on [http://download.opensuse.org/repositories/devel:/languages:/R:/base/SLE\\_11\\_SP1/x86\\_64/](http://download.opensuse.org/repositories/devel:/languages:/R:/base/SLE_11_SP1/x86_64/), are:

- R-base
- R-base-devel

## Installing R

### *Installing R on SLES:*

- 1 Ensure that the createrepo package is installed.

Createrepo is not a native SLES package. You can install createrepo from a SLES mirror using zypper or yast.

- 2 Remove any traces of R from your system:

```
ncli apm remove R
```

- 3 Create the repository directory in the local area:

```
mkdir <local directory path>
```

- 4 Download the two SLES R packages below to the local repository from this URL or an alternative R mirror, replacing <version> with the version you are installing:

[http://download.opensuse.org/repositories/devel:/languages:/R:/base/SLE\\_11\\_SP1/x86\\_64/](http://download.opensuse.org/repositories/devel:/languages:/R:/base/SLE_11_SP1/x86_64/)

- R-base-<version>.x86\_64.rpm
- R-base-devel-<version>.x86\_64.rpm

- 5 Create the local repository:

```
cd <local directory path>  
createrepo .
```

- 6 Install R from the local repository you created in the previous step:

```
ncli apm install R  
--repo=sles,<SLES mirror path>  
--repo=sles1,file://<local directory path>  
--usedefaultrepo=false
```

For example:

```
ncli apm install R  
--repo=sles,"http://mirror.asterdata.com/sles/11.sp1/x86_64"
```

```
--repo=srepo,file:///home/beehive/download
--usedefaultrrepo=false
```

## Configuration Parameters

[Table 3 - 18](#) describes R configuration parameters, which are stored in /home/beehive/bin/utils/R/rpackage.cfg.

Table 3 - 18: R Configuration Parameters

| Parameter           | Description                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Epelrepo            | This is the default fedora mirror for downloading R packages for a Red Hat cluster. For example:<br><code>epelrepo=http://download.fedoraproject.org/pub/epel/6/x86_64</code>                                                                                                                                                                                                            |
| alternateepelrepo   | Alternate fedora mirror repository that you can reconfigure if the original repository is not accessible.<br><code>alternateepelrepo=ftp://mirror.cs.princeton.edu/pub/mirrors/fedora-epel/6/x86_64</code>                                                                                                                                                                               |
| Epelrelease         | The Epel release package that sets up the fedora mirror for R package installation. This parameter might need to be updated if the Epel release version changes. For example:<br><code>epelrelease=epel-release-6-8.noarch.rpm</code>                                                                                                                                                    |
| Scientificmirrorsrc | The URL link to install the texinfo-tex package. By default, this URL points to:<br><code>http://ftp1.scientificlinux.org/linux/scientific/6.0/x86_64/os/Packages</code><br>However, you can specify an alternate URL path to download the texinfo-tex package. For example:<br><code>scientificmirrorsrc=http://ftp1.scientificlinux.org/linux/scientific/6.0/x86_64/os/Packages</code> |
| texinforpmname      | The name of the Texinfo-tex RPM package. For example:<br><code>texinforpmname=texinfo-tex-4.13a-8.el6.x86_64.rpm</code><br>This parameter might need to be updated if the package version changes in the Scientificmirrorsrc area.                                                                                                                                                       |
| Slesrepo            | This is the default openSuse mirror for downloading R packages for the SLES mirror.<br><code>slesrepo=http://download.opensuse.org/repositories/devel:/languages:/R:/base/SLE_11_SP1</code>                                                                                                                                                                                              |
| rpackagesource      | The R package source to download the R optional packages. This link might change if you want to point to alternate R optional package repositories.<br><code>rpackagesource=http://cran.r-project.org</code>                                                                                                                                                                             |
| Baseoptionalpkgs    | Base optional packages that are installed as a part of the R installation. This package source needs to be updated if the base optional package list changes.                                                                                                                                                                                                                            |

## Installing Prerequisites for Optional R Packages

Because R is installed in the Auxiliary-Root area, some optional packages might have dependencies that might not be installed during the R installation. Some packages might have additional system requirements that can be resolved by installing RPM packages in the R Sandbox.

Normally the R-cran mirror package documentation mentions the system requirements for optional R packages. You can check if a package is installed on the cluster using this command:

```
ncli apm show R --rpmpackage=<package name>
```

If the required package is not installed, you can install the package using this command:

```
ncli apm install R --rpmpackages=<list of comma-separated RPM packages>
```

### Example

The RCurl package requires libcurl-devel package to be installed in the system for successful installation.

- 1 Check for the installation status of libcurl-devel on the system using this command:

```
ncli apm show R --rpmpackage=libcurl-devel
```

If the package is not installed on the system, install the libcurl-devel package using this command:

```
ncli apm install R --rpmpackages=libcurl-devel
```

- 2 Install the RCurl package on the system using this command:

```
ncli apm install R --packages=RCurl
```

## Execution Model For R

R is installed in sandboxed area on each cluster node at /opt/aster/third-party/R. This requires R to be executed via the chroot command:

```
$ chroot /opt/aster/third-party/R <R/Rscript>
```

For simplicity, we have provided a wrapper script(Rexec) found in /home/beehive/bin/utils/exec/Rexec to allow R execution either from Act or as a root user executing Rscript files in /home/extensibility area.

This approach has numerous advantages:

- It makes it relatively easy to synchronize R across the Aster Database cluster or deploy R on a newly added cluster node.
- It minimizes security risks from R programs because the jail limits the portion of the file system the R program can see to the auxiliary root of the jail. In other words, the R program does not have access to the entire file system.
- It eliminates the possibility of conflicts between the dependent libraries of R and the other libraries installed on the cluster node because all the dependent packages installed by R and R packages are isolated from the rest of the software installed on the node.

## R Program Invocation

During the invocation of a query that runs a user-installed R program via the Stream SQL-MR function, the Stream function calls a runner function called Rexec, which runs as the extensibility system user, to set up the R environment and invoke the user-installed R program.

As a part of this setup, the extensibility user needs to be added to the sandbox area. Also, the SQL-MR working directory is mounted on the auxiliary root to allow Rscripts to read any ancillary installed files they need to access. Then, Rexec changes to the directory in the auxiliary root area containing the user-installed R program, as well as the ancillary installed files, and executes the R program.

To invoke a query that runs a user-defined Rscript via the Stream SQL-MR, use this command:

```
Rexec [<options>] [-e <expression>] <Rscript_file>
```

This command assumes that the specified Rscript file, the included model files, and other included Rscript files are already installed using the act \install command.

For example:

```
Rexec --vanilla naivebayes.R
```

## Writing an R Program to Run Inside the Aster Database

The main supported data structure to exchange data between the Aster database and the R environment is the R data frame, which has a similar data representation to a table in the Aster Database.

A typical R program may use `read.table` to read in input data and use `write.table` to write a result out, although you can use any other mechanism as long as a delimited list of values and rows are read and written by the program.

This example shows a simple R program, called `echo_input.R`, that receives a row from an Aster Database vworker and writes out the same row back.

```
IN = file(description="stdin", open="r")
while(1)
{
  # Create a frame to hold the input rows, without HEADER,
  # and also to deal with end of stream
  # read.table() is called inside the try block to detect when the
  # program reaches the end of rows.

  frame<-try(read.table(IN,header=FALSE,sep="\t",quote="",nrows=1),silent=TRUE)
  if(inherits(frame,"try-error"))
    break
  write.table(frame,stdout(),col.names=FALSE,row.names=FALSE,quote=FALSE,sep="\t")

}
```

The Aster Database vworker and the R program exchange rows using standard input and output. Hence, the R program should be written to read in a row from the standard input and write a row to the standard input. The program does not need to generate an output for each row read. For example, it can generate one row after reading the entire set of input rows.

You can invoke R programs using the SQL-MR Stream module. For example, you can invoke `echo_input.R` using the following SQL query:

```
SELECT *
FROM stream(
  ON mytable
  SCRIPT('Rexec --vanilla echo_input.R')
  OUTPUTS('*')
);
```

The R program can also take any input parameter during its invocation, just like the way parameters are passed to RScript invocations, and parse the parameters using an appropriate package like getopt.

Also, if the R program needs to operate on a subset of columns of the input table, the ON clause can be used to write a subquery that projects out the required columns. Also, if the R program returns rows with a different schema from the input rows or adds more columns to the input rows, then you can use the OUTPUTS clause to specify these columns.

For more information about the Stream module, see [Stream API for Python, Perl, and Other Types of Scripts \(page 117\)](#).

Note that returning rows from an R program involves two distinct steps:

- The first step is to specify the names and types of rows returned by the R program in the OUTPUTS clause. The column names used should comply with Aster database's identifier naming conventions.
- The second step is to write the R program to return rows without any headers. Also note that the correct delimiters need to be used.

If you need to transfer lists, matrixes, or other R data structures from R to the Aster Database, you have to convert them to data frames using the as.data.frame() function. Also, the R program must have only one result that is in the form of a data frame.



You can run an R program that does not read in any input data but produces an output table. In this case, you can either use some dummy table in the ON clause or just use a subquery like "SELECT 1" in the ON clause and ignore the input in the R program.

## Writing an Output File from an R Program to the File System

It is possible to have your R program consume an input table and write the output or some other data to the file system as long as R is running as a user that has permission to write in the target directory. If you do not specify any directory, your file is written in the default working directory, which gets cleaned up at the end of the query's execution. Hence, you need to specify some other directory under the auxiliary root (for example, /opt/aster/third-party) to which you have a write permission.

When writing files out, you need to keep in mind that multiple instances of your R program are running on each node of the cluster. Hence, make sure that the name of your output file from each instance contains some unique identifier.

This example shows a simple R program, called echo\_input2.R, that receives a row from an Aster Database vworker and writes out the same row back to a file.

```
IN = file(description="stdin", open="r")
while(1)
{
  # Create a frame to hold the input rows, without HEADER,
  # and also to deal with end of stream
  # read.table() is called inside the try block to detect when the
  # program reaches the end of rows.
```

```
frame<-try(read.table(IN,header=FALSE,sep="\t",quote="",nrows=1),silent=TRUE)
if(inherits(frame,"try-error"))
  break
outFile=file('/tmp/output.txt','w')
write.table(frame,outFile, row.names = FALSE,append = FALSE,
            col.names = FALSE, sep = "\t")
}
```

## Using the PARTITION BY Clause in Queries that Invoke R Programs

The PARTITION BY clause ensures that the rows that contain the same value for the partitioning (splitting) expression will occur on the same vworker for processing. Because they occur on the same vworker, they will be processed by a single R program task.

The R program is invoked once per vworker, not once per partition. The PARTITION BY ensures that each partition is processed by one vworker. But when multiple partitions are processed by a single vworker (e.g. when there are three or more partitions in a system with two vworkers), then more than one partition may end up on a single vworker. In such a case, the script detects the partition boundary (i.e. passing the partitioning column as a script argument) and ensures that partition-wise results are returned for each partition. Hence, the semantics of PARTITION BY when used with R are not exactly the same as that of a Reduce.

So essentially, you can be assured that all rows of a partition will be processed by a single vworker task, but that does not mean that the vworker will be processing just one partition.

You may choose to use the statistics returned from each vworker as your end result, or you may choose to aggregate the vworker results into a combined result using a client-side R program. In the latter case, make sure that your computation fits the split-apply-combine (data parallel) paradigm.

The following example R program shows the use of a PARTITION BY clause. This program, called partition\_sum.R, computes the sum of all the values in each row and then adds that sum to a partition-wise sum:

```
IN = file(description="stdin",open="r")
partition_sum=as.integer(0)
while(1)
{
  row_frame<-try(read.table(IN,header=FALSE,sep="\t",quote="",nrows=1),silent=TRUE)
  if(inherits(row_frame,"try-error"))
    break
  row_sum<- apply(row_frame[,3:ncol(row_frame)],1,sum)
  partition_sum <- row_sum + partition_sum
  last_row<-row_frame
}
write.table(t(c(last_row[,2],partition_sum)),stdout(),col.names=FALSE,row.names=FALSE,quote=FALSE,sep="\t")
```

Notice that the write-table function is called after all input rows are consumed.

The following is the SQL-MR query for running the above program in the Aster Database:

```
SELECT *
FROM stream (
```

```
ON (select a, b, c, d from inputtable
PARTITION BY b
SCRIPT ('Rexec --vanilla partition_sum.R')
OUTPUTS ('b int', 'sum int')
);
```

This query produces a result for each vworker. Aster Database redistributes the rows in the input table based on their values for the column “b” in such a way that all rows that share the same value are processed on one vworker, by one instance of the R program.



The Aster Database provides a special partitioning construct that can be used to execute just one instance of the R program thereby processing the entire input table by one R program instance. To use this construct, simply use “PARTITION BY 1”. This construct enables you to push any R program that cannot be parallelized (for example, model development) into the Aster Database and avoid having to extract the data out and instead exploit the larger computing power of the Aster Database.

## Data Type Mapping Between R and Aster Database

The exchange of R data and Aster Database data is done by mapping an R data frame to an Aster Database table. The R program needs to be able to accept and map the input data to the right types, and the output from the R program should match what is specified in the OUTPUTS clause of the Stream function (except in the case of a “\*” expression in the OUTPUTS clause, which expects data with the same types as the input data).

**Table 3 - 19** shows the recommended mapping of data types from R to Aster Database.

Table 3 - 19: Data Type Mapping

| R Type           | SQL Type                                        |
|------------------|-------------------------------------------------|
| integer          | int2, int4                                      |
| numeric (double) | int8, float4, float8, float(p), numeric [(p,s)] |
| boolean          | logical                                         |
| bytea            | object                                          |
| everything else  | character                                       |

However, there are a few mapping issues that you need to be aware of, which are described in the following sections.

### Character

Character types can be used to exchange a wide variety of data between Aster Database and R, but note that the exchange of characters can be very time-consuming. So, it is recommended that you consider whether it is necessary for a given character column to be transferred to the R environment or whether it can be substituted with the integer or double data type.

### Bit Data Type

The R environment cannot automatically determine the type of bit data and properly handle it. Hence, the Rscript needs to explicitly accept the bit string as a character string and convert

it to a raw data type. For example, when using the `read.table()` function, you can achieve this conversion as follows:

```
frame<-try(read.table(IN,header=FALSE,sep="\t",quote="",  
nrows=1,colClasses=c("character")), silent=TRUE)
```



The Bit Varying data type is properly handled by the `read.table()` function.

## Bytea Data Type

Bytea data values are handled as R raw types by the R environment. However, note that the `read.table()` function uses the number symbol (“#”) as a comment character and hence data after “#” is truncated. One way to avoid the truncation when reading bytea values using `read.table()` is to turn off the interpretation of comment characters as shown in the this example:

```
frame<-try(read.table(IN,header=FALSE,sep="\t",  
quote="",nrows=1,comment.char="",silent=TRUE))
```

## BIGINT Datatype

The base R environment uses finite precision arithmetic and numbers are accurately represented only up to 15 or 16 decimal places. Hence, you need to make sure you take the proper care when working with bigint values that are larger than the values that can be accurately represented. To transfer such numbers between Aster Database and the R environment, you can use a character representation as shown in this example:

```
frame<-try(read.table(IN,header=FALSE,sep="\t",quote="",  
nrows=1,colClasses=c("character")), silent=TRUE)
```

You can then use packages like `int64` and `gmp` to convert the numbers back to 64-bit integers in your R program.

For example, using the `int64` package (<http://cran.r-project.org/web/packages/int64/>), you can convert a bigint represented as a character back to a 64-bit integer value as follows:

```
y<-as.int64(c("-3940427841425010000", "-4236711481380030000"))
```

Also, here is an example that shows how you can perform arithmetic with a BIGINT value:

```
# head -20 *  
==> bigints.csv <==  
1,4699533205482374612  
2,-3526377826377322350  
3,1  
4,-1  
5,0  
  
==> bigints.R <==  
#!/usr/bin/Rscript  
library(int64)  
IN = file(description="stdin",open="r")  
while(1)  
{  
    # Create a frame to hold input Rows ,  
    # without HEADER and also to deal with end of stream
```

```

# read.table() should be inside try block
frame<-try(read.table(IN,header=FALSE,sep="\t",quote="",
                      nrows=1,colClasses=c("character")),silent=TRUE)
if(inherits(frame,"try-error"))
    break

# instantiate a int64 object based on BIGINT in table and add 1 to it
value<-as.int64(frame$V1[1])
value = value + 1

# create a new frame with the original BIGINT value
frame2<-cbind(frame, value)
write.table(frame2,stdout(),col.names=FALSE,
            row.names=FALSE,quote=FALSE,sep="\t")
}

==> bigints.sql <==
DROP TABLE IF EXISTS bigints;
CREATE TABLE bigints(seq INT, num BIGINT) DISTRIBUTIVE BY HASH(seq);

# ncluster_loader -c -w beehive -B bigints.sql bigints bigints.csv
Trying to connect to the loader '192.80.170.44'.
Loading tuples using node '192.80.170.44'.
5 tuples were successfully loaded into table 'bigints'.

# act -w beehive
Welcome to act 05.10.00.00, the Aster nCluster Terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with act commands
      \g or terminate with semicolon to execute query
      \q to quit

beehive=> \install bigints.R
beehive=> SELECT * FROM STREAM( ON (SELECT num FROM bigints)
SCRIPT('Rexec --vanilla bigints.R') OUTPUTS ('*', 'sum BIGINT'));
      num      |      sum
-----+-----
        1      |      2
  4699533205482374612 |  4699533205482374613
       -1     |      0
-3526377826377322350 | -3526377826377322349
        0      |      1
(5 rows)

```

## Null (Missing) Value Handling

Null values in the Aster Database are typically mapped to an “NA” value in the R environment. However, the Aster Database environment passes null values as either the value “NULL” or as empty strings.

To instruct the Aster Database to replace null values with “NA” before passing them to the R environment, you can use the NULLSTRING argument of the Stream SQL-MR function, which allows you to use any value as a replacement for null values before they are passed to the R environment. Likewise, the Aster Database recognizes values given in the NULLSTRING argument as NULL values in data returned from the R environment.

For more information about the handling of null values, see [“Stream Function Usage” on page 117](#).

## Hash(#) Character Handling

The read.table() function uses the number symbol (“#”) as a comment character and hence data after “#” is truncated. One way to avoid truncation when using read.table(), which applies to all types, is to turn off the interpretation of comment characters as shown in the this example:

```
frame<-try(read.table(IN,header=FALSE,sep="\t",
quote="",nrows=1,comment.char=""),silent=TRUE)
```

## Troubleshooting

Possible reasons for the failure in Rscript execution via Stream are:

- Improper R installation on the Aster Database cluster.
- Improper installation of optional R packages, required by a user-defined Rscript, on an Aster Database cluster.
- Temporary command execution failure due to node failure.
- Stream execution failure.

To detect any of these possible failures, execute R from the command line on the queen:

- `Rexec R <command_line_argument>`  
This command invokes the R shell.
- `Rexec Rscript [<options>] [-e <expression>] <Rscript_file_arguments>`  
This command invokes the specified Rscript file, assuming that the Rscript file and ancillary installed files are located in /home/extensibility area or any sub-directory.  
If Rscript execution is not successful on the queen, the R installation files in queen are corrupted and R needs to be reinstalled on the cluster.
- Check for R installation on all the nodes using this ncli option:  
`ncli aptm install R --localconfig`
- If R is not installed on a worker, execute this command to ensure seamless R installation on all the reachable worker nodes of the cluster:  
`ncli aptm administer R --synchronize`

- Check for package installation status of all the optional packages by running this command:

```
ncli apm show R -packages=<package name>
```

Then, check for the existence of the directories in the output of this command on the worker:

```
ncli node runonanother ls /opt/aster/third-party/R/usr/lib64/R/library
```

If some packages are missing, run this command on the queen:

```
ncli apm administer R --synchronize
```

- If the stream execution still throws an error, this means that the problem could be in Stream execution and needs to be investigated.

## SQL-MapReduce Security

Installation and use of SQL-MapReduce functions and installed files in Aster Database is governed by the schema membership of the function or file, and by the SQL user's GRANTed privileges.

### Schema Membership for Installed Functions and Files

Every installed file or function must belong to a schema. The schema is used in two ways:

- 1 The schema provides the context for GRANTing users rights to install files and functions. To install a file, the user must have the INSTALL FILE privilege on the schema, and to create a function he must have both the INSTALL FILE and CREATE FUNCTION privileges on the schema. To run a function, the user must have the EXECUTE privilege on the *function*.
- 2 The schema provides namespace isolation. When your queries refer to an installed file or function, you must now follow the same namespace resolution rules that you would follow for any table or view. In other words, Aster Database now uses the schema search path or an explicit schema qualifier in the query to know which query an installed file or function resides in.

To call the function, the user must schema-qualify its name in her SELECT statement, or she can make sure that the function's schema is in her current schema search path. The file or function is usable on tables, views, and files in any schema to which the user has access.

Schema membership provides namespace isolation for installed functions and files, meaning that if, for example, one of your installed functions has the same name as a function used by another analyst in another schema, there is no ambiguity as to which function is used.

You are also safe from unwanted changes made by people who work in the same schema where you installed your function. Within a single schema, only the owner of a function can remove or overwrite it. This means that your installed functions are also protected from changes by other users who have function-installation rights in your schema.

## GRANT Privileges for Installed Functions and Files

The SQL user must have the appropriate privileges (granted using GRANT) in order to install, use, or download a function or file. To manage users' rights on an SQL-MapReduce function, the DBA or the function's owner uses the GRANT and REVOKE commands. These commands set the rules that govern who can install, run, download, and uninstall the SQL-MapReduce functions in each schema. For example:

Table 3 - 1: Privileges for Installed Functions and Files

| Task the user wants to do                                  | Privileges the user must have to perform the task                                                                                                                                                     |
|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Install a file                                             | User must have the INSTALL FILE privilege on the <i>schema</i> where he wishes to install it.                                                                                                         |
| Install a function                                         | User must (a) have the INSTALL FILE privilege and (b) have the CREATE FUNCTION privilege on the <i>schema</i> where he wishes to install it                                                           |
| Run a function                                             | User must have the EXECUTE privilege on the <i>function</i> .                                                                                                                                         |
| Download an installed function or file from Aster Database | To download a file, the user must (a) have the USAGE privilege on the <i>schema</i> from which she wishes to download the file and (b) be the owner of the file (or be the database administrator).   |
| Remove a function or file from Aster Database              | To uninstall a file, the user must (a) have the USAGE privilege on the <i>schema</i> from which she wishes to uninstall the file and (b) be the owner of the file (or be the database administrator). |

## User Permissions for Installed Files and Functions

You grant and revoke users' rights to functions using the commands shown below. For more complete descriptions of these commands, see the reference documentation for GRANT and REVOKE in the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.

### GRANT INSTALL FILE and GRANT CREATE FUNCTION on a schema

To give a user or group the right to *install files* and *create functions* in Aster Database, an Aster Database administrator must use one of GRANT commands that gives the user privileges in the context of a schema:

```
GRANT { INSTALL FILE | CREATE FUNCTION } [, ...] [ PRIVILEGE ]
ON SCHEMA schemaname [, ...]
TO { username | GROUP rolename | PUBLIC } [, ...]
```

Note that there is no support for delegating privilege management, because the WITH GRANT OPTION clause is not supported.

### GRANT EXECUTE on a function

To give a user or group the right to *run a function* in Aster Database, an Aster Database administrator or the function's owner must use the GRANT EXECUTE command that gives the user the privilege for the specific function:

```
GRANT EXECUTE [ PRIVILEGE ]
ON FUNCTION [ schemaname. ] funcname
TO { username | GROUP rolename | PUBLIC } [, ...]
```

Note that there is no support for delegating privilege management, because the WITH GRANT OPTION clause is not supported.

## REVOKE INSTALL

To deny a user or group the right to install functions and files in Aster Database, an Aster Database administrator must use the REVOKE INSTALL command:

```
REVOKE [ GRANT OPTION FOR ]
    INSTALL { FILE | FUNCTION } [,...] [ PRIVILEGES ]
    ON SCHEMA schemaname [, ...]
    FROM { [ GROUP ] rolename | PUBLIC } [, ...]
```

## REVOKE EXECUTE

To deny a user or group the right to run a function in Aster Database, an Aster Database administrator or the function's owner must use the REVOKE EXECUTE command:

```
REVOKE [ GRANT OPTION FOR ]
    EXECUTE [ PRIVILEGES ]
    ON FUNCTION [schemaname.] funcname
    FROM{ [ GROUP ] rolename | PUBLIC } [, ...]
```

## Upgrade earlier SQL-MapReduce functions to Aster Database 5.0.x

Because the SQL-MapReduce security controls were new in version 4.6-GA, the schema membership and user privileges for your 4.5.1 and earlier functions and files are automatically upgraded during the Aster Database upgrade from 4.5.1-hp5 to 4.6.x. As a result, your SQL-MapReduce functions will remain executable by all users who have rights to the public schema. To limit the set of users who can run functions, follow the instructions in the *Aster Database Release Notes*, version 4.6-GA, in the section “Upgrade SQL-MapReduce Functions and Files to the 4.6 Permissions Scheme.”

# Manage Functions and Files in Aster Database

This section explains how to install and manage *SQL-MapReduce functions* in Aster Database, and how to install and manage *files* in Aster Database. You install and manage functions and files using the \install command (or the `INSTALL FILE` command if you're using ODBC) and related commands in the Aster Database ACT tool. In the sections that follow, we'll show you how to use these commands.

## What can I install on the cluster?

You can install the following on your cluster:

- SQL-MapReduce functions: Compiled Java and C executables that can be called by name in the FROM clause of a SELECT.
- Scripts for STREAM: Each script is installed as a file that you will invoke in a `SELECT ... FROM STREAM` query.

- **Files:** Installed files are typically used to provide fairly static input or operational parameters to SQL-MapReduce functions and to Stream functions. Installed files can only be used in your SQL-MapReduce and Stream functions. Installed files are not directly invocable or accessible via SQL.

## Get information About Installed Functions

From the ACT command prompt (SQL prompt), you can use the \dF commands to find out what functions and files are installed in your cluster:

- \dE - Lists all the SQL-MapReduce functions that you have permission to run, based on your SQL user account. For an explanation of permissions, see “[SQL-MapReduce Security](#)” on page 138.
- \dF - Lists all installed files and SQL-MapReduce functions *in your current schema*.
- \dF \*.\* - Lists all installed files and SQL-MapReduce functions *in the database*.
- \dF+ - Prints the list of SQL-MapReduce functions installed. For each function, the output shows the name, schema, owner, upload time, and fingerprint of the function. Type “\dF+” to show functions in your current schema, or “\dF+ \*.\*” to show them for all schemas in the database. The fingerprint indicates the revision number of the function. When you are reporting suspected errors in SQL-MapReduce functions, please provide the function’s fingerprint to Teradata Global Technical Support (GTS).

### Related Topics

For information on *setting* users’ SQL-MapReduce privileges, see “[SQL-MapReduce Security](#)” on page 138.

## Checklist for Installing a Function

A number of steps are required to install a function and make it usable for other analysts. In later sections, we’ll provide complete descriptions of all the steps, but first let’s quickly walk through all the steps needed to install a function and make it usable:

- 1 Create the function, compile it, and package it (typically into a jar file). The name of the file you’re installing *must* match the name of the function as you coded it. See “[Build and Package the SQL-MapReduce Function](#)” on page 101.



**Tip!** Note that SQL-MR function names are not case sensitive. Function names will be automatically converted to all lowercase when they are installed.



**Tip!** SQL-MR function names within a schema must be unique. If you wish to replace an existing SQL-MR function with a new function of the same name, you must follow these steps:

- Remove the existing function by issuing \remove in ACT.
- Install the new function using \install
- Set permissions on the new function (See “[SQL-MapReduce Security](#)” on page 138).

2 Run Aster Database ACT and log in as the SQL user who will install and manage the function.

3 Use the `\install` command to install the function, taking care to specify the schema in which the function will be installed. The command has the syntax:

```
\install file_pathname schemaname/installed_filename
```

If your installation attempt fails, make sure your SQL user account has the INSTALL FILE and CREATE FUNCTION privileges. See “[SQL-MapReduce Security](#)” on page 138.

4 Type `\dF+ <function name>` to check which schema the function belongs to.

5 Use the GRANT command to give the EXECUTE privilege to users who will run the function. The syntax is, roughly:

```
GRANT EXECUTE  
ON FUNCTION <schema-name>.<function-name>  
TO <user-name or group-name or PUBLIC>;
```

## Test the Function

Now the function is installed and usable by all users to whom you’ve granted EXECUTE rights. Test your function by following the steps below to run it:

- 1 Run Aster Database ACT and log in as an SQL user who has the EXECUTE privilege on the function.
- 2 Invoke the function in a statement such as a SELECT or other data-retrieval statement. Make sure you schema-qualify the function’s name, or have its schema in your schema search path. If the function name contains uppercase letters, you *must* enclose the function name in double quotes.

For an example that shows how to install a function, see “[Install and Use a Sample Function](#)” on page 111.

## Install a File or Function

In this and the sections that follow, we explain the ACT and ODBC commands for installing and removing files and SQL-MapReduce functions. In this discussion, we refer to a file or function as “local” when it resides on your local file system, and as “remote” when it resides in Aster Database.

To install SQL-MapReduce functions, Stream functions, and other files, use the `\install` command in ACT and the `INSTALL FILE` command in ODBC. For an example that shows how to install a function, see “[Install and Use a Sample Function](#)” on page 111.

Note: `\install` and related tools do *not* allow you to manage Aster Database-supplied functions such as STREAM, STREAM and other system functions cannot be managed by anyone (including the database administrator).

### Syntax for installing files and functions

In ACT:

```
\install file_pathname [[schemaname/] installed_filename]
```

In ODBC:

```
INSTALL FILE 'file_pathname' [[schema_name/] 'installed_filename']
```

In ODBC only, if the *schema\_name* contains mixed case letters or spaces, you must surround the *schema\_name* only in double-quotes.

Once you have installed a function, you must set users' privileges to run it as explained in “[GRANT EXECUTE on a function](#)” on page 139.

If your \install attempt fails, ask the schema's administrator to grant you the right permissions. See “[GRANT INSTALL FILE and GRANT CREATE FUNCTION on a schema](#)” on page 139.

### The parameters for \install

The parameters for \install, for installing SQL-MapReduce functions and files:

- The argument, *file\_pathname*, is the path name of the to-be-installed file, relative to the directory where ACT is running. When you install a function, the name of the file *must* match the name of the function. See “[Build and Package the SQL-MapReduce Function](#)” on page 101.
- The optional *schema\_name* parameter specifies the schema in which the function will be installed. To install a file or function, your SQL user account must have the INSTALL FILE privilege in the schema. If you're installing a *function*, your account must also have the CREATE FUNCTION privilege in the schema.
- The optional *installed\_filename* parameter specifies the name of the file or function, as it will be referred to in Aster Database. The *installed\_filename* is the name SQL users will use when they SELECT from the function. In the current version of Aster Database, *you must use the file's actual filename*, including its suffix, but not including any directory names. We strongly recommend that all letters in the function name be lowercase. If the name contains any uppercase letters, users will have to surround the function name in double quotes in their SELECT statements that call the function.

### Notes on installing files and functions

If you do not pass a *schema\_name*, the file or function is installed in the first schema in your schema search path. If you do not pass an *installed\_filename*, the file or function's file name (not including its directory path) is used as its *installed\_filename* in Aster Database. Keep in mind that, when you call an SQL-MapReduce executable in your queries, you drop its filename suffix (“.class” or “.so”, for example), but when you operate on it with a function-management command (such as \download), you include its suffix.

## Make a Local Copy of a File or Function

The \download command in ACT (or DOWNLOAD FILE command in ODBC) makes a copy of the specified, installed file or function (identified by its *installed\_filename*, optionally *schema\_name*-qualified) and saves it on the file system where the ACT client is running.

### Syntax

In ACT:

```
\download [schema_name/] installed_filename [file_pathname]
```

In ODBC:

```
DOWNLOAD FILE [schemaname/] installed_filename [file_pathname]
```

Optionally, you can specify a file/path name for the saved file by supplying the *file\_pathname* argument. This argument can be just a file name or a path name, but the destination directory must exist on the file system where you're running ACT.

The \install and \remove commands can be used transactionally in a BEGIN / COMMIT block just like any transactional SQL command.

## Remove a File or Function

The \remove command (or UNINSTALL FILE in ODBC) removes the file or SQL-MapReduce function from an Aster Database. The *installed\_filename* is the name of the file or function to be removed. To remove it, you must be the file or function's owner or a superuser.

### Syntax

In ACT:

```
\remove [schemaname/] installed_filename
```

In ODBC:

```
UNINSTALL FILE [schemaname/] installed_filename
```

The \install and \remove commands can be used transactionally in a BEGIN / COMMIT block just like any transactional SQL command.

## FAQs About SQL-MapReduce and Stream

### Can I have a PartitionFunction that's invoked on all rows, cluster-wide?

Yes, you can PARTITION BY <CONSTANT> to place all rows on a single worker.

```
SELECT * FROM EXACT_MEDIAN(ON source_data PARTITION BY 1);
```

Notice! Be careful when using this technique. The source data specified in the ON clause must fit on a single node. Nevertheless, this formulation can be useful when a RowFunction computes per-worker summaries that are then merged into a final result.

### How do I save the outputs of an SQL-MapReduce function?

Use a "CREATE TABLE AS SELECT" statement.

```
CREATE FACT TABLE output DISTRIBUTE BY HASH (word) AS  
SELECT word, sum(count) FROM tokenize(ON documents) GROUP BY word;
```



# Section II Aster Database Administrator Guide

This section of the guide is written for database administrators, system administrators, and database superusers who manage an Aster Database. The section is divided into:

- [Database Management](#)
- [Data Modeling Best Practices](#)
- [Query Tuning Best Practices](#)
- [User Management](#)
- [Workload Management](#)
- [Export and Load Tools](#)
- [Teradata-Aster Database Connector](#)
- [SQL-H: The Hadoop/HCatalog Connector](#)
- [Backup and Restoration](#)
- [Aster Glossary](#)
- [Error Codes](#)
- [Parent/Child Tables](#)
- [Licenses for Tools Used in Aster Database](#)

This section of the guide assumes some knowledge of SQL.



# CHAPTER 1 Database Management

This section introduces topics related to the management of databases in Aster Database. It includes information on managing space and viewing query execution plans. Creating, dropping, and managing databases, tables and database objects is also covered.

This section contains:

- [Get Started](#)
- [Create and Drop Databases](#)
- [Manage Tables](#)
- [Manage Database Objects](#)
- [Manage Space](#)
- [Compression](#)
- [EXPLAIN Plan](#)

## Get Started

It is easy to start using Aster Database for data warehousing and analytics. The simplest approach is to connect to the default database “beehive” and start creating tables, loading data, performing queries, and so on. Within this default database, it is straightforward to create and manage tables. (See [“Manage Tables” on page 151](#).)

Databases are physically stored throughout the cluster but each database is logically separate and requires a unique user session for access. In other words, a single connection to Aster Database connects to a single database. Similarly, queries and other transactions may refer only to tables and other objects contained within a single database.

Putting tables in different databases creates strict isolation between them. If you expect that queries or other operations will be needed between certain tables, then those tables must be placed in the same database.

A database spans multiple virtual workers (vworkers). Each worker node usually contains multiple vworkers. A database is logically mapped to all vworkers across all worker nodes in Aster Database. This maximizes parallel performance by enabling queries to take advantage of all computational resources.

## Bring Up a SQL Prompt

Use the Aster Database Cluster Terminal (ACT) to connect to the SQL command line. On the queen, or another machine with ACT installed, type:

```
$ act -d beehive -h <queen_IP_address> -U beehive -w beehive
```

The first time you connect to Aster Database, you will connect as one of the default users:

### **Default users:**

- db\_superuser: (default password: db\_superuser) the most powerful user, with full administrative access to all database objects.
- beehive: (default password: beehive) the user that owns the default database, also called beehive. By default, the beehive user has no administrative rights.

Important! Immediately after you install Aster Database, you should change the default password of db\_superuser to one that is more secure.

## Create an Aster Database User

Create Aster Database SQL user accounts with the CREATE USER command in SQL. You can grant roles to users, which allow them to perform specific tasks. See [Default Roles and Users \(page 246\)](#) for details about roles.

### **Create the User:**

In this example, we create a new user, “jstrummer”. At the ACT prompt, type:

```
CREATE USER jstrummer PASSWORD 'm4gn1f1c3nc3';
```

### **Grant the Superuser Role:**

Create a new superuser by granting the db\_admin role to an existing account:

```
GRANT db_admin TO jstrummer;
```

### **Revoking Superuser Rights:**

Revoke a superuser’s rights by revoking the db\_admin role:

```
REVOKE db_admin FROM jstrummer;
```

You may not revoke the db\_admin role from the user 'db\_superuser'.

## Basic Data Import

Aster Database Loader (“ncluster\_loader”) is a command-line application for bulk-loading data into Aster Database. Based on the COPY command, Aster Database Loader provides an alternative to the SQL INSERT statement and offers much better performance and error handling. For more information on Aster Database Loader, see the *Aster Client Guide*.

## System Configuration

### Cluster-wide parameters

For information on cluster-wide parameters, please refer to “Data Dictionary Views” in the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.

### Setting up NTP

NTP (Network Time Protocol) Setup happens when Aster Database is installed. Once installation is complete, changing it requires root access or help from Teradata Global Technical Support (GTS).

## Create and Drop Databases

This section covers:

- [Create a Database](#)
- [Drop a Database](#)

### Create a Database

You create databases with the SQL command CREATE DATABASE:

```
CREATE DATABASE name;
```

To create a database, you must be a superuser or have the special db\_admin privilege.

When you create a database, no other users have the right to use it. You must manage user privileges as follows:

- To grant users the right to use the new database, you must GRANT at least the CONNECT privilege on the database to the users or roles who will use it.
- To grant users the right to create tables in the new database, you must grant them at least the CREATE privilege on one of the schemas in the database.

The user who created the database is the owner of the new database. It is solely the privilege of the owner of a database to drop it later. Removing a database removes all the objects (e.g. tables) within it, even if the individual object has a different owner than the database owner.

You need to be connected to the database server to execute the CREATE DATABASE command. The first database is always created when Aster Database is initialized. This default database is called “beehive.” To create the first “ordinary” database, you can connect to “beehive”.

### Database Name Limitations

Database names must comply with these rules:

- A database name may contain only alphanumeric characters and the underscore character (A-Z, a-z, 0-9, and \_).
- A database name may be up to 50 bytes long.

- The name must be allowed by PostgreSQL (i.e. it cannot be a PostgreSQL keyword).
- The name must not start with the prefix "`_bee`" which is reserved for use in naming Aster Database system objects.
- The name cannot be a substring of the word '`database`'.

## Character Set Encoding

The character set encoding for a database can be specified using the WITH ENCODING parameter in the CREATE DATABASE statement:

```
WITH ENCODING = encoding
```

Encoding specifies the character set encoding to use for the new database. The only valid value is '`'utf8'`', which represents UTF-8, a variable-length character encoding for Unicode. UTF-8 can represent any character in the Unicode standard. The initial encoding of byte codes and character assignments for UTF-8 is backwards compatible with ASCII.

## Drop a Database

Databases are permanently destroyed with the command DROP DATABASE:

```
DROP DATABASE name;
```

Only the owner of the database can drop a database. Dropping a database removes all objects contained within the database. The destruction of a database cannot be undone.

You cannot execute the DROP DATABASE command while connected to the database you are attempting to destroy. However, you can drop a database you own while you are connected to any other database.

# Manage Tables

This section covers:

- [Table Design Guidelines](#)
- [Create Tables Examples](#)
- [Load Data Into Tables](#)
- [Alter Tables](#)
- [Drop Tables](#)
- [Export Table Contents](#)

## Table Design Guidelines

This section introduces the concepts of fact and dimension tables and offers tips for creating an efficient data model in Aster Database.

## Table Types: FACT and DIMENSION Tables

Two types of tables are supported in Aster Database: FACT and DIMENSION:

- **Fact tables** are usually very large (e.g. millions or billions of rows). These tables contain two types of columns: the columns that contain facts and the columns that refer to the dimension tables. Fact tables require a distribution key column to be declared.
- **Dimension tables** are usually much smaller (e.g. tens to thousands of rows) than fact tables. Each dimension table specifies a set of known descriptive values for a particular dimension. For example, a “customers” table can be a dimension table that contains detailed information about each customer: for example, a customer ID, name, address, and phone number. A distribution key column is optional for dimension tables.

Data warehouses are frequently built on a “star schema” design. The star analogy refers to the typical data model where a central fact table is surrounded by multiple dimension tables, with the result that the schema diagram has a “star-like” shape. There are tables that record facts and other tables that record the dimensions that describe various facts. Fact tables provide granular information (e.g. performance metrics) of a business process, while dimensions provide more descriptive attributes about facts. By joining facts with dimensions, queries can expose deeper insights into the granular fact data.

For more information, see [“Fact Tables and Dimension Tables” on page 21](#).

## A Star Schema Example

For example, consider a popular website. To better understand the behavior of its customers, the company that owns the site may keep track of all the clicks on the site. In a star schema, each click event is a fact; the user who performed the click, the time of the day of the click, the page on which the click occurred, and so on are all dimensions of the fact. Thus an initial design for the tables in the website would be:

```
CREATE DIMENSION TABLE users (
    userid int,
    u_name varchar,
    u_gender char,
    u_zipcode int,
    ...
)
DISTRIBUTE BY HASH(userid);

CREATE DIMENSION TABLE pages(
    p_id int,
    p_category int,
    p_name varchar,
    ...
);
CREATE FACT TABLE clicks(
    userid int,
    pageid int,
    ts timestamp,
    ...
)
DISTRIBUTE BY HASH(userid);
```

The above examples define a distribution key for the tables; the next section describes how to decide which column to use as the distribution key.

## Distribution Key

When creating a table, choosing a column as the distribution key allows Aster Database to optimize the placement of rows in the cluster. In general, the distribution key should be based on the most interesting or frequent column to join.

For example, consider again the website we discussed in the preceding section. In most cases, it is interesting to understand the behavior of users on the site. For this reason, it makes the most sense to use the “userid” column as the distribution key of the clicks table.

For small dimension tables, it can be best to not define a distribution key at all. In this case, Aster Database will store the rows of the table redundantly at several nodes (via replication). While storing the rows redundantly increases storage requirements, it can make some queries more efficient because those rows will never need to be transferred to another node during query processing (local joins). See also “[Table Distribution](#)” on page 31.

## Normalization

For best performance, tables in Aster Database should be designed as a star schema or even more normalized form. Tables in Aster Database should not be heavily denormalized, as join operations (even complex ones) perform very well in Teradata Aster’s MPP architecture.

## Compression

Aster Database allows you to compress tables to four degrees: HIGH; MEDIUM; LOW; and none. You can compress using the COMPRESS keyword when you create it with CREATE TABLE, or later using the COMPRESS action of ALTER TABLE. Compression is not supported for temporary tables.

You can check the current compression level of a table by doing a `\d tablename` on the table in ACT. You can view the total amount of compressed data in the AMC by clicking the Dashboard Tab and looking in the **Nodes > Total Data Stored** section. You can see a node-by-node breakdown in the **Nodes > Node Overview** tab. Note that if you see **N/A** displayed for any items under **Total Data Stored** and **Disk Capacity**, this is a temporary status that is displayed while the data is being computed.

When choosing a compression level for a given table, take into consideration how often the data in the table will be accessed:

- For hot data, which is accessed frequently, Teradata Aster recommends that you use uncompressed or LOW compressed tables. This ensures maximum performance (fast access times) for queries against hot data.
- For cold data, which is used less frequently, Teradata Aster recommends LOW or MEDIUM compression in Aster Database. Applications or analysts using archived data should be made aware of the cost of decompression which may result in slower access times when querying such data.

The steps to compress a table are shown in ALTER TABLE and CREATE TABLE in the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.

## Logical Partitioning

Aster Database supports logical partitioning, also known as “automatic logical partitioning,” which is a useful feature in designing tables that contain data that is naturally segmented.

When tables are declared using a partitioning hierarchy, Aster Database queries may be routed to the data relevant to the query, avoiding unnecessary disk and network usage for data that does not affect the query.

For example, clicks data (Internet clickstream data) on a website are naturally segmented by time -- clicks that occur in January frequently are analyzed separately from clicks that occur in August. We can thus define our main “clicks” table as a partitioned table:

```
CREATE FACT TABLE clicks (
    userid int,
    pageid int,
    ts timestamp,
    ...)
DISTRIBUTE BY HASH(userid)
PARTITION BY RANGE(ts)
    PARTITION clicks_pre2011( END '2011-01-01'), -- everything pre-2011
    PARTITION clicks_201101( END '2011-02-01'),
    PARTITION clicks_201102( END '2011-03-01'),
    ...
    PARTITION clicks_201112( END '2012-01-01')
)
);
```

It is important when tables are created in a partition hierarchy that data is loaded only into the most specific tables, or the “leaves” of the hierarchy. For instance, in the example above, data should be loaded into “clicks\_201101” and not into “clicks.”

To perform a query, it is not necessary to consider the partition hierarchy. For instance, in the example above, a query should be directed to “clicks.” If the “WHERE” clause of the query considers only clicks in the months of January, February, and March, Aster Database will only access the partitions containing rows for those months and will efficiently avoid accessing tables for the other months of the year.

For more information, see [“Automatic Logical Partitioning” on page 35](#).

## Temporary Tables

Often it is necessary to create a table temporarily for the purpose of storing intermediate results. For best performance of Aster Database, these intermediate results should be stored in a temporary table. Compression is not supported for temporary tables. (See [“Example: Creating a Temporary Table” on page 156](#).)

## Create Tables Examples

- [Example: Creating a Table \(page 155\)](#)
- [Example: CREATE TABLE AS SELECT \(CTAS\) \(page 155\)](#)
- [Example: Creating a Temporary Table \(page 156\)](#)

### Example: Creating a Table

Empty tables are created using the CREATE TABLE command, with which you specify the table type (fact or dimension), table name, column names, column datatypes, constraints, and distribution key for a distributed fact or dimension table. For example:

```
CREATE DIMENSION TABLE regions (
    region_id INT NOT NULL,
    region_name VARCHAR NOT NULL );
```

Or consider this fact table:

```
CREATE FACT TABLE users (
    userid INT NOT NULL,
    name VARCHAR,
    email VARCHAR,
    age SMALLINT NOT NULL,
    favorite_color CHAR(5),
    region_id INT NOT NULL,
    CHECK (region_id>=0)
)
DISTRIBUTE BY HASH( userid );
```

### Example: CREATE TABLE AS SELECT (CTAS)

Tables can be created and populated in a single step using the CREATE TABLE AS SELECT command. Just as with the CREATE TABLE command, CREATE TABLE AS SELECT requires specification of the table type, table name, column information, constraints and optionally the distribution key. The command also specifies a SELECT query that populates the table. Note that indexes do not get recreated automatically on the new CTAS table.

For example, suppose we want to create a table of young users of age 25 and under from North America:

```
CREATE FACT TABLE users_young_northamerica (
    userid INT,
    name VARCHAR,
    email VARCHAR,
    age SMALLINT,
    favorite_color CHAR(5),
    region_id INT
)
DISTRIBUTE BY HASH( userid )
AS SELECT
    u.userid,
    u.name,
    u.email,
    u.age,
    u.favorite_color,
    u.region_id
FROM users u, regions r
WHERE u.region_id=r.region_id
AND u.age<=25 and r.region_name='North America';
```

Note: NULL, NOT NULL and CHECK constraints are not supported in CREATE TABLE AS SELECT statement.

## Example: Creating a Temporary Table

Dimension and fact tables that have not been subsequently dropped are committed when the enclosing transaction commits. Tables can also be explicitly designated as temporary tables, in which case upon transaction commit they are automatically dropped. Temporary tables are created using CREATE TEMP TABLE and CREATE TEMP TABLE AS SELECT commands, which have the same syntax as CREATE TABLE and CREATE TABLE AS SELECT commands. For example:

```
BEGIN;
CREATE TEMP FACT TABLE users_young_northamerica (
    userid INT,
    name VARCHAR,
    email VARCHAR,
    age SMALLINT,
    favorite_color CHAR(5),
    region_id INT
)
DISTRIBUTE BY HASH( userid )
AS SELECT
    u.userid,
    u.name,
    u.email,
    u.age,
    u.favorite_color,
    u.region_id
FROM users u, regions r
WHERE u.region_id=r.region_id
AND u.age<=25 and r.region_name='North America';
END;
```

Note: Table inheritance is not supported on temporary tables.

TEMP tables are very useful in a complex workflow. Since they can occur only within a transaction, there are some specific optimizations that make them faster to materialize and manage than permanent tables.

## Load Data Into Tables

There are a number of ways to load data into a database in Aster Database:

- [INSERT Statements \(page 156\)](#)
- Aster Loader Tool (see the *Aster Client Guide*)

### INSERT Statements

Consider again the example tables that were created in “[Example: Creating a Table](#)” on [page 155](#). In this section, we will take a closer look at the commands required to insert actual data rows into these tables.

The SQL statement INSERT lets you populate tables with rows. Usage examples that insert data into the two created tables ‘regions’ and ‘users’ are shown below:

```
INSERT INTO regions VALUES (1, 'Midwest');
INSERT INTO regions VALUES (2, 'Northeast');
INSERT INTO users VALUES (1, 'Michael West', 'michael@west.com', 38,
'blue', 2);
```

```
INSERT INTO users VALUES (2, 'Denise Smith', 'denise.smith@yahoo.com',  
23, 'red', 1);
```

All datatypes use rather obvious input formats. Constants that are not simple numeric values usually must be surrounded by single quotes ('), as in the example. The date type is actually quite flexible in what it accepts, but for these examples, we will stick to the unambiguous format shown here.

The VALUES clause in an INSERT statement is useful if you want to insert multiple rows in a single statement:

```
INSERT INTO regions VALUES (3, 'Southwest'), (4, 'Northwest');
```

The syntax used so far requires you to remember the order of the columns. Alternatively, you can list the columns explicitly, as shown below:

```
INSERT INTO users (userid, name, email, age, favorite_color, region_id)  
VALUES (3, 'Michael West', 'michael@west.com', 'green', 4);
```

You can list the columns in a different order if you wish, or even omit some columns, e.g., if the userid is unknown:

```
INSERT INTO users (name, email, userid, age, favorite_color, region_id)  
VALUES ('Michael West', 'michael@west.com', 4, 'green', 3);
```

Many developers consider explicitly listing the columns to be better style than relying on the order implicitly.

For more information on loading data, see the *Aster Client Guide*.

## Alter Tables

A table that exists must sometimes be modified – because of new data collection, for example. One option is to drop the table and recreate it, but this often is not desirable, particularly if the table already contains data. At these times, you should use the ALTER TABLE command.

You can use the ALTER TABLE command to make the following modifications:

- [Add columns](#)
- [Remove columns](#)
- [Add constraints](#)
- [Remove constraints](#)
- [Change datatypes of columns](#)
- [Rename columns](#)
- [Rename tables](#)

### Add columns

To add a column, use a command like this:

```
ALTER TABLE products ADD COLUMN description text;
```

The new column is initially filled with null values. You can define constraints on the column at the same time, using the typical syntax:

```
ALTER TABLE products ADD COLUMN description text
```

```
CHECK (description <> '') ;
```

In fact, anything you can do with the CREATE TABLE command can be done to an existing table with the ALTER TABLE command.

## Remove columns

To remove a column, use a command like this:

```
ALTER TABLE products DROP COLUMN description;
```

Whatever data was in the column disappears. Table constraints and serial column dependencies (if any) involving the column are dropped too.

However, if the column is referenced in a relationship with another table, Aster Database will not silently drop that constraint. You can authorize dropping everything that depends on the column by adding CASCADE to the command:

```
ALTER TABLE products DROP COLUMN description CASCADE;
```

## Add constraints

To add a constraint, the table constraint syntax is used. For example:

```
ALTER TABLE products ADD CHECK (name <> '') ;
```

You cannot add a NOT NULL constraint as a table constraint. However, you can add a NOT NULL constraint to a column with the following syntax:

```
ALTER TABLE products ALTER COLUMN product_no SET NOT NULL;
```

The constraint will be checked immediately, so the table data must satisfy the constraint before it can be added.

## Remove constraints

You need to know the name of a constraint to remove it. If you did not name the constraint when you created it, the system would have generated and assigned a name, which you need identify. The ACT command “\d tablename” can be helpful here. Other interfaces might also provide ways to inspect table details. Once you know the constraint’s name, the command to drop the constraint is:

```
ALTER TABLE products DROP CONSTRAINT some_name;
```

Note: As with dropping a column, you need to add CASCADE if you want to drop a constraint that something else depends on.

This works the same for all constraint types except NOT NULL constraints. To drop a NOT NULL constraint use:

```
ALTER TABLE products ALTER COLUMN product_no DROP NOT NULL;
```

Note: Recall that not-null constraints do not have names.

## Change datatypes of columns

To convert a column from one datatype to another, use a command like this:

```
ALTER TABLE products ALTER COLUMN price TYPE numeric(10,2);
```

In this example, you are altering the column “price” from its old datatype (e.g. integer) to the numeric datatype. This will succeed only if each existing entry in the column can be converted to the new type by an implicit cast. If a more complex conversion is needed, you can add a USING clause that specifies how to compute the new values from the old. For example,

```
ALTER TABLE products ALTER COLUMN price TYPE numeric(10,2)
USING price/2;
```

### Rename columns

To rename a column

```
ALTER TABLE products RENAME COLUMN product_no TO product_number;
```

### Rename tables

To rename a table

```
ALTER TABLE products RENAME TO items;
```

## Drop Tables

The DROP TABLE command is to be used in dropping the table along with its data. If only the data is required to be removed, consider using the DELETE command instead.

To drop a table, use

```
DROP TABLE products;
```

Note: Only the owner of a table can drop that table.

The above command would fail if there are other objects that depend on the products table. In such cases, if these objects are also to be dropped, specify the CASCADE modifier at the end of the DROP TABLE command:

```
DROP TABLE products CASCADE;
```

Dropping a table requires the table to exist. Otherwise, the DROP TABLE command will trigger an error. However, there may be times when you are not certain whether a table exists but must drop the table if it does exist. For example, in an automated script, you must ensure that a specific table does not exist and if it exists you must drop it. In such cases, consider using the following variant:

```
DROP TABLE IF EXISTS products;
```

If the table “products” exists, it is dropped. If it does not exist, nothing happens. In either case, the command returns successfully. Note that the IF EXISTS variant does not prevent other forms of errors, such as permission violation, object dependencies, etc.

## Export Table Contents

SELECT statements allow you to do complex operations on tables. Sometimes, however, you may need to export as-is the base data residing in a table without performing any operations on it. For this purpose, a COPY command is recommended:

```
COPY tablename [ ( column [, ...] ) ] TO stdout;
```

This selects out all data – e.g. the entire table or specifically named columns – of the table from each of the worker nodes and sends it on the client connection directly without materializing on the queen node. Thus, the speed of this command is usually limited only by the network bandwidth of the connection between the client and the queen node in Aster Database.

Only a table created via Aster Database can be exported using the COPY command above.

Note: Exporting the results of a query using the above command is not supported. However, you can work around this limitation by creating a separate table (e.g. via CREATE TABLE AS SELECT) and then exporting this table.

The COPY command described above streams data as and when it is available. If the user desires to control the streaming, then the cursor functionality can be used.

A cursor can be defined over the table required to be exported using the DECLARE command. In fact, this command accepts any Aster Database-valid SELECT statement. Once a cursor is declared, then as many rows as required can be obtained from the cursor using the FETCH command. Once done with the data export, the cursor can be closed using the CLOSE command. Here is an example:

```
BEGIN WORK;

-- Set up a cursor:
DECLARE liahona SCROLL CURSOR FOR SELECT * FROM films;

-- Fetch the first 5 rows in the cursor liahona:
FETCH FORWARD 5 FROM liahona;

code | title | did | date_prod | kind | len
-----+-----+-----+-----+-----+-----+
BL101 | The Third Man | 101 | 1949-12-23 | Drama | 01:44
BL102 | The African Queen | 101 | 1951-08-11 | Romantic | 01:43
JL201 | Une Femme est une Femme | 102 | 1961-03-12 | Romantic | 01:25
P_301 | Vertigo | 103 | 1958-11-14 | Action | 02:08
P_302 | Becket | 103 | 1964-02-03 | Drama | 02:28

-- Close the cursor and end the transaction:
CLOSE liahona;
COMMIT;
```

The cursor functionality is server-side and operates by obtaining the data from the worker nodes to the queen node.

**Best Practice:** For very large tables, it is recommended to use the COPY command. For smaller tables, a cursor or a SELECT statement can be used.

## Manage Database Objects

This section covers:

- [Browse database objects](#)
- [Analyze Tables](#)
- [Manage Integrity Constraints](#)

- [Manage Indexes](#)

See also: [Manage Space](#) and [Manage Tables](#).

## Browse database objects

To inspect the objects in your database, check your data dictionary as explained in “Data Dictionary Views” in the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.

## Analyze Tables

ANALYZE is critical for good performance when you are querying large tables. Aster Database’s default statistics are based on a table having few rows, but these statistics are inaccurate when a table contains a large number of rows, so the database query planner may choose suboptimal execution plans if you do not analyze the tables.

ANALYZE collects statistics about the contents of tables in the database, and stores the results in internal tables. The query planner then uses these statistics to help determine the most efficient execution plans for queries.

For example, suppose you have a single-table group-by aggregation query on a large table that you do not analyze. Based on the default statistics of the table having a few rows, the query planner would choose a hash aggregation plan. At run-time, the hash aggregate could run out of memory processing the large table and the query would fail. In contrast, if you run ANALYZE to generate accurate statistics, the query planner will choose a sort aggregation plan on the large table.

In principle, running ANALYZE after creating a small table with tens or hundreds of rows is not required, as Aster Database’s default statistics expect a small number of rows. However, the table is small, so you can run ANALYZE very fast, likely in less than a second.

In addition, Teradata Aster recommends running ANALYZE after modifying tables (CREATE TABLE AS SELECT, INSERT, UPDATE, DELETE), regardless whether the table size is large or small.

To run ANALYZE, type the following command:

```
ANALYZE [ table [ (column [, ...] ) ] ] ;
```

where “table” is the name of a specific table to analyze and “column” is the name of a specific column. The default is all columns.

With no parameter, ANALYZE examines every table in the current database. With a parameter, ANALYZE examines only that table. It is further possible to give a list of column names, in which case only the statistics for those columns are collected.

## Manage Integrity Constraints

### Check Constraints

Note that this section only applies to [“Parent/Child Tables” on page 386](#). Beginning with Aster Database release version 4.6, there is a newer, preferred method of partitioning, [“Automatic](#)

[Logical Partitioning](#) on page 35. It is included here for backwards compatibility with parent/child tables created in previous versions of Aster Database.

CHECK constraints enforce data validity constraints and are used in tables involving inheritance to intelligently filter out irrelevant data.

For example, suppose for data quality assurance purposes you want to ensure that user age is in the range 0-150, and user's favorite color must be red, green, or blue. You would define the user table like this:

```
CREATE FACT TABLE users (
    userid INT NOT NULL,
    name VARCHAR NOT NULL,
    email VARCHAR NOT NULL,
    age SMALLINT NOT NULL,
    favorite_color CHAR(5) NOT NULL,
    CHECK ( age >=0
        and age <= 150
        and favorite_color in ('red','green','blue') ) )
DISTRIBUTE BY HASH( userid );
```

CHECK constraints and inheritance are often used together to logically partition a large table based on date. Data is stored in child tables, with one child table per range of dates you wish to store together. The CHECK constraint ensures only the appropriate rows are inserted into each child table. For example, suppose you want to partition a large pageview table by day. (This example could also be extended to partitioning by week, month, etc):

```
CREATE FACT TABLE pageviews (
    userid INT NOT NULL,
    cookieid BIGINT NOT NULL,
    sessionid BIGINT NOT NULL,
    ts TIMESTAMP NOT NULL,
    url VARCHAR NOT NULL
)
DISTRIBUTE BY HASH( cookieid );
CREATE FACT TABLE pageviews_20080801 () INHERITS (pageviews);
ALTER TABLE pageviews_20080801 ADD CHECK (ts >= '2008-08-01 00:00:00'
    and ts < '2008-08-02 00:00:00');
CREATE FACT TABLE pageviews_20080802 () INHERITS (pageviews);
ALTER TABLE pageviews_20080802 ADD CHECK (ts >= '2008-08-02 00:00:00'
    and ts < '2008-08-03 00:00:00');
```



**Notice!** Aster Database does not detect overlapping constraints on peer child tables. As a result, the correct placement of a row during loading can be indeterminate.

**Workaround:** Take care that the constraints you define do not create overlapping logical partitions. A simple mistake would be to set up range constraints like this:

```
CHECK ( ymdh BETWEEN '2005-07-01' AND '2005-08-01' );
CHECK ( ymdh BETWEEN '2005-08-01' AND '2005-09-01' );
In this example, it is not clear in which partition the ymdh value '2005-08-01' resides.
```

---

Suppose we need to write a report on the first seven days of August pageview data. The SQL statement would look like:

```
SELECT ...
    FROM pageviews pv ...
    WHERE pv.ts >= '2008-08-01 00:00:00'
```

```
and pv.ts < '2008-08-08 00:00:00'
```

The Aster Database system would intelligently determine that only the seven tables pageviews\_20080801 through pageviews\_20080807 are required for this query and ignore pageviews data for other days.

Currently the intelligent data exclusion mechanism only supports constant values. For example, the following statement contains a data formatting function (to\_timestamp) and would not result in data exclusions. Instead it would result in all pageviews data being accessed:

```
SELECT ...
  FROM pageviews pv ...
 WHERE pv.ts >= to_timestamp(1217574000)
   and pv.ts < to_timestamp(1218178800) ...
```

As a workaround, you should convert the range filter predicate to constant values first, and substitute the values in the query:

```
beehive=> select to_timestamp(1217574000)::timestamp without time zone
as a ;
a
-----
2008-08-01 00:00:00
(1 row)

beehive=> select to_timestamp(1218178800)::timestamp without time zone
as b ;
b
-----
2008-08-08 00:00:00
(1 row)
```

Substitute constant values back into the intended query:

```
SELECT ...
  FROM pageviews pv ...
 WHERE pv.ts >= '2008-08-01 00:00:00'
   and pv.ts < '2008-08-08 00:00:00'
```

## NOT NULL Constraint

The NOT NULL constraint enforces that a column has some value other than “null.” For example, in a users table, suppose we want to enforce that we always have the user id and user age information:

```
CREATE FACT TABLE users (
  userid INT NOT NULL,
  name VARCHAR,
  email VARCHAR,
  age SMALLINT NOT NULL,
  favorite_color CHAR(5)
)
DISTRIBUTE BY HASH( userid );
```

## PRIMARY KEY Constraint

Aster Database supports defining a primary key (PK) for tables in which you wish to specify a column or combination of columns whose values establish a unique record. (Note that, for tables that contain a large amount of data, PKs are typically not used.)

The PRIMARY KEY constraint specifies one or more columns whose values uniquely identify a row. The PRIMARY KEY constraint also serves as a UNIQUE and a NOT NULL constraint on the involved columns.

For example, suppose `cookieid` and `ts` uniquely identify a record in a `pageviews` table:

```
CREATE FACT TABLE pageviews (
    userid INT NOT NULL,
    cookieid BIGINT NOT NULL,
    sessionid BIGINT NOT NULL,
    ts TIMESTAMP NOT NULL,
    url VARCHAR NOT NULL,
    PRIMARY KEY (cookieid, ts) )
DISTRIBUTE BY HASH(cookieid);
```



**Important!** For distributed fact and dimension tables, the column you choose as the distribution key must be one of the columns of the primary key. See "[Rules for distribution keys](#)" on page 32.

## Manage Indexes

For distributed tables, Aster Database supports local indexes that index the rows on the local vworker. A global index (on a distributed table) that spans all vworkers would be prohibitively expensive to maintain during loads, and is typically not used in data warehousing environments.

### Create Indexes

Indexes are created using the `CREATE INDEX` command, specifying the name of the index, the underlying table to index, and the column(s) to index. For specific SQL syntax on indexes, please refer to the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.

Indexes are useful for quickly accessing selective amounts of data. For example, suppose we want to filter a table called “`pageviews`” by a highly selective criteria: the `sessionid` in (11, 12). Then an index on the `sessionid` column of the `pageviews` table would speed up this query. In contrast, suppose we have a non-selective criteria: find the average age of all users in the North America region. Then a sequential scan is likely more efficient than index-based access which results in many random I/Os.

Having an index also adds index maintenance overhead on subsequent `INSERTs` and `UPDATEs`. In general, indexes should be created after the underlying table has been populated.

### Indexing Example: A Geographic-IP Address Join

A scenario where indexes are very helpful is joining a large table of values (“`netflow`” in this example) to a small table of ranges. In this example we join the network traffic flow table (“`netflow`,” a table of source and destination IP addresses) to a geolocation table (“`na_all`”)

that maps IP ranges to geographic locations. Suppose we have the following tables and a few sample rows as shown below:

```
BEGIN;

CREATE TABLE netflow
(
    srcip ip4,
    dstip ip4
)
DISTRIBUTE BY HASH(srcip);

CREATE DIMENSION TABLE na_all
(
    iplow ip4,
    iphigh ip4,
    ctry varchar
);

INSERT INTO na_all
VALUES
('79.43.226.6','79.43.255.255','usa'),
('10.24.35.45','10.24.35.48','india'),
('10.0.0.0','10.24.35.45','afghanistan'),
('233.3.45.44','233.3.45.255','afghanistan'),
(NULL,'233.3.45.255','afghanistan');

INSERT INTO netflow
VALUES
('79.43.228.96','10.24.3.5'),
('233.3.45.200','79.43.226.7');
```

The following statements create a GiST index on the iplow and iphigh columns of the netflow table, and join the two tables together:

```
CREATE INDEX rangeindex
    ON na_all
    USING gist (ip4range(iplow, iphigh));

SELECT na1.ctry
    FROM netflow, na_all na1, na_all na2
    WHERE
        ip4range(na1.iplow, na1.iphigh) >= srcip
        AND
        ip4range(na2.iplow, na2.iphigh) >= dstip
    ORDER BY na1.ctry;

ABORT;
```

## Guidelines for Indexes

Teradata Aster recommends the following guidelines for vworker-local indexes. (Note: Aster Database does not support cluster-global indexes. Instead, indexes are per-vworker, which means each index includes all rows stored on that vworker.)

1 Create indexes after inserting data:

It is more efficient to create indexes on fact tables after data has been loaded. If done the other way, i.e., if an index exists before the load begins, the database will need to maintain the index for every inserted row, slowing loads tremendously.

2 Recognize when indexes are appropriate, such as in cases like these:

- Low row selection: If a workload has queries that frequently access less than 10-15% of the rows in a large table, an index might be appropriate. Of course, such a percentage value depends largely on the relative speed of table-scan and the distribution of the row data in relation to the order of the index key. The faster the table-scan, the lower the above percentage; the more clustered the row data, the higher the percentage.
- JOINS on multiple tables: Performance of JOINS across multiple tables could improve with indexes, as the execution plan avoids sequential scans of all tables in the JOIN.
- Suitability of columns for indexing: If a column contains many NULLs, and the workload has frequent queries that access the non-NULL values, an index might be appropriate.

3 Order index columns for performance:

When creating an index with a composite (multi-column) key, the order of the columns should be based on the general rule that the most frequently occurring columns in queries should be placed first. For example, an index over columns  $\langle c1, c2, c3 \rangle$  will be used by queries that access either  $c1$ ,  $c1$  and  $c2$ , or  $c1$  and  $c2$  and  $c3$ . Queries that access  $c2$ ,  $c3$ , or  $c2$  and  $c3$  will not leverage the index.

4 Limit the number of indexes for each table:

An index will speed up SELECTs, but will slow down DMLs such as INSERTs, DELETEs, and UPDATEs, because index maintenance is a per-row operation and entails random I/O. This trade-off between SELECTs and DMLs must be kept in mind when deciding how many indexes to create for a table. Tables that are primarily read-only will benefit from indexes. Tables that get modified very often will do better with fewer indexes.

5 Drop indexes that are no longer required:

If you suspect that there may be some indexes that are no longer being used:

- a Run ANALYZE if it is possible that the statistics for the table in question may have changed substantially since the most recent ANALYZE.
- b Periodically look at query execution plans to inspect whether indexes are used in queries anymore. Because of the changing nature of data and workloads, it is possible that plans that once used indexes are no longer using them. It is better to drop such indexes because they unnecessarily slow down DMLs on the associated tables.

## Altering Indexes

Indexes can be renamed using the ALTER INDEX command, specifying the old index name and the new index name.

For example, we can rename the index from the previous subsection:

```
ALTER INDEX user_ips_ip RENAME TO my_index_name;
```

# Manage Space

## Check Current Disk Capacity and Free Space

- The AMC's Dashboard > Nodes section and the AMC's Nodes > Node Overview tab show the amounts of available storage in the cluster and on individual nodes, respectively. See the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide* for more information.

## Disk Usage

When provisioning storage in Aster Database, there are a few things to keep in mind:

- 1 Data loaded into Aster Database may be stored in a different format than the format used during loading. In some cases, the storage used may actually increase, while in other cases it may decrease, especially if stored in a compressed table in Aster Database.
- 2 Data is replicated in Aster Database to tolerate failures, so you should consider these replications when provisioning an Aster Database. Using the recommended replication factor of two, all data is stored twice in the cluster.
- 3 Because Aster Database is comprised of multiple nodes, the data placement algorithms in Aster Database distribute data across the nodes to optimize parallel performance of the system. The amount of available storage shown for the entire system can often be used as a rough estimator of the amount of available storage at an individual node. For more precise information about used and available storage at a particular node, see the **Nodes > Node Overview** tab in the AMC.

### **nc\_relationstats function**

This section summarizes the features of the nc\_relationstats function, which is useful in determining disk usage. For detailed information on this function see “[“nc\\_relationstats” on page 210](#)”.

**On disk size:** The nc\_relationstats function reports the on disk size of the table. The table may be compressed or uncompressed. The size includes the TOAST sizes. Note that we only report the size of a table on each vworker, not on the whole cluster. If a user wants to find the total fact table size on the cluster, he needs to perform a “Group By”.

**Tuple counts:** This function returns the current active number of tuples in the input relations. Since an exact count is expensive, the user is given an option if an estimate of the value is acceptable. The number of dead tuples in the table is reported separately. The number of tuples with TOAST storage is not included in this count value. Again remember that for a fact table, the tuple count for individual vworkers is given.

**TOAST (The Oversized-Attribute Storage Technique):** If a tuple size exceeds the size of a page, the variable length attributes are compressed and/or broken up into multiple physical

rows. These extra rows are stored separately. The TOAST process happens transparently to the user.

**Diagnosability and Serviceability Goals:** The table size feature is very important for the cluster administrator to understand how the tables are growing. An administrator can perform a VACUUM if the dead tuple count is too high. A decision on adding extra disks or nodes can be taken based on the growth of the tables. Even partition skew can be observed from the partition size statistics.

## Maintain Sufficient Free Disk Space

To ensure proper operation of the cluster, you must maintain at least 30% free space at all times. It is important to keep some storage unused and available for normal operation of Aster Database. Aster Database temporarily requires the use of unused storage for certain operations, including large aggregation and sorting operations, creation of temporary tables, replication of data, and so on.

If the disk usage on Aster Database or on any node exceeds 70%, you must take action immediately to add nodes, reclaim dead disk space, or delete data from your cluster. Do one of the following:

- To add nodes, read the section “Adding Nodes to Aster Database” in the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*; or
- To reclaim space or delete data, see the section, “[Dead Space](#)”, below.
- If you suspect data skew is filling one or more nodes, see “Detecting and Managing Data Skew” in the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*.

If one of the disks on a worker node becomes full (> 90% capacity). Aster Database will shut down. This protects the system from data loss.

Follow these recommended best practices to ensure system health related to disk full conditions:

- 1 Set up alerts to warn administrators of impending disk full conditions.
- 2 Always stay aware of disk usage on all nodes of the cluster.

## Dead Space

### Operations That Create Dead Space

In Aster Database, the following SQL operations result in dead space being created on disk in the data files for a given database table:

- DELETE,
- UPDATE,
- aborted INSERT,
- aborted COPY,
- aborted bulk loading.

Dead space occurs when data rows are marked invisible but the space they take up is not compacted or reused. For example, SQL DELETE command is executed by marking all qualifying rows as invisible. The SQL UPDATE command, for example, operates as follows: When you update a row, the existing row is marked as invisible, and the updated row is appended at the end of the table's data file. Dead space, such as the invisible row in this example, is not automatically marked to be reclaimed or compacted. Reclaiming such space requires the administrator to run specific commands, which we will discuss below.

### Effects of Too Much Dead Space

It is important to be vigilant about dead space and proactively reuse or compact dead space. Even though dead space does not contain live data for a table, it affects your cluster in these ways:

- Too much dead space may result in node failures: Excessive dead space may result in a full disk on a worker node.
- Too much dead space may result in slow query/commit performance: A sequential scan on a table requires scanning through all data files for the table, including the dead space. Cluster-wide replication at transaction commit time is performed at the file level, so dead space also needs to be replicated over the network.

### Calculate the Amount of Dead Space in the Cluster

Dead tuples that are no longer in use (either due to a DELETE or an UPDATE) are not physically deleted. This dead tuple space is only reclaimed after a VACUUM operation. In order to decide if you need to perform a VACUUM operation, first determine the exact dead tuple count.

Use “[nc\\_relationstats](#)” on page 210 to perform this calculation. The nc\_relationstats section contains a list of “[Example nc\\_relationstats Calls](#)” on page 216 that contain information on how to find the exact dead tuple count.

This is one such example:

```
SELECT schema, relation, sum(dead_tuple_count) as dead_tuples,
sum(tuple_count) as total_tuples
  FROM nc_relationstats(
    ON (SELECT 1)
    PARTITION BY 1
    DATABASES ('niray')
    RELATION ('schema."TblName"')
    REPORT_STATS('tuple_count')
    REPORT_STATS_MODE('exact')
  )
 GROUP BY schema, relation;
```

### Remove Dead Space

Once we have identified the tables with the dead space, there are several approaches to reuse or reclaim the dead space. The approaches are:

- “[“TRUNCATE \[tablename\];”](#), below
- “[“VACUUM ANALYZE \[table\\_name\];”](#) on page 170,

- “[VACUUM FULL ANALYZE \[table\\_name\];](#)” on page 170, and
- “[CREATE TABLE ... AS SELECT ...](#)” on page 171



**Tip!** Your indexes may also be consuming too much space if they contain dead index pages. This can occur with B-tree indexes in Aster Database under certain access patterns. To free this space, use the `REINDEX` command, which reclaims space by rewriting a new version of the index without the dead pages.

#### ***TRUNCATE [tablename];***

The TRUNCATE command quickly deletes all live data rows and removes all dead/free space from a table. If you have a table that has zero live rows and a lot of dead or free space, it is preferable to use the TRUNCATE command to remove the dead/free space.

#### ***VACUUM ANALYZE [table\_name];***

The SQL VACUUM command sequentially scans a table and marks dead space as free space, which are available for future INSERT/COPY. VACUUM requires a shared lock on the table, and hence allows concurrent activities on this table. VACUUM runtime is slightly more than the time it takes to sequentially scan the table.

The key caveat is that the dead space is not actually compacted. For fact tables like event data that are typically loaded once (e.g., per day), they are usually never loaded again. In such a scenario, without a subsequent INSERT/load, the free space converted by VACUUM command will never be reused. The VACUUM command is useful in scenarios like slowly changing dimensions, where we have a constant stream of appends from INSERT and UPDATE that could potentially reuse the free space. For slowly changing dimensions, Teradata Aster commonly sees customers run upsert (INSERT/UPDATE) followed by VACUUM as a daily batch job.

Note that in scenarios like slowly changing dimensions where VACUUM is applicable, the converted free space is not always optimally reused. The free space reuse depends on heuristics such as whether blocks of new rows fit in contiguous blocks of free space. The administrator should regularly monitor the growth of dead space and free space for tables affected by dead space. If we observe that with daily VACUUM, dead space stays near zero but free space is still steadily growing, then we need to also implement one of the approaches below on a regular basis (e.g., weekly VACUUM FULL ANALYZE or CTAS).

#### ***VACUUM FULL ANALYZE [table\_name];***

The SQL VACUUM FULL command physically rearranges on disk pages of a given table to compact the dead/free space. VACUUM FULL requires an exclusive lock on the table.

**WARNING:** This operation is very resource intensive. For large tables we recommend using the next approach (CTAS) instead, which runs much faster on large tables.

Note for VACUUM and VACUUM FULL, we have ANALYZE in the command. That option means we will update table statistics while the VACUUM/VACUUM FULL runs. Also VACUUM/VACUUM FULL must be executed outside of transactions.

VACUUM FULL is an optional feature of Aster Database and is not enabled by default.

### ***CREATE TABLE ... AS SELECT ...***

The CREATE TABLE AS SELECT (CTAS) approach copies the contents of the table to a new table, DROPS the original table, and renames the new table to the original table's name.

Note that indexes do not get created automatically on the new CTAS table. For a very large table, the additional time required to recreate the indexes may be far greater than the savings gained by using CREATE TABLE versus VACUUM.

For example:

```
BEGIN;
CREATE FACT TABLE test_table_new (col1, col2, ... )
DISTRIBUTE BY HASH(...)
AS
SELECT col1, col2, ...
FROM test_table;

ANALYZE test_table_new;

-- Sanity Check
SELECT COUNT(*) AS test_table_old FROM test_table;
SELECT COUNT(*) AS test_table_new FROM test_table_new;

-- Replace old table with new table
DROP TABLE test_table;
ALTER TABLE test_table_new RENAME TO test_table;

END;
```

Aster recommends the CTAS approach as it is much faster than VACUUM FULL. However note that if you use the CTAS approach, we will lose any table privileges that you had set up. The administrator would need to issue the SQL GRANT command to re-establish access privileges for all users of the table, execute the CREATE INDEX command to re-build all indexes, and re-establish parent-child inheritance if appropriate.

Note in addition that indexes do not get created automatically on the new CTAS table. For a very large table, the additional time required to recreate the indexes may be far greater than the savings gained by using CREATE TABLE versus VACUUM.

### **Verify That Dead Space Has Been Reclaimed**

Finally we can check if the dead space has been reclaimed using a query similar to the following. This example assumes we want to check that space associated with table, "test\_table", has been freed:

```
--Sanity Check
SELECT * FROM ncluster_storagestat('test_table');
```

Note that in the case of a dimension table that is distributed by replication, the tuple\_count returned reflects the total number of tuples on all workers.

# Compression

Aster Database features compression, which allows you to create compressed tables and save space. You can compress to varying degrees by including a HIGH, MEDIUM, or LOW constraint in your compression syntax. The syntax for creating compressed tables is:

```
CREATE [ FACT | DIMENSION ] TABLE table_name ( [
    { column_name data_type [ column_constraint [ ... ] ]
      | table_constraint }
    [, ... ]
] )
[ COMPRESS [ HIGH | MEDIUM | LOW ] ]
[ INHERITS ( parent_table ) ]
```

and, for CREATE TABLE AS SELECT, the syntax is:

```
CREATE [ FACT | DIMENSION ] TABLE table_name ( [
    { column_name data_type [ column_constraint [ ... ] ]
      | table_constraint }
    [, ... ]
] )
[COMPRESS [ HIGH | MEDIUM | LOW ] ]
AS SELECT STATEMENT
```

There is no change in query syntax for compressed tables. For all query purposes, a compressed table will be treated the same as a normal table. Compression is currently not supported for temporary tables. Compressed tables are replicated in their compressed form.

To alter the compression level of a table, use the following syntax:

```
ALTER TABLE table_name [ NOCOMPRESS
    | COMPRESS [ HIGH | MEDIUM | LOW ] ]
```

Before you alter existing table compression properties – compression levels, initial compression of a table, decompression of a table – you should ensure that there is sufficient disk space available for the operation.

Table compression occurs in an online fashion without disruption to Aster Database. One useful application of compression is to combine it with Teradata Aster's logical partitioning feature for information lifecycle management. As you recall, logical partitioning enables creation of a hierarchy such that a large table can have partitions, which in turn can have their own partitions, and so on. If the child partitions are range-partitioned (e.g. monthly partitions), compression can be used to compress the monthly child partitions over time, as they become less frequently accessed.

For example, assume it is November. You may leave the October and November child partitions uncompressed as they are more frequently accessed. However, older data can be compressed at increasing levels since query frequency may drop as data gets stale. For example, Q3 data (July-Sept.) may be compressed LOW, Q2 data (April-June) may be compressed MEDIUM, and Q1 data (Jan.-Mar.) may be compressed HIGH.

Realized compression ratios depend on the compression level selected by the user and the data characteristics. While realized compression rates vary, typical ratios range from 3x to 12x.

# EXPLAIN Plan

Given an SQL query, Aster Database generates a series of steps that need to be executed to satisfy the query. EXPLAIN lets a user view this query plan, cost estimates associated with each phase, and estimates of how data is distributed across workers (if applicable).

## Example

Consider the following simple example, which shows how to interpret EXPLAIN output. We type

```
EXPLAIN SELECT * FROM explain_t1;
```

Figure 12: Output of a sample EXPLAIN query

| Number | Statement                             | Result Table | Operation Type             | Location |
|--------|---------------------------------------|--------------|----------------------------|----------|
| 1      | select explain_t1.a from explain_t1 ; | _tmp_0       | Pre-condition              | Workers  |
| 2      | select a from _tmp_0 ;                |              | Actual statement execution | Queen    |

Figure 13: Output of a sample EXPLAIN query, continued

| Query Plan and Estimates                                                                                       | Data Size Distribution (in bytes)   |
|----------------------------------------------------------------------------------------------------------------|-------------------------------------|
| alCost=0.00..34.00 rows=9600 width=4 networkCost=37<br>Scan on explain_t1 (cost=0.00..34.00 rows=2400 width=4) | mean size=9600 standard deviation=0 |
| alCost=0.00..97.00 rows=9600 width=4 networkCost=0.00<br>Scan on _tmp_0 (cost=0.00..97.00 rows=9600 width=4)   |                                     |

You read the high-level phases that Aster Database first generates from top to bottom. Each row in the EXPLAIN output is one such phase, and is essentially an SQL statement that gets executed at either the queen or worker nodes. The output columns are:

Number: This is the number of the high-level phase.

Statement: The actual statement being executed at the individual workers.

Result Table: If the statement being executed generates output that will be used later, this goes into the table name displayed in this column.

Operation Type: This column describes the kind of operation being performed. This can be one of the following:

- 1 Operation Type: Pre-condition
- 2 Operation Type: Actual statement execution
- 3 Operation Type: Repartition tuples <reason for repartition>  
(Column\_name(s)) -- This indicates that rows ("tuples") are being shuffled across different workers. The Column name(s) is (are) the column(s) on which data is being partitioned. Here are the exact reasons for repartitioning data:
  - Repartition tuples and populate table

- Repartition tuples for subselect
  - Repartition tuples for aggregation
  - Repartition tuples to satisfy join
- 4 Operation Type: Broadcast tuples <reason for broadcast> – This indicates that the result set of rows (“tuples”) is being sent to all the workers. The following are the cases that could cause broadcast of rows:
- Broadcast tuples and populate table
  - Broadcast tuples for subselect

Location: This column indicates where this phase is being executed. This can state either queen, Workers or AnyWorker.

Query Plan and Estimates: This column displays the low level query plan and cost estimates for the corresponding phase and is read bottom-up. Note that all costs are relative, and the cost of fetching 8K of sequential data from the disk is set to 1. The cost of transferring 1K of network data is set to 1. The first line of the output displays the summary of the entire query. Here is how to interpret the query plan and estimates in the EXPLAIN output for phase 1:

```
localCost=0.00..34.00 rows=9600 width=4 networkCost=37
```

The cost of getting the first row is 0. The cost to read all the rows is 34.00. The number of rows returned will be 9600. The average width of the row is 4, and the network cost to transfer these rows from the workers to the queen is 37.

From the second line onward, the output is from the node where the query is going to be executed. In case the query executes at multiple workers, output from the slowest worker is displayed to the user. One can see which low-level algorithms will be applied at the nodes. In phase 1 for example, the slowest worker will perform a sequential scan on explain\_t1 to satisfy the query.

Data Size Distribution (in bytes): This column gives an estimate on the mean and standard deviation of data coming from worker nodes. This is only applicable for queries that require transfer of data from either the worker nodes to the queen node or amongst the workers.

### **Questions to ask when looking at EXPLAIN output:**

Are any of the phases taking an inordinately long amount of time?

Is the network cost of the phases too high? The network often is the biggest bottleneck, so queries should be written to minimize network traffic.

Is the standard deviation of data really high? This could indicate a large data skew. This could be helpful in determining how data should be distributed across workers.

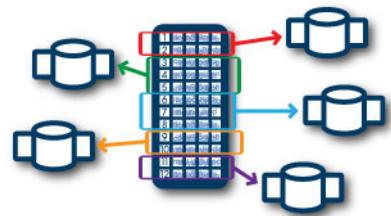
### **Update EXPLAIN Statistics**

EXPLAIN output generated for queries is only as accurate as the table statistics for tables being queried or modified. Thus it is important that you periodically ANALYZE tables.



## CHAPTER 2 Data Modeling Best Practices

Aster Database is designed to let you perform fast analysis on large data sets, and just as importantly, Aster Database is designed to perform well in environments where analysts frequently run *ad hoc* queries in an effort to derive new insights from their data. In many environments with traditional database systems, analysts are accustomed to writing a query, running it overnight, then tweaking it based on the results, running the new query, and waiting for hours for the new results. Using a properly configured Aster Database, this cycle of query iteration can be shortened to minutes or seconds, rather than hours.



To achieve these results, though, you must take the time to properly design your data model, so that it suits the characteristics of your data and your queries. This does *not* mean you are designing your data model around pre-canned queries! Instead, it means that you are providing Aster Database with clues as to which data should be collocated with which other data, and which data is likely to be queried more often, or used more often for joining or filtering results.

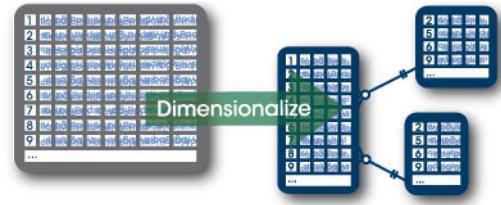
This section covers:

- [Eight Fundamental Rules for Modeling Big Data in Aster Database](#)
- [Introduction to Data Modeling in Aster Database](#)
- [Dimensionalize Your Schema](#)
- [Use Columnar Tables When Appropriate](#)
- [Distribute Your Data with Joins in Mind](#)
- [Replicate Common, Frequently Joined Data](#)
- [Split Data Into Child Partitions](#)
- [Verticalize Your Schema](#)
- [Index Your Tables](#)
- [Consider Using a Denormalized Data Model](#)
- [Data Modeling FAQ](#)

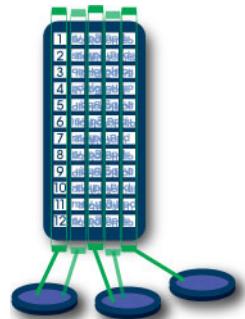
## Eight Fundamental Rules for Modeling Big Data in Aster Database

The most important rules to remember when building your data model for fast performance in Aster Database are:

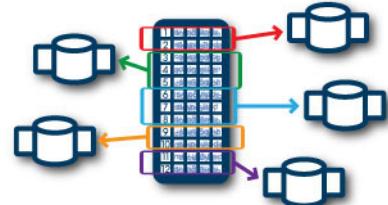
**Dimensionalize your schema:** Use a star schema (also known as a dimensionalized schema) to make your fact tables skinny. Skinny tables let your queries run faster, because they have less data to read. With a properly dimensionalized schema, queries can run up to 20x faster.



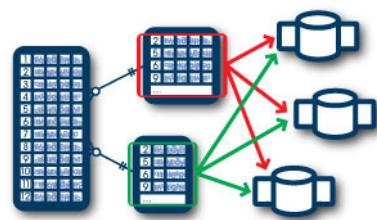
**Columnarize if you can:** If you are willing to use a table as an append-only table, then your queries may run faster if you create the table with a columnar storage layout, rather than the traditional row-wise layout.



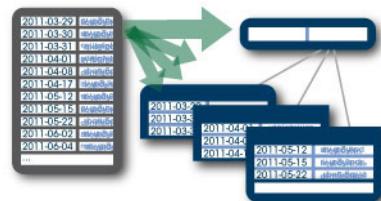
**Distribute your data with joins in mind:** Physically distribute large fact and dimension tables by selecting a distribution key column that lets Aster Database distribute rows throughout the cluster but still enables your most common, costly operations (joins, DISTINCT computations, GROUP BY computations) to be performed locally on individual workers in the cluster. With properly distributed data, queries that make use of the distribution key can run as much as 100x faster.



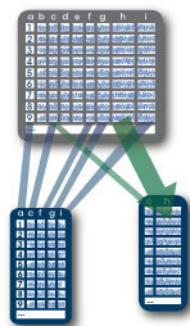
**Replicate frequently joined rows:** Use replicated dimension tables in Aster Database to ensure that a copy of each frequently joined dimension table is always present on the local machine for joins. By using replicated dimension tables, you eliminate the need to move dimension data over the network, increasing query performance by as much as 4x.



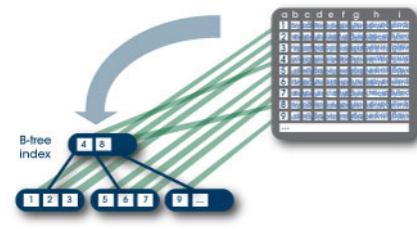
**Split data to child partitions:** Use logical partitioning to give the effect of smaller fact tables, which shortens query runtimes and simplifies management of the data over time. By using logical partitioning, queries will improve in performance with the number of partitions added.



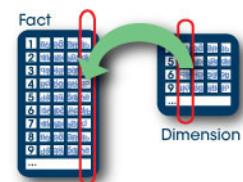
**Verticalize your schema:** Make your fact tables narrow by creating relevant materialized projections – that is, duplicate tables that contain only the frequently queried-together columns. By verticalizing your schema, queries on the projections can run 2x to 3x faster than they would run on a wide fact table.



**Index your tables:** Create relevant indexes on your tables to make retrieving selective information fast. Indexes allow more targeted lookups and prevent irrelevant data from being read and processed. Queries that took hours to run on a non-indexed table can run just a few seconds on a properly indexed table, if the query is highly selective.



**Consider using a denormalized data model:** Denormalizing, or allowing some columns to be duplicated in your model, lets you make your most frequently queried table wide enough (but remember to make it only *just* wide enough!) to include all your most frequently queried columns, reducing the need to perform joins.



In this document, we'll discuss these techniques in detail and explain why they're useful for improving query performance. But first, let's talk about why you should care about these techniques – that is, let's examine how Aster Database is different from other database systems, why in Aster Database you need to apply new techniques to make your queries run fast, and set some ground rules for working with large amounts of data.

# Introduction to Data Modeling in Aster Database

## Distributed and Replicated Tables

Aster Database's built-in support for fact and dimension tables helps your queries run fast on star schema databases, even if the databases hold a very large amount of data. Specifically, Aster Database *distributes* a fact table's rows across all workers, and it typically *replicates* the contents of each dimension table to every vworker. For instructions on creating fact and dimension tables, see “[Tables](#)” on page 20.

When you create a fact table, you specify how its rows will be distributed by designating one of the table's columns as the distribution key column. As each row is loaded, it will be sent to a particular vworker on a particular worker (physical machine) in the cluster based on the value it has in this column. Later, when you query the fact table, Aster Database handles the lookup needed to retrieve rows from the various workers throughout the cluster.



**Tip!** In pre-4.6 versions of Aster Database, we called this the “partition key,” which made it easy to confuse this with the separate Aster Database concept of *logical partitioning*. In 4.6 and later, it's called a *distribution key*, and you declare it with the `DISTRIBUTE BY` clause in your `CREATE TABLE` statement.

When querying the database, you will frequently join the fact table with one or more dimension tables, since the dimension tables provide the context information you need in order to understand why or how a particular fact occurred. Joins like this can be challenging in MPP databases like Aster Database, since the large dimension table is distributed across the nodes of the cluster, and the corresponding rows of the dimension table being joined to might not be present on the same machine as the fact rows.

Aster Database's default treatment of dimension tables solves this problem by storing a complete copy of the dimension table on every vworker. This is possible for most dimension tables because they are usually much smaller than fact tables (a row in a dimension table typically stores dimension information that corresponds to many rows of the fact table). The term for such a table is “replicated dimension table.” By replicating the dimension table, Aster Database ensures that joins of the fact and dimension tables run quickly, since the set of fact rows on a given node will always find all its corresponding dimension rows on the same node. In other words, more joins can be done locally on each vworker.

## Aster Database is Different

Aster Database is *not at all the same* as traditional, single-server database systems such as Oracle or SQL Server, and Aster Database is *also not similar* to other MPP, shared nothing systems that you may have used in the past. Aster Database has been optimized to provide extremely fast ad-hoc query performance and analytical function execution. For ad-hoc or iterative analysis of large amounts of data, Aster Database's performance is among the fastest available today. *However, your queries will only run fast if you have set up your data model properly to match the characteristics of your data set, queries, and analytic routines.*

As the data architect or administrator, you need to invest a small amount of time up front to choose the right table structures and to tell Aster Database how best to distribute your data in the cluster. You and your data analysts know which columns are queried and joined on most frequently, and you must configure Aster Database to distribute the data in a way that supports these queries. If you don't do this, your performance will be barely better than that of an un-tuned, single-box database system!

If you are not seeing the performance you want from Aster Database, it's a good idea to review the design of your data model and ensure you're following the fundamental rules spelled out in this document.

## Aster Database's Three Principles for High Performance on Big Data

Soon, we'll discuss in detail the rules that should guide your data model design, but first let's establish the basic principles that underlie these rules. Set aside the techniques you might have learned while tuning other database systems. To get the best performance from Aster Database, you need to keep in mind that the amount of data you'll be working with is very large. With big data, new rules apply:

### **Networking: Thou shalt not move big data.**

Corollary: If you need to move big data, make it small first, and then move small data.

### **Disk: Thou shalt not read irrelevant data.**

Corollary: Prepare the data model in advance to ensure that queries touch the least amount of data.

### **Processor: Thou shalt not do redundant processing.**

Corollary: Prepare your queries such that each computation is done exactly once, and never again.

## What You'll Learn in This Document: The Aster Database Way to Model Big Data

Aster Database is designed to support a number of techniques that let you run in-database analytics and ad-hoc queries on petabytes of data. Using these techniques, you will ensure that your queries and analytics run efficiently and fast. If you find that you see performance issues with Aster Database, review this document and your data model to identify improvements you can make. Many performance issues can be solved with proper data modeling.

## Review: Eight Fundamental Rules for Big Data

Recall the most important rules for data modeling that we stated at the beginning:

- 1 Dimensionalize your schema.
- 2 Use columnar storage if you can.
- 3 Distribute your data with joins in mind.
- 4 Replicate frequently joined rows.

- 5 Split data to child tables.
- 6 Verticalize your schema.
- 7 Index Your Tables.
- 8 Consider using a denormalized data model.

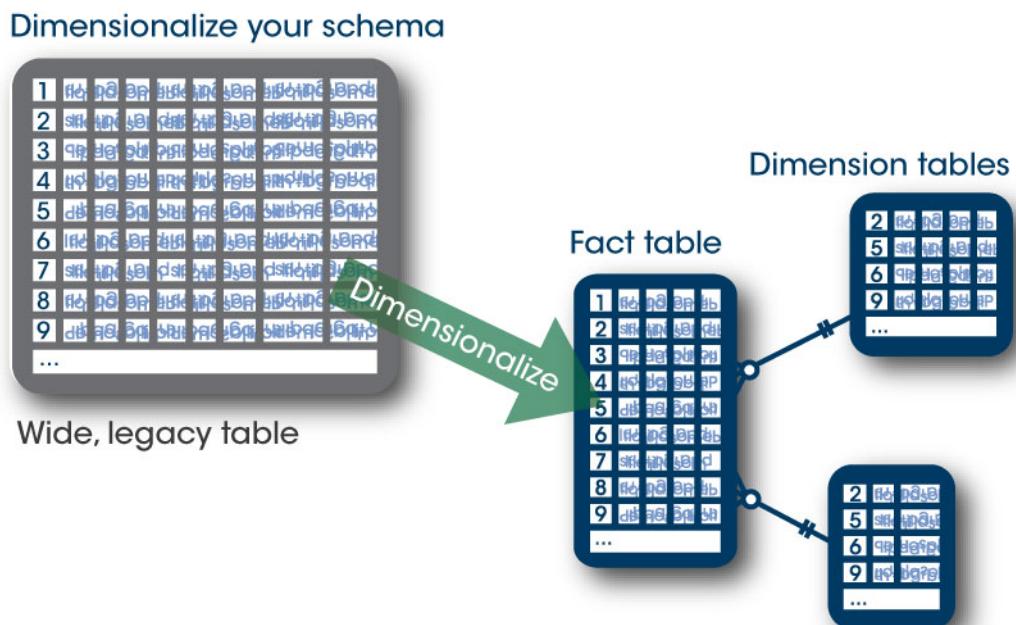
The sections that follow will discuss each rule in detail.

## Dimensionalize Your Schema

### Summary

Use the star schema model (also known as “dimensionalizing data”) to put your most frequently read columns into skinny fact tables, and to relegate the less frequently read columns to separate dimension tables. The goal is to make your fact tables skinny. This lets your queries run faster, because your queries don’t have to read the less relevant dimension information. Put more precisely, a query can scan more rows at a time, since each row is smaller, and this makes lookups faster.

Here’s a partial drawing showing dimensionalizing a schema:



### The star schema model

The star schema approach is commonly used in data warehousing for databases that contain large amounts of data. The star schema is efficient for large data sets because it relies on a single, narrow table (called the ‘fact table’) that avoids storing descriptive values and repeated values. Such columns are instead moved to helper tables called ‘dimension tables’. Time-

sensitive queries are run against the fact table only and can run very quickly because the narrowness of the table allows fast scanning. Queries that have to join against dimension tables take slightly longer to run (but note that Aster Database supports a number of techniques, outlined earlier in this document, for making these joins run fast).

## How to dimensionalize your data model

To dimensionalize your data model, identify your candidate fact table, which typically stores millions or hundreds of millions of rows that record the basic facts that database was built to track. Examine this table to find columns that might be better stored in a dimension table, so that scans of the fact table can avoid reading the contents of those columns.

The following types of columns are ones that you should consider putting in a dimension table:

- wide columns (for example varchar and bytea columns) that you don't read all the time.
- columns with low cardinality (ones that contain a relatively low number of unique values).

Where possible, move these lower-priority columns to one or more separate dimension tables. In the fact table, replace each removed column with a reference column that contains unique IDs (ints or bigints) that point to the corresponding rows in the dimension table.

In other words, you should be looking for columns with 'tag-along' values that map closely to values in other columns, and in particular you should be looking for columns where values are repetitive and large. Here, 'large' can be understood to mean any column that takes up a significant fraction of the total table width. Anything above 10% should stand out as a good candidate for removal. Of course, we assume that these columns are not frequently needed; if they're frequently queried, then you might need to retain them in the fact table.

### A note about satisfying both narrow and wide queries

If you have some queries that require the fact table to be narrow for fast performance, and other queries that need to include more columns in their results, then you may want to consider keeping the fact table wide enough to satisfy the wider queries but also creating materialized projections to satisfy the more narrowly focused, performance-intensive queries. This is explained in "[Verticalize Your Schema](#)" on page 197.

## The retail1 example schema

The *retail1* example database is modeled according to the star schema. The fact table, *sales\_fact*, stores only the most important details of a sales transaction, namely the date and time of the sale, the customer, store, basket (to group all the items purchased in a single sales transaction), product, quantity sold, and discount amount. Rather than being identified by (potentially large) varchar-typed names, most attributes are recorded in the fact table using only an int or bigint ID number: *customer\_id*, *store\_id*, *basket\_id*, and *product\_id*. The names and descriptions of the customer, store, and product are stored in separate dimension tables that need not be read by every query: the *customer\_dim*, *store\_dim*, and *product\_dim* tables.

Here's a partial drawing of the schema:

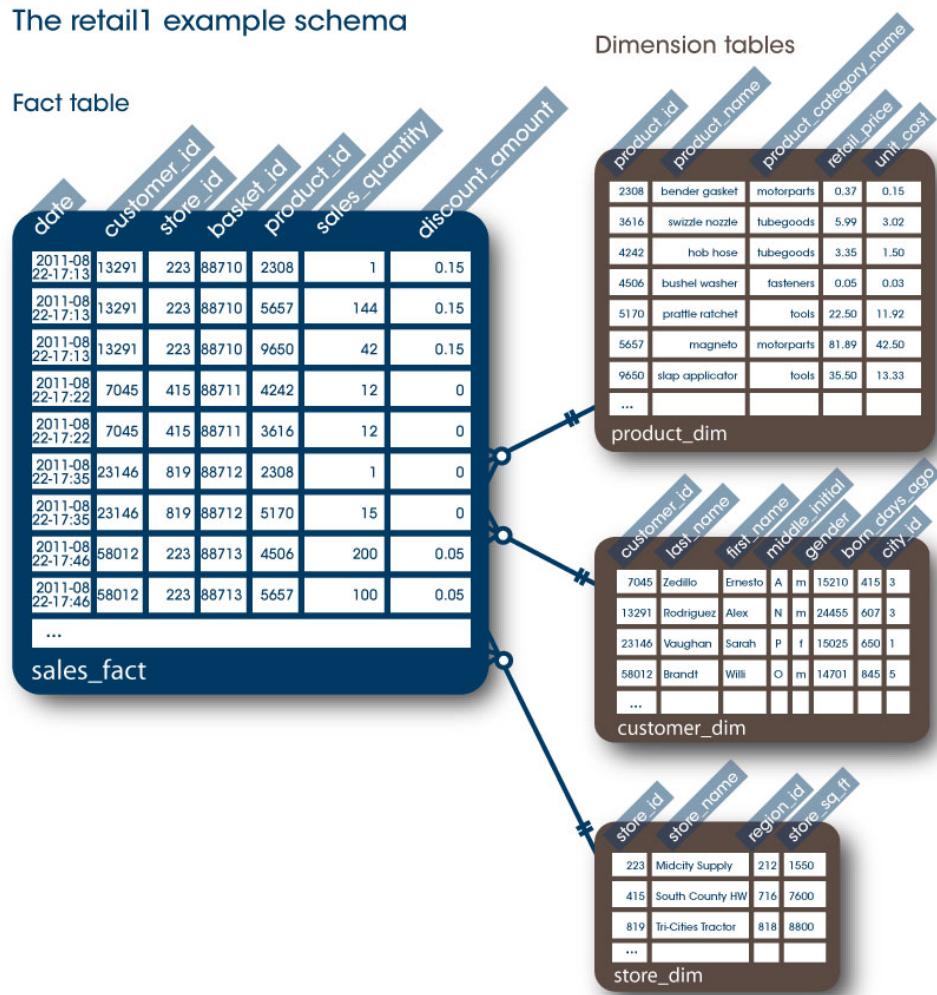


Table 2 - 2: A partial list of the tables in the retail1 schema

| Schema | Name         | Type  | Owner   |
|--------|--------------|-------|---------|
| public | customer_dim | table | beehive |
| public | product_dim  | table | beehive |
| public | sales_fact   | table | beehive |
| public | store_dim    | table | beehive |

Table 2 - 3: The sales\_fact table contains these columns

| Column | Type      | Modifiers |
|--------|-----------|-----------|
| date   | timestamp | not null  |

Table 2 - 3: The sales\_fact table contains these columns (continued)

| Column          | Type    | Modifiers |
|-----------------|---------|-----------|
| customer_id     | integer | not null  |
| store_id        | integer | not null  |
| basket_id       | bigint  | not null  |
| product_id      | integer | not null  |
| sales_quantity  | integer | not null  |
| discount_amount | real    | not null  |

Table 2 - 4: The product\_dim table contains these columns

| Column                | Type                  | Modifiers |
|-----------------------|-----------------------|-----------|
| product_id            | integer               | not null  |
| product_name          | character varying(20) | not null  |
| product_category_name | character varying(20) | not null  |
| retail_price          | double precision      | not null  |
| unit_cost             | double precision      | not null  |

Table 2 - 5: The customer\_dim table contains these columns

| Column               | Type                  | Modifiers |
|----------------------|-----------------------|-----------|
| customer_id          | integer               | not null  |
| last_name            | character varying(20) | not null  |
| first_name           | character varying(20) | not null  |
| middle_initial       | character(1)          | not null  |
| gender               | character(1)          | not null  |
| born_days_ago        | bigint                | not null  |
| city_id              | integer               | not null  |
| frequent_buyer_level | integer               | not null  |

Table 2 - 6: The store\_dim table contains these columns

| Column     | Type                  | Modifiers |
|------------|-----------------------|-----------|
| store_id   | integer               | not null  |
| store_name | character varying(20) | not null  |

Table 2 - 6: The store\_dim table contains these columns

| Column      | Type    | Modifiers |
|-------------|---------|-----------|
| region_id   | integer | not null  |
| store_sq_ft | integer | not null  |

## Benefits of using star schemas in Aster Database

You can apply dimensionalization with verticalization (see “[Verticalize Your Schema](#)” on [page 197](#)), to speed up query performance by further reducing the size of tables that must be scanned to find the desired rows. If you’re running SQL-MapReduce functions, a dimensionalized schema provides a much smaller memory footprint for SQL-MapReduce operations that do not need the dimension data, because in most cases these operations need only deal with an integer ID number, rather than the dimension data.

For more information and examples on how to dimensionalize data, refer to *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* by Ralph Kimball and Margy Ross.

# Use Columnar Tables When Appropriate

In Aster Database, for any table that you are willing to use as an append-only table, you have the option of using a columnar storage layout for the table. For many types of queries, the columnar layout allows faster query runtimes.

Use the following criteria to determine whether your table should use a column-oriented or row-oriented storage layout.

Use *row-oriented* storage if:

- You need to UPDATE, DELETE or MERGE rows in the table.  
(Columnar tables are append only.)
- Your cluster might in the future require *partition splitting* to grow the cluster, and you will want this table to survive the partition split.
- Your users have important queries that select *all* the columns of the table, or *more* than 60% of the table's width in bytes.

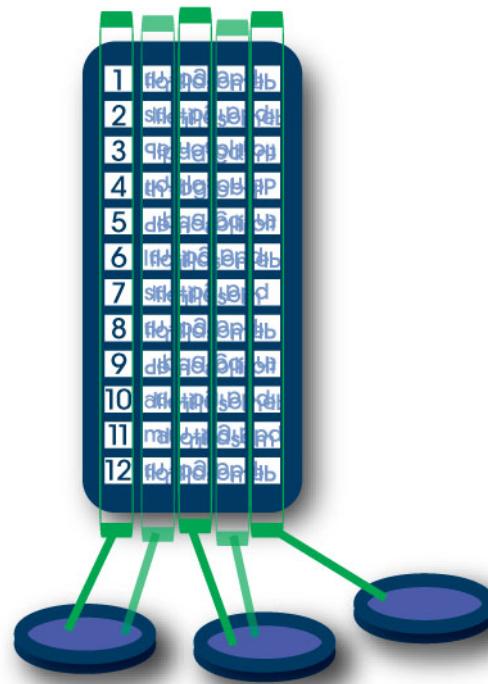
Use *column-oriented* storage if:

- You will never UPDATE, MERGE, or DELETE a row from the table.
- You can't dimensionalize because there is little redundancy in the data.
- Your important queries select less than 60% of the total width of the table. The width of the table is its width measured in bytes, not in number of columns. For example, if your table is ten columns wide, and the queries you care about select eight narrow INT columns and ignore the two remaining VARCHAR(255) columns, then using a columnar table is likely to provide performance benefits, since the queries select less than 60% of the table's width in bytes.



**Tip!** If many views are defined on your table, it can be hard to determine whether queries are selecting less than 60% of the table's width. Contact Teradata Global Technical Support (GTS) for help in assessing whether a table that underlies views should be converted to columnar storage.

Column-oriented storage makes most queries run faster



If your table meets the criteria for column storage, then turn to “[Columnar Tables](#)” on page 54 for instructions on using columnar storage.

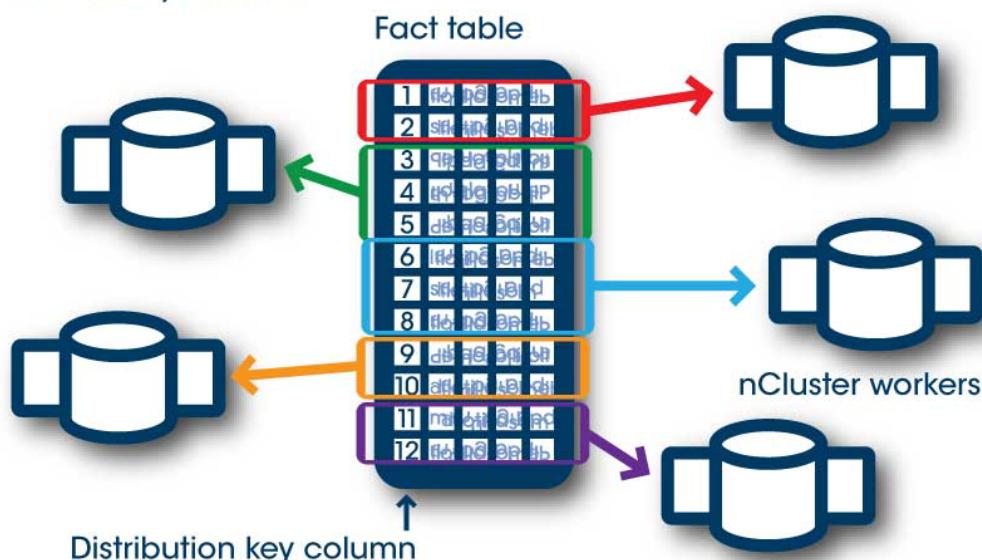
# Distribute Your Data with Joins in Mind

## Summary

Aster Database always distributes the contents of fact tables throughout the cluster (large dimension tables can be distributed, too). This allows each worker to manage its own slice of the data. Your job is to choose a good distribution key column to specify how the data will be sliced for distribution. A good distribution key is one that enables a large portion of your most common, costly operations (joins, DISTINCTs, and GROUP BYs) to be performed locally, within individual workers, before the operation is handed off to the Aster Database queen to assemble the results.

When choosing the distribution key, you should also, as a secondary concern, bear in mind the evenness of your data distribution. That is, make sure the distribution key column contains enough unique values to allow Aster Database to distribute data so that no single worker holds significantly more data than any other. When a worker holds too much or too little data, relative to other workers, we say there is *data skew*. Avoiding data skew is important, but it is really a secondary concern because *matching your schema to your joins* is more important.

### Distribute your data



*Tip:* For step-by-step instructions that show how to set up a distributed table, see “Table Distribution” on page 31.

## Choosing a Distribution Key

When you create a fact table that will contain a large amount of data, you must physically distribute it across the worker nodes in the cluster. To physically distribute a table in Aster Database, you specify a single column as the distribution key when you create the table. For a

given row, the value in the distribution key column determines where in the cluster (on which worker) that row is stored. By designating the distribution key, you indicate to Aster Database which records you want grouped together on each worker.

Follow these guidelines to pick the distribution key column for a table:

- 1 Consider your joins. Choose the column that will be used in your most performance-critical joins. When picking your distribution key, choose the column that is most frequently used in joins or aggregations (GROUP BY or DISTINCT), in that order. Since Aster Database is optimized for joins, it is cost effective to design table schemas so that as many joins happen on the distribution key as possible. Aster Database is also optimized for aggregation on the column specified as the distribution key. Therefore, when there are no joins but only aggregations (via GROUP BY or DISTINCT), then using the column most frequently involved in the aggregation as the distribution key provides better performance.
- 2 Data skew is secondary. Consider using a distribution key that avoids data skew, but remember that when you're optimizing for performance, it's more important to pick a distribution key that matches your joins than to pick one that avoids data skew. Data skew occurs when the distribution key causes a disproportionately large number of rows to be routed to a single worker in the cluster. As a result, one worker has a very large amount of data, and all other workers have correspondingly smaller amounts of data. The one worker with the majority of records performs slowly, as expected, and this can slow down queries that need to access the skewed data.
- 3 What if there's no good candidate? If no appropriate distribution key column exists, you may need to create a surrogate distribution key during loading. To do this, look for a column or columns whose values might be transformed or combined to create more useful distribution key values. As discussed in point 1 above, let your users' actual join predicates guide you to find useful values to distribute on. You can use SQL-MapReduce functions to perform the needed transformations during loading. If you have no existing join predicates to guide you, then you can create distribution key values that just minimize data skew. For example, you might define a `an "id" column of type UUID`, and then, in your data loading code, include an SQL-MapReduce function that uses a utility like `java.util.UUID` to create a fairly universal identifier for every row. The broad distribution of these values ensures good data distribution in Aster Database.

## Examples of Physically Distributed Schemas

Let's assume you've designed a database to hold customer product orders, and you've followed the star schema approach in your design. A core *fact table* holds the customer order transactions, a *dimension table* holds customer account details (name, address, etc.), another *dimension table* holds product details (product id, name, price, etc), and so on. The fact table is very large, holding on the order of a billion rows. Obviously, you want to physically distribute the fact table so that each Aster Database worker holds a reasonable portion of its data.

To choose the distribution key, you need to review the typical queries you will run on this system. Let's assume your analytical queries include ones that ask:

- Which products were frequently purchased together?

- What products were purchased by customers who have shopped here before?

These queries typically join on `order_id`. In addition, `order_id` will give an even distribution across workers, because there is a new `order_id` number for each customer purchase. This means you won't have any skew-causing spikes of rows with the same `order_id`. Thus, a good choice of distribution key in this example is the `order_id`.

Note that you might also consider using `customer_id` as a distribution key. This is a reasonable choice since some of the queries do require a join on `customer_id`. `Customer_id` could, however, introduce skew, especially if most of your orders come from a small percentage of your customers. Choosing the right distribution key takes some thought, and it often requires some testing and performance benchmarking to find the optimal key for your data.

Web analytics schemas offer another good example. They typically contain a fact table called `page_views` that stores (you guessed it) a new row every time a user loads a page from the website. In such tables you should always distribute the `page_views` fact table. Good choices for the distribution key include columns like `user_id` and `cookie_id` for the following reasons:

- The `user_id` column is typically used to join the `page_views` fact table with a dimension table that stores the users (typically, the table has a name like "users").
- It is common to query for the count of distinct values of `user_id` or `cookie_id`.

For an example of a table distributed using a distribution key, see "[Another Example: The orders Fact Table](#)" on page 23.

## Single-Column Distribution Key

In Aster Database, the distribution key is always a single column. Multi-column distribution keys are not supported. Single-column distribution keys are efficient for GROUP BY, DISTINCT, COUNT (DISTINCT), and joins on the distribution-key column, and they are also efficient for GROUP BY, DISTINCT, COUNT (DISTINCT), and joins on the distribution key column plus any other column in the table. For example, let's refer back to our earlier example with the 'orders' fact table. Our distribution key is `order_id`. Since each `order_id` group is local on each individual worker, each `<order_id, customer_id>` group is also local.

If your use case matches both of the following criteria, you may need to *simulate* the effect of having a multi-column distribution key. Assume a distribution key column that consists of both `col1` and `col2` for this example. You should check your database for data skew if:

- 1 your data distribution is skewed on `col1`, it's skewed on `col2`, but it's NOT skewed on the combination of `<col1, col2>`; and if
- 2 your queries on `<col1, col2>` are more important than your queries on just `col1` or just `col2`.

If your database suffers from skew, then you should consider using one of the following workarounds:

- Choose a different distribution key column.
- Perform an ELT (inside db) or an ETL (outside db) transformation to redistribute the skewed value(s) to a different value range. For example, one customer stores users distributed by `user_id`, but guest user records all have a `user_id` of "0", presenting a skew

problem. To avoid the problem, the loading routine performs a transformation to remap guest user ids to a range of 1,000,000,000 - 1,000,001,000.

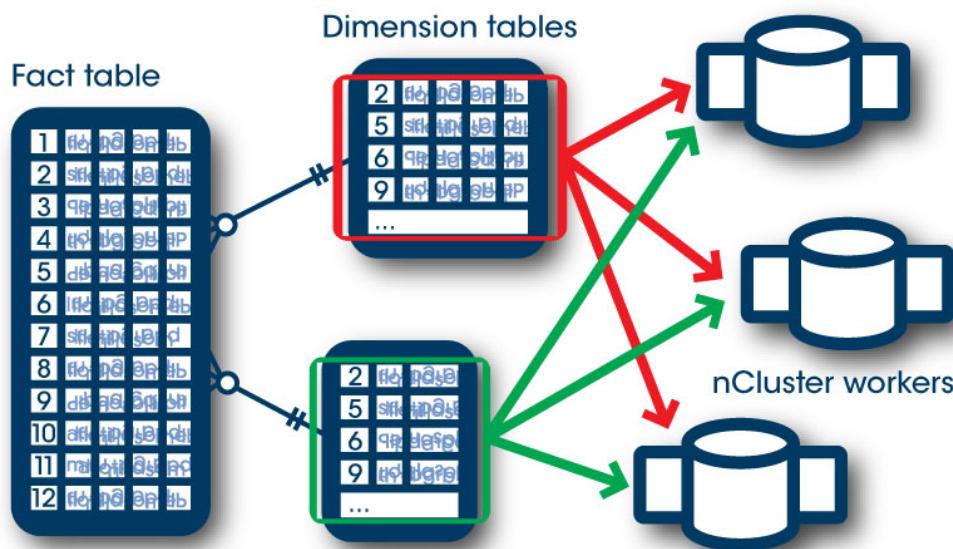
- Perform an ETL/ELT transformation to generate a new, synthetic column that will serve as a good distribution key.

## Replicate Common, Frequently Joined Data

### Summary

Use replicated dimension tables in Aster Database to ensure that a copy of each frequently joined dimension table is always present on the local machine for joins. Replicated dimension tables are copied to every node and therefore always available locally on every worker in the cluster. As we mentioned earlier, the rows of a fact table are distributed throughout the cluster for efficient storage and processing. To perform fast joins against these physically distributed fact table rows, it helps to have a local copy of the dimension table you are joining with.

#### Replicate common, frequently joined data



**Tip!** If you have low-cardinality columns in your fact table, those are good candidates to be removed from the fact table and put in a replicated dimension table. We call this "dimensionalization." See "Dimensionalize Your Schema" on page 181.

### Replicated Dimension Tables in Aster Database

A replicated dimension table is a dimension table whose entire contents are copied to all vworkers for faster lookup. This is the default behavior of a dimension table in Aster Database. As long as you don't include a distribution key in your CREATE DIMENSION TABLE

statement, you are creating a replicated dimension table. Replicated dimension tables are especially good for small data sets, such as a table that translates zip codes to states, because such tables are likely to stay in cache in memory and thus can be accessed extremely quickly even when a join is involved.

For example, in a web analytics schema, dimension tables such as ads, pages, and domains should be fully replicated. If the users dimension table has thousands of rows, then it should be replicated. If the users dimension table has millions or hundreds of millions of rows, then it should be distributed on the column, user\_id.

Don't confuse replicated dimension tables with Aster Database replication! They are not closely related. What's being replicated *in Aster Database replication* are workers (to be very precise about it, we're talking specifically about "vworkers" here, which are also sometimes called "partitions" by Aster Database oldtimers) whereas what's being replicated *in a replicated dimension table* are the whole contents of the table.

## Very Large Dimension Tables Should Be Distributed, Not Replicated

A replicated dimension table should be small enough to fit in the memory available to each vworker. To determine the maximum recommended size of a replicated dimension table in your environment, divide the amount of memory in each worker node (machine) by the number of vworkers on each node. To find out the number of vworkers on each node, open the Aster Database AMC management console and click the **Nodes > Partition Map** tab. This tab shows the nodes of the cluster. Inside each node icon, you'll see a number of small squares representing the vworkers on that node.

If you think a dimension table will exceed the maximum recommended size, do not create it as a replicated dimension table. Because Aster Database maintains a full copy of each replicated table per vworker (there's no sharing, even among multiple vworkers on the same worker node) the large size of the table can slow your queries.

Instead, use one of these solutions:

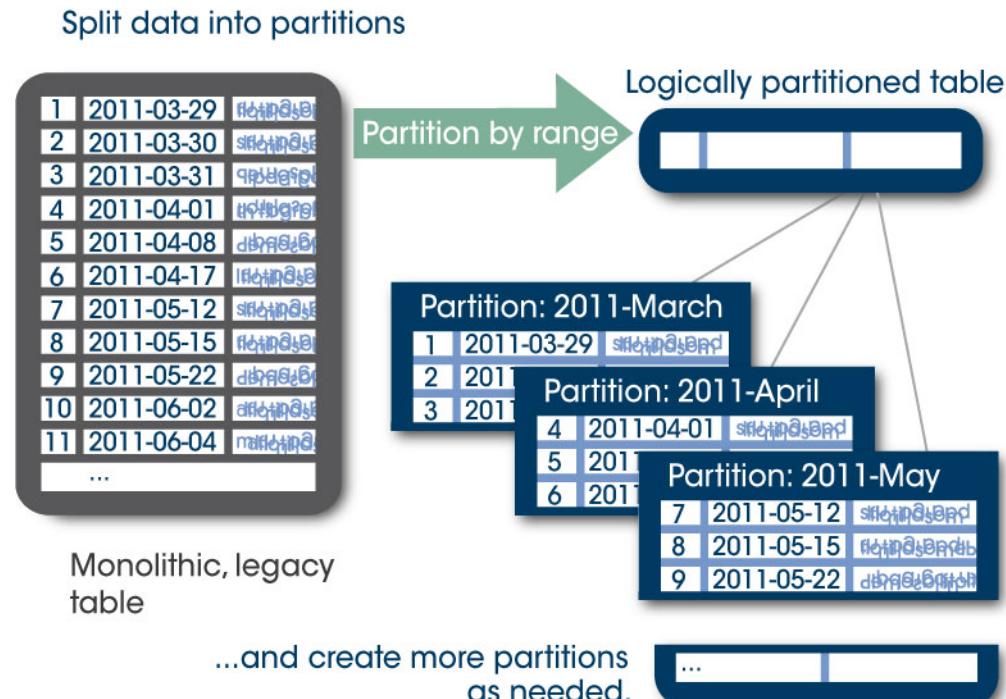
- Create the table as a distributed dimension table (a dimension table with a distribution key) instead of a replicated dimension table (but be aware that some joins may now have to redistribute the fact table, resulting in slower query performance); or
- Denormalize your schema to remove the dimension table entirely, moving the needed rows to the main fact table. If you do this, you should look at your large dimension table to see if it contains some low-cardinality columns that would be good candidates for a smaller dimension table that could be replicated.

## Split Data Into Child Partitions

### Summary

Use logical partitioning to give the effect of smaller fact tables, which shortens query runtimes and simplifies management of the data over the course of its lifecycle. When a large table has

been split into child partitions, most queries on that table can avoid time-consuming sequential scans of the large table by instead scanning only the applicable child partitions.



Logical partitioning creates a hierarchy of child partitions that act as a single table but give users the option of dealing with only one or a few subsets of rows (each stored in its own child partition) at a time. Partition constraints (for example, defined on a date, product\_id, customer\_id, or similar column) ensure that each child partition contains only the desired slice of data. Users run their SELECT queries on the top level partitioned table. Each query should include in its WHERE clause a predicate that falls within the partition definition of one or more child partitions. Provided that the child partition structure matches the user's predicate, Aster Database reads only the relevant child partitions, resulting in fast query runtimes.

## When to Use Logical Partitioning

When should you use logical partitioning? The short answer is, “most of the time.” In most schemas, the fact tables are logically partitioned. It makes sense to logically partition most fact tables because they typically contain data that you’ll want to analyze in a bucketed manner (e.g. by specific time ranges) and also because logical partitioning makes it easier to expire old data.

Specifically, you should use logical partitioning in the following use cases:

- When your queries typically have filter predicate that looks at predictable subsets of data (for example, monthly or daily sets of rows).
- When you need to perform data expiration according to a regular schedule. Dropping child partitions is faster and simpler than deleting rows from a large table. Detaching and

deleting a child partition is a relatively fast operation and quickly frees up space in the cluster for new data. In contrast, performing an SQL DELETE of rows in a large table will scan 100% of the table's data if the table is not indexed (potentially taking a long time). This operation also creates dead space in the cluster that you must free in a second operation (i.e. VACUUM).

## When should I not use logical partitioning?

In practice, it's rare to find fact tables that don't benefit by being split into child partitions. However, if your data and queries meet the following conditions, then you might be better served by a single, monolithic fact table:

- the data in your fact table never expires, and
- a significant number of your queries need to process the entire dataset.

In other words, if your queries don't have common WHERE clause predicates that can guide you to create useful child partitions, and if you plan to continue analyzing older data for a long time to come, then you may want to consider leaving your fact table as a single, nonpartitioned table.

## Benefits of Logical Partitioning

Logical partitioning and indexes can be used together or separately. If the largest partition is not very large, you might decide not to use indexing. Compared with indexes, child partition hierarchies offer sequential I/O rather than index access doing random I/O, and they offer lower storage/maintenance costs because:

- unneeded child partitions can be quickly detached and dropped to recover space;
- there is no space overhead required to maintain a child partition hierarchy, in contrast with a physical index, which takes up space;
- there is no maintenance overhead for row insert/update/delete operations, in contrast with indexed tables, which incur a performance cost due to the overhead of inserting/updating/deleting to or from the index.

You can use logical partitioning and indexes together. If the child partitions are indexed, update and delete performance is typically much faster than it would be with a single, indexed table, since each piece of the table has a much smaller index than an index on the entire data set would be.

## How Logical Partitioning Works

Logical partitioning refers to splitting what is logically one large table into smaller child partitions for faster performance and easier management. A partitioned table may contain many levels of parent, child, grandchild, and so on. Data is typically stored only at the lowest level partitions in the hierarchy.

The partitioned tables occurring above the lowest level in the hierarchy are normally empty of data. These tables are there to represent the structure of the hierarchy and to act as the target for SELECT, UPDATE, and DELETE queries on the logically partitioned table. Some logical partitioning schemas contain multiple generations of parents. For example, you might have a

schema in which the table sales\_2008 has child partitions sales\_2008\_01 through sales\_2008\_12, and each child partition, such as sales\_2008\_01, in turn has a set of daily child partitions (for example, sales\_2008\_01\_01 through sales\_2008\_01\_31) that contain data rows.

You create the hierarchy by creating a partitioned table with one or more levels of partitions by range or by list of values.

## Tips for Creating Logically Partitioned Tables

### Child partitions: How many is too many?

Don't create too many child partitions. For good performance, make sure that each partitioned table contains no more than 200 child partitions at any one time. Having more than 200 partitions in a hierarchy requires significant query planning time from the queen. If you find your set of child partitions growing too large, you should consider implementing a policy to drop older partitions at a regular interval.

To choose your level of granularity for child partitions, take a look at your reporting requirements. The granularity should match the lowest granularity at which you typically prepare reports. For example, if you are doing time-based analysis and you frequently prepare hourly reports, then you should create hourly child partitions. Or, if you prepare daily reports but nothing more granular than that, then you will create daily child partitions.

### Constraints and predicate must match

You only get the performance benefits of child partitions if the user's WHERE clause matches the definitions of the child partitions. This means that, if your goal is to improve performance, then you should only logically partition your tables if the queries you care about use consistent WHERE clause conditions that you can match in the list or range of values used to define the child partitions.

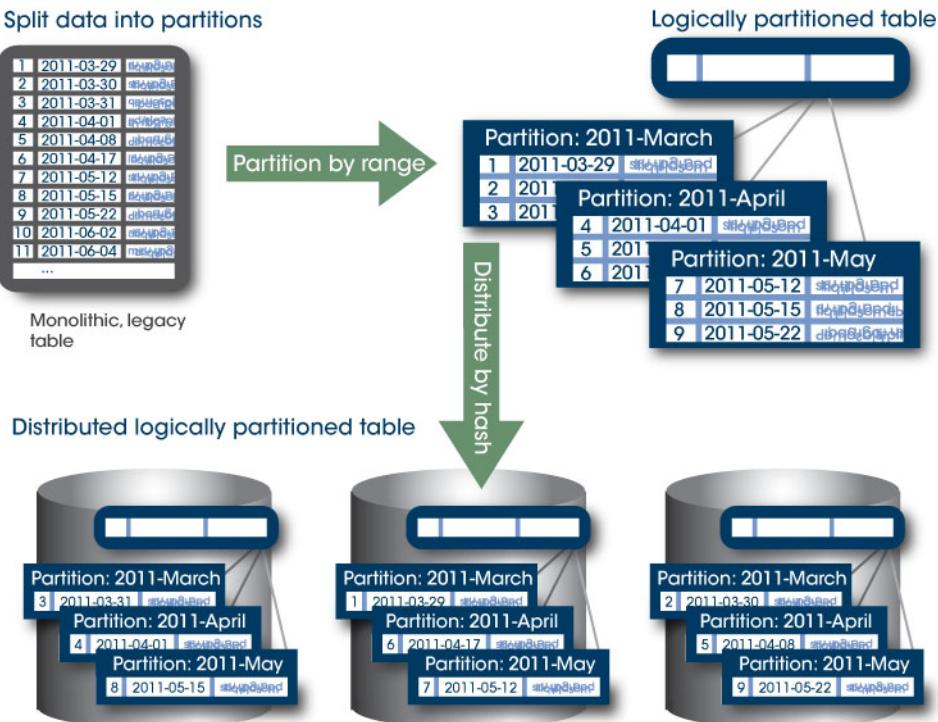
### More tips

The section "[Automatic Logical Partitioning](#)" on page 35 contains more tips for working with logically partitioned tables.

## Interaction of Logically Partitioned Tables and Aster Database Distribution Keys

Since nearly all logically partitioned tables are also fact tables in Aster Database, their data is not only split into partitions but also distributed throughout the cluster using a distribution key. These two methods of spreading out your data are mostly unrelated, but you should take care to choose a distribution key that does not work at cross-purposes with your logical partitioning scheme.

When choosing a distribution key that will distribute the rows of a logically partitioned table (that is, a table that will be both distributed among workers and logically partitioned), you will typically use the same distribution key you would have used, had you not also logically partitioned the table. However, please note that it rarely makes sense to use as a distribution key *the same column whose value determines the partitioning of the table*. In other words, most of the time you'll DISTRIBUTE BY one column and PARTITION BY another.



## Using a date/time column as both the distribution key and logical partitioning constraint

There are times when it's OK to use values from a single date/time column as both your distribution key and as your logical partitioning constraint. To do this, you should use a granular timestamp (down to the second) as the distribution key, and use the less granular date as the logical partitioning constraint. Such schemas are typical for transaction databases, which typically have precisely timestamped data but do their reporting at a more coarse-grained level (per day, per quarter, or similar).

When you choose a common column as the distribution key and partitioning constraint, the high granularity of the timestamp value provides a distribution key with good uniqueness, ensuring even distribution of the rows in the cluster, and the less granular date value provides a useful, logical bucketing of the data into child partitions that allow faster query response and easier management of the data over its lifecycle.

## Date dimension precludes use of child partitions

### The problem with date dimensions

Many star schema-style data models store dates in a date dimension table rather than in the fact table. This means that the fact table contains a date\_id column rather than a date column. The date\_id points to a date entry in the date dimension table. This type of schema presents a problem for logical partitioning: If you use a date dimension and retain only a date\_id in your fact table, then you cannot bucket the fact table data into partitions, because there is no date column in the fact table, so you cannot use the date column for partitioning. This problem

arises because the planner check the partition definition before it performs the join to the dimension table, so it fails to see the date and cannot make use of it to optimize the query.

Background: Why do some use cases require a date dimension? Typically it's because the analysts require various bucketing granularities that they record in the dimension table. So if some people produce monthly reports, others produce calendar-year quarterlies, and others produce fiscal-year quarterlies, each of these designations can be recorded in a column of the date dimension table. With this in place, the analyst can write his report queries without needing to add logic that resolves dates into buckets.

### **Solution for date dimensions: Use indexes, instead**

If your schema requires the use of a date dimension, then you can use indexes to get some of the benefits that you would have had from using logical partitioning. To do this, define an index on the date\_id column in the fact table, and set up Aster Database to encourage the planner to choose a nested loop join when joining the fact and the date dimension. (See the section “[Index Your Tables](#)” on page 198 for further instructions.)

For example, imagine a query that searched for dates in “Q1 2011”. Based on the join, the “Q1 2011” date resolves to, say, 20 different date\_ids. Using the index on the date\_id and performing a nested loop join, Aster Database can quickly retrieve the desired records from the fact table because each loop in the nested loop can benefit from the date\_id index in the fact table.

Beware wide date ranges! Applying indexes doesn't work as well if a typical query in your environment searches for a wide range of dates (like, for example, searching for a two-month range in a database that stores many rows per hour). In this case, more random access happens, resulting in slower query performance. If this applies to you, consider the next solution, explained below.

### **Another solution: Denormalize, adding a date column to your fact table**

Denormalize your fact table, placing copies of the *date* column and any required date-bucketing columns (for example, quarter\_id) in your fact table. This allows you to logically partition the fact table. If your fact table has both a date column and one or more date-bucketing columns, then you can use multiple levels of partitioning in your hierarchy. A partitioning hierarchy with multiple levels provides fast query performance because non-matching partitions are eliminated from the search.

## **Simulating the effect of logical partitioning with clustering**

If you want the benefits of logical partitioning but cannot use logically partitioned tables (for example, if no suitable column exists that you can evaluate in to define partitions) then you can get many of the same benefits by clustering your data on an index or indexes. This practice also goes by the name of “clustered indices”.

When you do this (with the CLUSTER command) the table's data is laid out in a sorted fashion along the index of choice. The problems with this approach are that sorting is an expensive operation, and the act of clustering data is a one-time event and the on-disk sort

order is not maintained when new data is inserted or old data is updated. If you substantially change the data in the table, you must re-CLUSTER it.

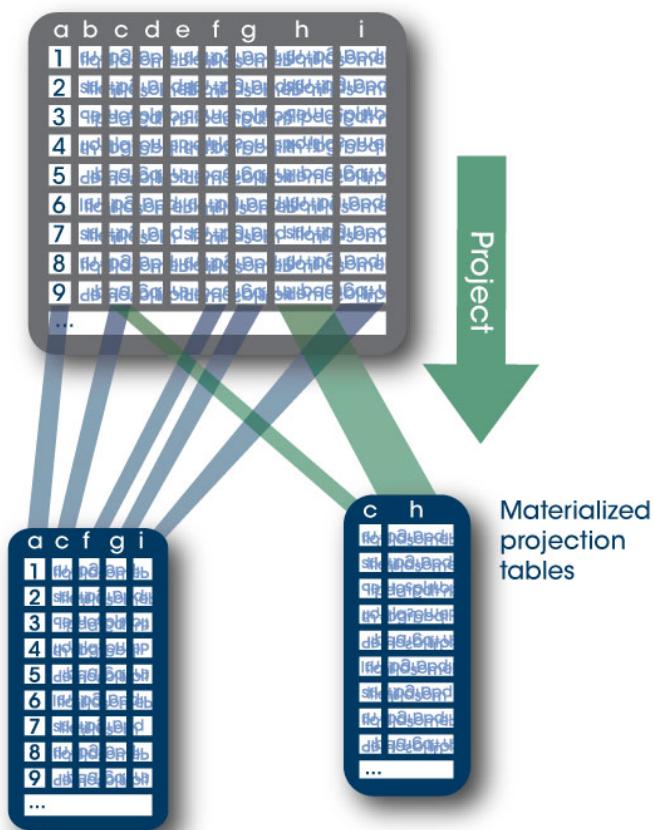
## Verticalize Your Schema

### Summary

Make your fact tables narrow by creating relevant materialized projections. Run your queries against the projections for fast execution. This can make queries faster when columnar storage is not an option.

#### Verticalize your schema

Wide, legacy table



### When is verticalization useful?

Columnar storage provides efficient access to tables when only a few columns are being read. However, columnar tables are not always the best option (see [“Use Columnar Tables When Appropriate” on page 186](#)).

Verticalization can be used for row-formatted tables to achieve some of the similar performance benefits of columnar storage. The only caveat is that a verticalized table requires more manual effort to maintain as the data changes.

If your schema contains a wide fact table that must remain so to satisfy some users, but you wish to narrow the table to allow other queries to run more quickly, then you can improve performance by creating materialized projection tables that include only those columns needed by your high-performance queries. We call this “verticalization” because the projections are more vertical in nature than the wide fact tables they represent. Verticalization is useful, for example, if you have wide tables on which you run daily reporting queries that select only a few of the columns and must run quickly.

Tip: In most cases, 60% is the magic number: If most of your queries hit less than 60% of the fact table’s columns, then you should consider creating a verticalized projection of the fact table.

## Costs

When you weigh the usefulness of verticalization, bear in mind the costs of implementing it:

- You must build a two-step loading process to create both your fact table and its materialized projection(s). This typically means a standard load to the main fact table and a `CREATE TABLE AS SELECT` to create the projection.
- You must rewrite some of your existing queries to run against the materialized projection, rather than the main fact table.

## How to verticalize your schema

To verticalize your schema, you create new tables that contain copies of only those columns that are frequently queried together. We refer to such copy-tables as materialized projections. Maintaining materialized projections requires that you periodically update or recreate the materialized projection with data from the source fact table.

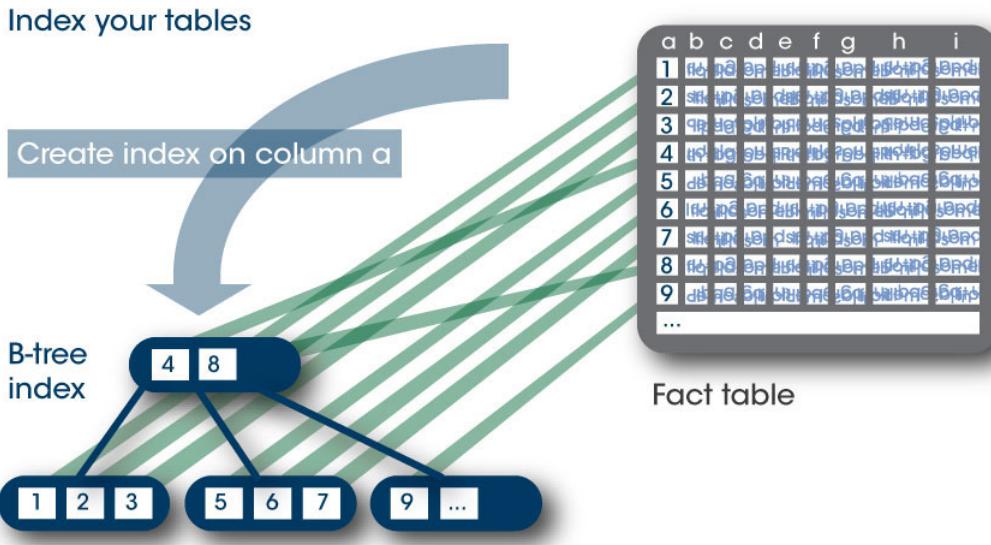
Your queries that use only the projected columns will now run faster, while other queries that need data from the columns not in the projection can continue to use the original, wider table.

For example, in a clickstream tracking database that logs users’ views of website pages, you might precompute `page_view` summary statistics for every combination of user, `page_view`, and domain.

# Index Your Tables

## Summary

Indexing tables makes retrieving selective information fast. Targeted lookups prevent irrelevant data from being read and processed.



## When to use indexes in Aster Database

Generally, it's a good idea to define indexes on a table if that table is frequently the target of point lookup queries (queries looking for a single row) or queries that use highly selective filters. In other words, if you have a large table from which the typical query selects only a few rows, then you should define an index on the column(s) that most queries use to filter their results.

Indexes can also be very useful for speeding up group-bys and joins. In particular, if your queries result in nested loop joins or merge joins, you may see faster query runtimes if you add indexes.

## How indexes improve performance

Indexes allow you to avoid sequential scans (that is, full table scans) which may be relatively time-consuming. If, when running a query, Aster Database finds that the table has an index or indexes that are likely to be useful in finding that query's desired results, then Aster Database will choose an alternative, potentially faster scan method. These faster methods are the *index scan* and the *bitmap index scan*.

Index scans can be very fast. If the query has the indexed column in its WHERE clause and Aster Database estimates that the predicate will select less than 20% of the rows in the table (based on Aster Database's table statistics) then Aster Database is likely to choose an index scan rather than a sequential scan.

## Indexes can speed up nested loop and merge joins

Having indexes on join columns can substantially speed up the performance of nested loop joins and merge joins in Aster Database. (To find out what type of join Aster Database is choosing for your query and to find out how to tune Aster Database if it's choosing a sub-optimal type of join, follow the instructions in “[Query Tuning Best Practices](#)” on page 208.)

Below, we explain where you can add indexes to speed up nested loop and merge joins.

### A note about hash joins

In typical data warehouse join scenarios where the set of distinct values being joined on or grouped by is small enough to fit in memory, hash joins are likely to run more quickly than nested loop and merge joins. Hash joins work well when neither table is small but the smaller table's joining column can be used to construct a hash table that the planner estimates will fit in memory. If your queries typically involve tables small enough to meet this criterion, but not small enough for nested loop joins to apply, then hash joins will be more common than nested loops and merge joins in your environment. Indexes do not help to speed up hash joins, but they can help the planner substitute a nested loop or merge join, if appropriate.

### Indexes for nested loop joins

In a nested loop join, Aster Database reads a row in the left relation, then reads every row in the right relation looking for matches, then proceeds to the next row in the left relation, and so on. By defining an index on the join column in the right relation (this is typically the fact table, or the larger of the two tables being joined), you can speed up the nested loop join since Aster Database can perform a fast index scan of the right relation, rather than a relatively slow sequential scan (full table scan).

### Indexes for merge joins

Merge joins are relatively rare in the workloads typically run on Aster Database, because, for the planner to choose a merge join, the distinct values of the columns being joined must be small enough fit in the memory of each vworker. If your tables do meet that criterion, then read on below for details.

In a merge join, each relation is sorted on the join attributes before the join starts. Then the two relations are scanned in parallel, and matching rows are combined to form join rows. By defining indexes on the join column of both tables being joined, you can speed up the merge join since Aster Database can sort both relations using the index. If the joined-on columns of both tables are indexed, this type of join can be significantly faster than a hash join because the sorted structure of the index is already in place before the query runs.

## Conditions that must be met in order to use an index scan

Aster Database performs an index scan only if all of the following criteria are met:

- indexes are available; and
- the query filters on a column that has an index; and
- the optimizer thinks that the query's filtering on the indexed column will remove at least 80-90% of the rows from the result. Please note that what matters is what the optimizer thinks, so always make sure you run ANALYZE on the table after any significant change to the table's contents.

## Notice! Enabling index scans can hurt performance

There are cases in which index scans might seem to be useful but shouldn't be turned on due to the potential for slow performance. For many slightly complicated queries, the planner doesn't estimate the cost of an operation exactly right. In such scenarios, the planner might mistakenly use an index scan because it assumed high selectivity, when in actuality the selectivity is not high enough. If this happens, the query's disk access pattern will be more like random I/O, resulting in slower query performance.

## Logical partitioning as an alternative to indexes

When your table contains a large number of rows but your typical queries only select a small portion at any point, indices are usually helpful because they can reduce the amount of scanning (reading all rows in the table) needed to find a query's results, but indexes are not the only way. In these situations, using logical partitioning may also reduce scanning and should be considered as an alternative to indexing.

To decide whether indexes or logical partitioning is a better approach, you should consider two main criteria. First, consider the selectivity of your queries: Highly selective queries tend to benefit more from indexes, while less selective queries tend to benefit more from logical partitioning. Second, consider the work needed to create and maintain the indexes or partitions. Both require effort and impose certain limitations or penalties when loading data. In the sections that follow, we'll explain these two criteria.

Often, you may not know in advance what your data distribution or query patterns will look like. If this is the case, then the relatively lighter-weight approach of adding indexes may work best.

### Query selectivity: Indexes vs. logical partitioning

In deciding whether to apply indexes or child tables to reduce table scan times, you should consider how selective your typical query predicates are. If your queries typically perform highly selective filtering based on a column that is useful for that, such as a user\_id or transaction\_id, then it's probably a good idea to define an index on that column. By contrast, if your queries are typically less selective and filter on a column with fewer unique values (lower cardinality), such as a gender column or a month column, then you're likely to get better results by logically partitioning on the column, rather than creating an index on it.

A good candidate for logical partitioning (instead of indexing) is any table on which the typical query filters out 10%-20% of the data. For example, if the column on which you filter contains null values in 15% of the rows, but your queries are looking for specific values in that column, then having a child partition that collects the null-in-that-column rows will speed up your queries substantially.

### Maintenance costs: Indexes vs. logical partitioning

Neither indexes nor logical partitioning provide their benefits for free; both need to be created and maintained, and both impose costs at data loading time. For indexes, any significant insertion, update, or deletion of data requires re-creating the indexes. See "Operational considerations when you use indexes" for details. For logical partitioning, loading can be somewhat slower since rows must be directed to the right child partitions. See "[Automatic](#)

[Logical Partitioning](#)” on page 35 for details.

### Nulls: Avoiding scanning of rows with/without nulls in a column

When a user submits a query containing an IS NULL or IS NOT NULL clause, it's useful for the planner to have fast way to avoid scanning rows that it doesn't care about (that is, to avoid scanning any row in which the filtered-on value is NULL or NOT NULL, as the case may be).

The planner does not, by default, use an index to help speed up such IS NULL / IS NOT NULL searches. If you like, you can encourage the planner to use indexes in such cases. To do this, you must create a partial index whose definition includes an IS NULL or IS NOT NULL predicate.

However, it's usually more efficient to skip indexes entirely for these types of searches, and instead use logical partitioning. This solution is better since it groups together related rows and therefore avoids the random pattern an index creates.

To do this, create a logically partitioned table hierarchy in which the partition definitions ensure that all rows with a null in the relevant column are saved in a partition that you've set aside for that. Queries whose predicate requires a specific value or a NOT NULL in the relevant column will not scan the null-valued rows.

## Multicolumn indexes

Multicolumn indexes (also called composite indexes) are useful when a set of columns are frequently filtered on together. They become extremely beneficial when a particular combination of buckets often occur together (e.g. user\_id between 10 and 100 and session\_id between 1 and 20 occurring together in many scenarios).

To ensure that Aster Database takes advantage of multicolumn indexes, define an index in the form (assuming a table t with columns *cola* and *colb*):

```
CREATE INDEX t_idx ON t (cola, colb);
```

and make sure users include the indexed columns in their predicates. The handling of multicolumn indexes in Aster Database matches that of PostgreSQL. As stated in the PostgreSQL documentation, a multicolumn index can be used with query conditions that involve any subset of the indexed columns, but the index is most efficient when there are constraints on the leading (left most) columns.

## Bitmap index scans (using multiple, single-column indexes)

### Background

If your workload includes ad-hoc queries that often filter on multiple columns, you may be able to speed up these queries by defining multiple indexes on the table (each index on a single column). This enables Aster Database to perform bitmap index scans when trying to match on those columns.

It's important to emphasize that in this context we're talking about *ad hoc* queries whose predicates are fairly unpredictable. If your workload's predicates don't vary much (that is, they use a fairly constant combination of columns in their predicates), then you can define a

multicolumn index (as discussed earlier) that speeds up scans of the table for those queries. Unfortunately, many data analysis environments are not so predictable, meaning that you don't know what combinations of columns your analysts will be filtering on. If this case applies to you, you may wish to consider enabling bitmap index scans in Aster Database.

As explained in the PostgreSQL documentation, in bitmap index scanning, if a query has a predicate like, for example, "WHERE tab.col1 = 4 AND tab.col2 = 9", and there is no multicolumn index on col1 and col2, but there is an index on col1 and another on col2, it is possible to search both indexes and combine the results in memory, then do heap fetches for only the rows matching both the col1 and col2 restrictions. This is very useful in environments that have a lot of unstructured queries where it is impossible to create indexes that match all possible access scenarios. Bitmap scans are useful even with a single index, as they reduce the amount of random access needed; a bitmap index scan is efficient for retrieving fairly large fractions of the complete table, whereas plain index scans are not.

### **When to use bitmap index scans**

Bitmap index scans can be fast when the query filters on multiple columns and the table has indexes on these columns. Stated more precisely,

- if the table has multiple indices AND
- if the query has a predicate that filters on the columns on which these indices have been constructed AND
- if the Aster Database optimizer thinks that the filtering on these columns will retrieve no more than 10-20% of the table's contents,

then Aster Database will use a bitmap index scan. To perform a bitmap index scan, the query optimizer creates an in-memory bitmap that indicates which rows match the filtering criteria in all filtered columns, enabling Aster Database to select the matching rows very quickly. (Note that the bitmap index is created on the fly, based on the indexes you've already created on your table.)

### **Setting up Aster Database to use bitmap index scans**

To set up Aster Database to allow bitmap index scanning, do this:

- 1 Define single-column indexes on all columns that you expect analysts to use frequently in their predicates; and
- 2 Contact Teradata Global Technical Support (GTS) to have bitmap index scanning enabled on your Aster Database.

Set up in this way, Aster Database will consider using a bitmap index scan if multiple columns in the predicate are indexed columns. If the planner chooses to use a bitmap index scan, Aster Database creates the bitmap on the fly and uses a bitmapped representation of the indexed columns to quickly look up the desired rows.

### **Notice about enabling bitmap index scans**

It's not always a good idea to enable bitmap index scanning in Aster Database. The main reason is very similar to why indices shouldn't be turned on in all cases – for many slightly complicated queries, the planner might not estimate the cost correctly. If this happens, and if

bitmap index scans are enabled on your cluster, then Aster Database will perform a bitmap index scan when a different type of scan would have been faster. Talk with Teradata Global Technical Support (GTS) to find out what's best for your workload.

## Operational considerations when you use indexes

### Index only after loading

Always wait until after you have loaded data into a table before you define indexes on that table. The presence of the indexes will slow your loading significantly. If you are loading to an existing table, drop the index, load your data, and re-create the index.

Note that the same advice applies to primary key constraints: Do not add primary key constraints to a table until you have finished loading data to that table.

### Remove indexes before bulk deleting

Always drop the indexes of a table before you perform a bulk delete on that table. The presence of the indexes will slow your delete operation significantly.

### Indexes on logically partitioned tables vs. parent/child tables with inheritance

If you are placing indexes on logically partitioned tables you need only create your indexes on the top level partitioned table, because indexes are inherited. Note that this is not true for parent/child tables created through inheritance (the older method which has been retained for backward compatibility). With parent/child tables, indexes are not inherited; you must add the indexes to each child table individually. The typical workflow for creating child tables using parent/child tables with inheritance is to create the table just before you need to load data to it, load the data, and create the indexes.

### Reindexing

Each time you INSERT, UPDATE, or DELETE a significant number of the rows of a table, you should run the REINDEX command to rebuild the table's indexes. This removes fragmentation from the index.

## Tips and FAQs about indexes in Aster Database

### Should I place an index on my distribution key column?

You will recall that, for any large table in Aster Database, you must declare one column of the table as the distribution key that is used to distribute the table's rows in the cluster. You may also recall that, as stated earlier, it's best to choose as distribution key a column that is commonly used in joins or for filtering query results. Given that Aster Database is already distributing rows based on the distribution key column, many administrators ask whether adding an index on the distribution key column will improve performance. The answer is that, yes, you can sometimes gain even faster query performance by adding an index on a distribution key column.

The most clear-cut case for defining an index on a distribution key column is for a table in which the distribution key is also the column most frequently used in the WHERE clause of

point lookup queries. For example, if `user_id` is the distribution key, and your queries frequently look up one or a few user records based on `user_id`, then you should probably create an index on `user_id`. Stated more precisely, if most queries on the table are highly selective, selecting 1% or fewer of the table's rows, then the table probably needs an index on the column being used to select the results, regardless of whether that column is the distribution key.

### Testing the effects of an index

To test whether an index might improve a query's performance, run a test transaction in which you CREATE the index, run an EXPLAIN on the SELECT statement, and then ABORT the transaction. Check the explain plan to see if the plan chose to use an index scan. If so, you can proceed to creating the index and running a test on your test data.

### Planner flag: `enable_seqscan`

If the Aster Database query planner is choosing a sequential scan when you feel an index scan would be faster, then you can set the planner enable flag for sequential scans, `enable_seqscan` to "OFF". For instructions, refer to "[Query Tuning Best Practices](#)" on page 208.

### Match your indexes to your predicates

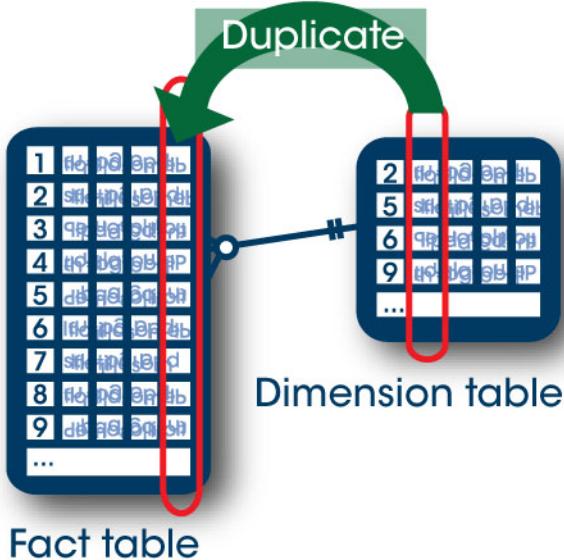
Make sure your indexes match the predicates that your queries are likely to use. This can mean defining multicolumn indexes if predicates frequently filter on certain combinations of columns, or it can mean defining partial indexes (add a WHERE clause to the CREATE INDEX statement). If your predicates contain an expression in the WHERE clause, then you can define an index that applies a matching expression to the indexed values.

## Consider Using a Denormalized Data Model

### Summary

A denormalized data model makes your most frequently queried table wide enough to include all the frequently queried columns, reducing the need to perform joins.

## Denormalize your schema



### How denormalization works in Aster Database

Denormalization in Aster Database schemas typically means column duplication, and specifically we're talking about duplicating a column from a dimension table, placing a copy of it in the fact table. By doing this, we eliminate the need to join against that dimension table to get the contents of that column.

### When to use denormalization

You should consider denormalizing (that is, copying a dimension column to the fact table) if you very frequently join against that dimension to obtain the contents of that column. If you suspect that the join is slowing query performance, then you should test your query using a denormalized schema to see if it runs faster.

The case for denormalizing is more clear cut in the following cases:

- If you always, or nearly always, read the dimension column when you read a fact table row.
- If the dimension table is very large (contains very many rows, particularly if these rows map roughly 1:1 with fact tables).

### When not to use denormalization

If the dimension table column that you're considering duplicating contains frequently updated data, you may want to avoid duplicating the column because it will complicate the management of the data in the duplicate columns.

## Denormalization compared with other techniques

As an alternative to denormalization, you can consider adding indexes that can speed up joins, and, along with that, adjusting the Aster Database planner parameters to make the planner more likely to choose an index scan rather than a sequential scan (full table scan).

# Data Modeling FAQ

### ***Can I use a multi-column distribution key?***

No. See “[Single-Column Distribution Key](#)” on page 189.

### ***Will defining a primary key help improve database performance?***

Typically, no. Aster Database does not consider the primary key when it plans the execution of your queries. Primary keys are useful mainly for ensuring data integrity.

### ***How can I use Aster to improve fact-fact and fact-dimension join times, as compared with my current database system?***

The answer depends on whether you’re currently using an MPP system or a non-MPP system:

- Compared with a *non-MPP* system, Aster Database exhibits superior performance as a result of its parallel processing.
- Compared with *other MPP systems*, Aster Database can typically provide better performance gains if your data model and queries take advantage of one or more of the following features:
  - Aster Database’s ability to support many layers of logical partitioning (also called range partitioning or list partitioning). See “[Split Data Into Child Partitions](#)” on page 191.
  - Faster search of indexed tables (in Aster Database’s distributed model, individual indexes are smaller and can be read faster as a result). See “[Index Your Tables](#)” on page 198.
  - Aster Database’s distribution-aware query optimizer, which can run certain sorting and aggregation tasks in parallel. See “[Distribute Your Data with Joins in Mind](#)” on page 187.
  - Replace costly joins in your queries with Aster Database SQL-MapReduce functions. Well-built SQL-MapReduce functions can often perform much of their work in parallel, as directed by the PARTITION BY clause in the calling query. Teradata Aster provides a number of SQL-MapReduce functions, such as nPath, and you can build custom SQL-MapReduce functions using the Aster Developer Environment (ADE).

# CHAPTER 3    **Query Tuning Best Practices**

This section explains a number of best practices you should follow when designing schemas and managing databases in Aster Database.

- [Introduction](#)
- [Three Principles for High Performance on Big Data](#)
- [Top Tips for Analysts](#)
- [Top Tips for DBAs](#)
- [General Tips on Tuning](#)
- [Optimize Queries for Parallel Execution](#)
- [Read the EXPLAIN Plan](#)
- [A Step-by-Step Approach to Query Tuning](#)
- [Understand Table Joins](#)
- [Use Linux Utilities to Isolate Query Performance Issues](#)
- [Additional Resources](#)
- [Checklist for Query Tuning](#)

## Introduction

### Who should read this guide?

This guide shows you how to make your queries run as quickly as possible on Aster Database. We have organized this document to help two main types of Aster Database users:

- The analyst, who may not have the authority to restructure tables and columns, but who needs to make his or her queries run fast in the table structure that exists; and
- The database administrator, who not only runs and manages queries but also manages and may design the data model of the databases being queried.

The sections “Query Tuning for Analysts: Top Tips” and “Query Tuning for DBAs: Top Tips” address these two audiences in turn, and direct the reader to more detailed tuning procedures, later in this document.

## Am I ready to tune my queries?

To be effective, query tuning must build on good hardware choices, proper Aster Database system setup, and good data model design. The order in which you should tune your cluster is as follows:

- 1 Hardware and network set-up: Make sure that everything from the file system setting to NIC settings to switch settings is appropriate for your Aster Database workload.
- 2 Data modeling: To ensure that queries run quickly, you must choose the proper table structure, including effective use of distributed tables and replicated tables. For help choosing the right Aster Database table and distribution options, see “[Data Modeling Best Practices](#)” on page 176.
- 3 Workload management: If your database cluster supports a mix of different types of workloads (for example, if users are running interactive queries while large amounts of data are being loaded), then you must set up appropriate workload management rules which includes admission control and resource management. See “[Workload Management](#)” on page 268 for details.
- 4 Query tuning: This is the last step in tuning, and the subject of this document. Tuning your queries can be a time-consuming operation because there can be many thousands of individual SQL statements that access the database. If you have carefully optimized the workload as a whole, then you will only need to tune those SQL statements that are performance outliers (that is, those queries that perform worse than expected).

In short, it only makes sense to optimize your queries once the data model has been optimized and the hardware and Aster Database settings have been tuned properly. Without taking those steps first, the effects of your query tuning will be limited. If any of these aspects of your cluster is not yet set up properly, consult the relevant section of the Aster Database documentation.

## Three Principles for High Performance on Big Data

As we mentioned in the section on data modeling, observing the following principles helps to ensure your queries run as quickly as possible:

### Networking: Thou shalt not move big data.

Corollary: If you need to move big data, make it small first, and then move small data.

### Disk: Thou shalt not read irrelevant data.

Corollary: Prepare the data model in advance to ensure that queries touch the least amount of data.

## Processor: Thou shalt not do redundant processing.

Corollary: Prepare your queries such that each computation is done exactly once, and never again.

## Top Tips for Analysts

- 1 When given a choice between using a join and using another construct like a subselect, use the join. Aster Database typically handles joins faster. See “[Understand Table Joins](#)” on [page 232](#).
- 2 Unroll your expression's subselects by moving the subselect from the WHERE clause to the FROM clause, or by rewriting it as a join. This speeds up queries. See “[Rewrite subselect clauses as joins](#)” on [page 216](#).
- 3 Use an inner subquery to shrink the size of data on which outer queries need to operate. Likewise, use an inner subquery to shrink the size of a relation before the data is redistributed on a different distribution key. See the section, “[Redistributed queries](#)” on [page 221](#).
- 4 Avoid per-row function invocation. Group rows first, and then apply the function. See the section, “[Avoid per-row processing](#)” on [page 222](#).
- 5 Avoid doing JOINS, GROUP BYs, and WHERE clauses on non-native data types. See the section, “[Operations on native data types](#)” on [page 222](#).
- 6 EXPLAIN is your friend. Learn to read the Aster Database query EXPLAIN plans to identify potential bottlenecks in executing the query. See “[Read the EXPLAIN Plan](#)” on [page 223](#).

## Top Tips for DBAs

- 1 Run the ANALYZE command after every data load or data update. This will update the table statistics and allow the query planner to choose the best plan. See “[Run ANALYZE regularly to ensure the most optimal query plans](#)” on [page 212](#).
- 2 Use transactions when you modify data (for example, when using CREATE, DROP, CREATE TABLE AS, INSERT, UPDATE, DELETE, or ALTER statements). This allows several statements to run before doing a commit. This reduces the overall execution time as compared with doing several separate commits. See “[When updating or adding data to Aster Database, wrap statements in a transaction](#)” on [page 213](#).
- 3 As mentioned in the list of tips for analysts, use the EXPLAIN command to get to the bottom of performance problems with individual queries. See the section, “[Read the EXPLAIN Plan](#)” on [page 223](#).
- 4 Use the Aster Database AMC (Processes tab) and Ganglia to find and investigate long-running queries. See “[Use Ganglia to find the overworked worker](#)” on [page 211](#).

- 5 Control how ACT handles errors when running a single, multi-statement sql, or a sql file. The “on-error-stop” option can be set to stop running SQL queries. See “[Using the “on-error-stop” Option in ACT](#)” on page 66 and “[Database Parameters Set with \set](#)” on page 80.

## General Tips on Tuning

This section explains the best practices that should guide you when you create queries that will analyze large amounts of data in Aster Database.

### A good data model is the starting point

Tuning is only as good as the data model. Get that right and optimized for Aster Database! Refer to the “Aster Database Hands-On Guide To Data Modeling.”

### Use the AMC and Ganglia to find bottlenecks

#### Use the AMC Processes tab to find bottlenecks

For Aster Database administrators, the Processes tab of the Aster Database AMC is a good place to start your efforts at tracking down long-running queries, or investigating a query that a user is having a problem with.

To find a failed or slow query in the AMC, click the Processes tab, and, in the Processes sub-tab, click the Execution Time column to sort the query history by running time. Click the ID number of a query to view its details. A new Query Inspection tab opens, labeled with the ID number of the query. This tab displays information about each phase of the query. Look for the phase that failed or that took longer to run than you would have expected it to. Armed with this information, you can read the section, “Reading an Aster Database EXPLAIN plan” later in this document to diagnose the cause of the failure or slowness.

Alternatively, you can click the Sessions sub-tab in the Processes tab, and, in the Running Process ID column, click the ID number to open the Query Inspection tab.

On the Process Detail page, the Source column shows you where a query was run or is running. A Source value of AnyWorker means the query runs on one or more workers.

#### Use Ganglia to find the overworked worker

If you have a slow query, check Ganglia ([http://queen\\_ip\\_address/ganglia](http://queen_ip_address/ganglia)) to find the overly busy worker or workers.

On the worker that seems to be the bottleneck, you can use Linux utilities such as the 'top' command to find the process with high CPU usage or high I/O wait times. For a list of flags, see the section, “[CPU Monitoring with TOP](#)” on page 238.

## Run ANALYZE regularly to ensure the most optimal query plans

The Aster Database query optimizer uses table statistics to plan the most efficient path for queries. As you insert into or delete from tables, such as through a COPY operation, UPDATE, INSERT or DELETE statement, these statistics will become out of date unless you run the ANALYZE command. ANALYZE ensures that Aster Database has accurate statistics so that the planner can choose the most appropriate query plan, and thereby improve the speed of query processing.

You should run ANALYZE on the specific table that has changed after the change is made. If your table has child tables, don't forget to include the CASCADE option. In Aster Database Loader, Aster Database makes this easy for you. You can pass –z as a command line parameter to tell Aster Database Loader to automatically analyze the table and all child tables once the load has completed.

However, running ANALYZE after every table write can be expensive, so it is not done automatically after each write. Teradata Aster recommends that you run ANALYZE after every batch of writes so that the statistics are refreshed in bulk.

You should run ANALYZE on the tables that changed after any running of a CREATE TABLE AS SELECT, INSERT, COPY, UPDATE, DELETE, or ALTER TABLE statement. You also need to run ANALYZE after a load operation. Make sure you only run ANALYZE on the tables that have changed – there is no need to run ANALYZE across the entire database. To run ANALYZE, simply type “analyze <tablename>” in ACT or the SQL client utility you are using.

## Run VACUUM to ensure queries achieve best performance

With Aster Database, rows that are deleted or obsoleted by an update are not physically removed from their table. They remain present, but invisible to users, until a VACUUM is done. Having large amounts of deleted rows, called dead space, can impact query performance. Therefore, you should perform a VACUUM operation periodically, especially on frequently updated tables. If your table has child tables, don't forget to include the CASCADE option.

For our analytical use cases, any table that is a target of an UPSERT or DELETE operation or has a failed load or COPY operation should be VACUUMed. Tables that see frequent UPDATE and DELETE operations should be VACUUMed frequently.

See [“Dead Space” on page 168](#) for more details on VACUUM and the various VACUUM options.

## Use a COMPRESS level that matches the frequency of data access

When choosing a level of compression for a table, consider how frequently the data in that table is accessed:

- For hot data, which is accessed frequently, Teradata Aster recommends that you use uncompressed or LOW compressed tables. This ensures maximum performance (fast access times) for queries against hot data.
- For cold data, which is used less frequently, Teradata Aster recommends LOW or MEDIUM compression in Aster Database. Applications or analysts using archived data

should be made aware of the cost of decompression which may result in slower access times when querying such data.

## When updating or adding data to Aster Database, wrap statements in a transaction

To provide availability, Aster Database is designed to maintain multiple copies (usually two) of your data. The maintenance of these copies is called Replication. When you write data to Aster Database, through a load operation, for example, Aster Database first inserts the data into the primary vworker and then replicates a copy of that new data to the secondary vworker. In this way, if you are to lose a worker node, the primary vworker fails over to the secondary vworker and you suffer no downtime or data loss.

Aster Database starts the replication process once a transaction commits. For statements issued outside a transaction block, Aster Database will automatically wrap the statement in a transaction. This means that if you have six statements that write data to the cluster, each statement will be encapsulated within a transaction and six commits will be issued. As a result, six replication threads will be started.

It is much more cost effective to batch write operations together into one transaction, when possible. However, when using longer transactions, the operations tend to hold locks longer, which will reduce concurrency. So there is a trade off, and decisions about how long your transactions should be must be considered in light of the benefits and drawbacks.

For example, you want to execute the following sequence of statements:

```
CREATE FACT TABLE t1
(
    a INT,
    b INT,
    PARTITION KEY (a)
)
AS
    SELECT s1.a, s2.b
    FROM s1, s2
    WHERE s1.a=s2.a;

INSERT INTO t1 (b)
    SELECT b
    FROM s2
    WHERE a IS NULL;

ANALYZE t1;

CREATE FACT TABLE t2
(
    a INT,
    count INT, PARTITION KEY (a)
)
AS
    SELECT t1.a, COUNT(DISTINCT t1.b) AS count
    FROM t1
    GROUP BY t1.a;

ANALYZE t2;
```

If you execute the above five statements, they are treated as five separate transactions. As a result, Aster Database has to execute five cluster-wide transaction commits and replication for each.

If you combine the five statements into a single transaction, Aster Database executes only one transaction commit and replication.

```
BEGIN;

CREATE FACT TABLE t1
(
    a INT,
    b INT,
    PARTITION KEY(a)
) AS
    SELECT s1.a, s2.b FROM s1, s2 WHERE s1.a=s2.a;

INSERT INTO t1 (b)
    SELECT b
        FROM s2
        WHERE a IS NULL;

ANALYZE t1;

CREATE FACT TABLE t2
(
    a INT,
    count INT,
    PARTITION KEY(a)
) AS
    SELECT t1.a, COUNT(DISTINCT t1.b) AS count
        FROM t1
        GROUP BY t1.a;

ANALYZE t2;

END;
```

## Optimize Queries for Parallel Execution

In this section, we show how you can write queries that “push down” processing to as many workers as possible in order to take advantage of the Aster Database’s parallel processing architecture.

### Join tables on the distribution key for best performance

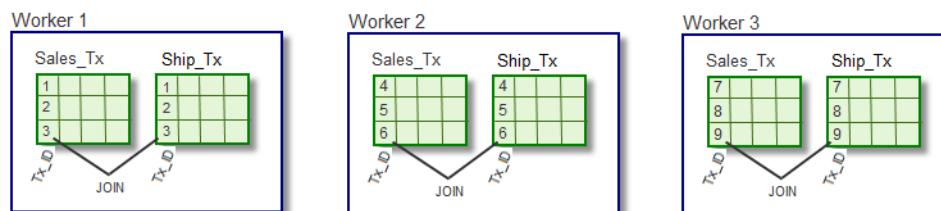
Aster Database is optimized for joins. A query that joins two distributed tables on their distribution key column is highly optimized, and Aster Database executes the query locally on each Worker node. This is very important. A local join will perform better than a join that requires Aster Database to move data between worker nodes.

For example, the following distributed fact tables, `sales_tx_fact` and `ship_tx_fact`, are both distributed on the transaction ID column (`tx_id`). When calling queries that join on the

`tx_id` column between these two tables, the join is performed locally within each worker node.

```
SELECT customer_id
FROM sales_tx_fact sales, ship_tx_fact ship
WHERE sales.tx_id = ship.tx_id and return_date is NULL;
```

The diagram below shows the join:



Join on `Tx_ID` is completely local to the worker. No data is moved between workers.

But, what happens when you write a query that does not join on the distribution key?

If the query does not join the distributed tables on the distribution key column, Aster Database exchanges data between, or re-distributes, the workers to perform the join. This data movement process is known as ICE (Inter-Connect Express). Although Aster Database is optimized to make ICE as fast as possible, it can be an expensive operation, especially if very large amounts of data have to be moved. Remember the first principle for high performance on big data: *Thou Shalt Not Move Big Data* (over the network). One of your goals in tuning queries is to ensure that Aster Database redistributes only when absolutely necessary.

When Aster Database redistributes the table to use the new distribution key, it stores this new table as a temp table. Once the transaction is completed, this temp table is dropped.

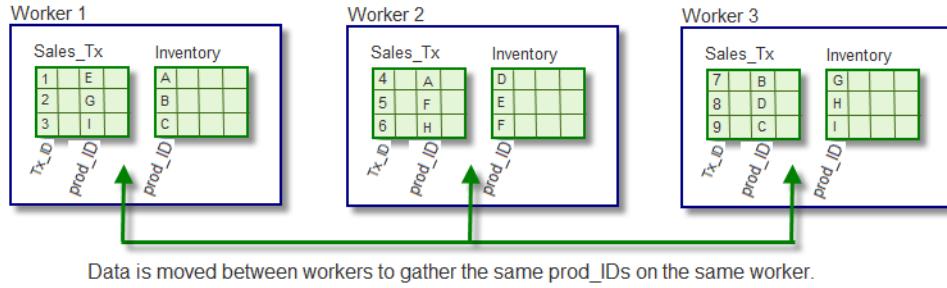
Using our `retail_sales` database, let's look at another example. The fact table `inventory_fact` tracks all inventory details for the retail chain and is distributed on the `product_id` column.

You want to run a query that will tell you all customers that have returned products. For this query, you join on the `product_id` column, which is not the distribution key for `sales_tx_fact`.

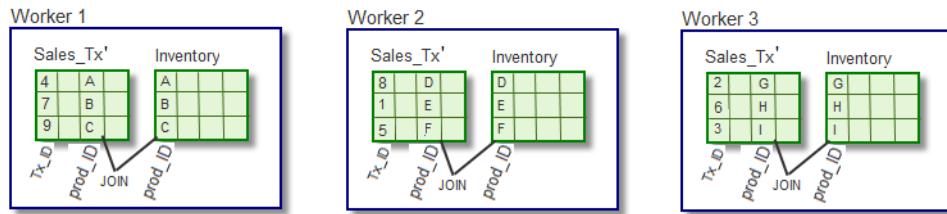
```
SELECT s.product_id, s.customer_id
FROM sales_tx_fact s, inventory_fact i
WHERE s.product_id = i.product_id
AND i.quantity_returned_by_customer > 0;
```

When this query executes, Aster Database first redistributes the data by the join column. This ensures that each worker can perform a local join on the `product_id` column.

### Step 1:



### Step 2:



Once repartitioning (ICE) occurs so that the same prod\_IDs across the two tables are all on the same worker, each worker performs a local join.

Note the tick mark on the sales\_tx\_fact table, this shows that it is actually a temporary table that Aster Database is joining to.

Whenever possible, write your queries to join on the distribution key. Always optimize for a local join and avoid queries that require redistribution, if possible.

## Rewrite subselect clauses as joins

There are many ways to write queries. With Aster Database, which is optimized for joins, it is almost always better to write queries using joins. Whenever possible, rewrite queries to perform joins, and especially local joins.

For example, take our retail\_sales data warehouse and write a query that shows all customers that have returned products.

One way to write this query is like this:

```
SELECT customer_id FROM sales_tx_fact
WHERE tx_id in (SELECT tx_id FROM ship_tx_fact
WHERE return_date IS NOT NULL);
```

Here is how the Aster Database Planner breaks down this query.

- 1 The Planner executes the inner SELECT query and stores the resulting temp table at the Queen
- 2 The Planner then pushes the temporary table to all workers via ICE.
- 3 The planner then executes the outer SELECT query as a local JOIN operation to the temp table.
- 4 The result set from each worker are assembled at the queen and sent to the user.

This is inefficient as Aster Database creates several tables and there is unnecessary data traffic to/from the queen.

Rewrite this query as follows:

```
SELECT customer_id FROM sales_Tx_fact sales, ship_tx_fact ship  
WHERE sales.tx_id = ship.tx_id AND return_date IS NOT NULL;
```

Here is how the Planner breaks down this query.

- 1 The Planner executes a local join on each worker node on the column tx\_id.
- 2 The planner executes the SELECT query on the joined table on each worker.
- 3 The result set from each worker are assembled at the queen and sent to the user.

As you can see, this is far more efficient. There is no ICE required and no temp table creation, except for the final result set.

Whenever possible, rewrite subselect clauses as joins. Aster Database is optimized for joins. Even if the join is not on the distribution key, it is still going to be better than a subselect statement.

The example above helps demonstrate why Aster Database recommends that you replicate smaller dimension tables instead of distributing them. By having the entire dimension table present on the worker node, you make local joins more likely. Local joins run faster than non-local joins because less data is exchanged between workers.

## Use GROUP BY and COUNT DISTINCT on the distribution key

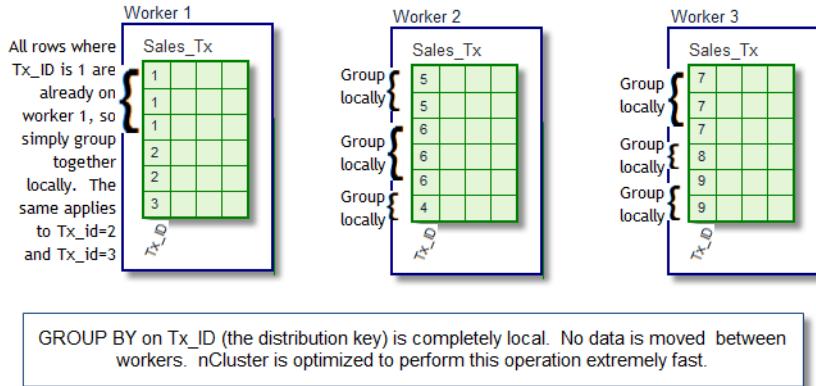
Aster Database is very fast at performing GROUP BYs on the distribution key, including when the distribution key is in the list of GROUP BY values. You can use this property of Aster Database to speed up many queries.

The reason Aster Database is fast doing GROUP BYs on the distribution key is because with the distribution key, the planner knows that all the same values for that column are grouped on the same worker. Therefore, there is no sorting or shuffling (ICE) that needs to occur to bring all similar values to a single worker or the queen. Each worker contains all the rows for a unique distribution key value. As a result, a GROUP BY on the distribution key can be executed, immediately, across all workers in parallel.

The good news is that even if you have multiple expressions in your GROUP BY clause, as long as one of them is the distribution key, you will get optimal performance.

Let's look at an example:

```
SELECT tx_id, count(*) FROM sales_tx_fact GROUP BY tx_id;
```



As you can see above, since `tx_id` (the transaction ID) is the distribution key, the Aster Database planner simply performs the GROUP BY locally, on each worker, in parallel. It knows that all the `tx_ids` of the same value will be on the same worker. This makes the operation extremely fast.

In addition, Aster Database is very fast at performing COUNT DISTINCTs on the distribution key. The reason for this is that, again, the planner knows that all the same values for that column are grouped on the same worker. As in the previous example, where the distribution key column is the `tx_id` column, all the rows whose `tx_id` value is “1” reside on *Worker A*, all rows whose `tx_id` value is “6” reside on *Worker B* and so on (see diagram above). *Worker B* can never contain a row whose `tx_id`, the distribution key, has a value of “1”. Therefore, the Aster Database planner knows that each COUNT DISTINCT performed on the distribution key will result in the complete set of distinct values from each worker. There is no need to redistribute (move data between nodes). This operation can be performed completely in parallel across all workers simultaneously and is very fast.

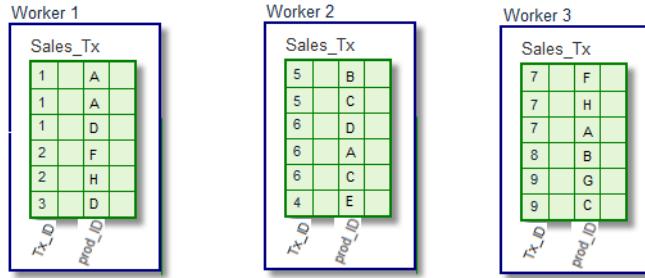
Note: Running a COUNT DISTINCT or GROUP BY on a distribution key cast to a different datatype results in different behavior. For example, if the distribution key is cast to an int, the GROUP BY or COUNT DISTINCT will be slower than if the distribution key is used as is, and the distribution applied may not be the same as is seen with the original datatype.

It is not always possible to GROUP BY the distribution key, so let's look at what happens when you GROUP BY a column that is not the distribution key. For this example, let's look at this query on our example data set:

```
SELECT product_id, COUNT(1) FROM sales_tx_fact GROUP BY product_id;
```

This query performs its GROUP BY on `product_id`, which is NOT the distribution key of the table `sales_tx_fact`.

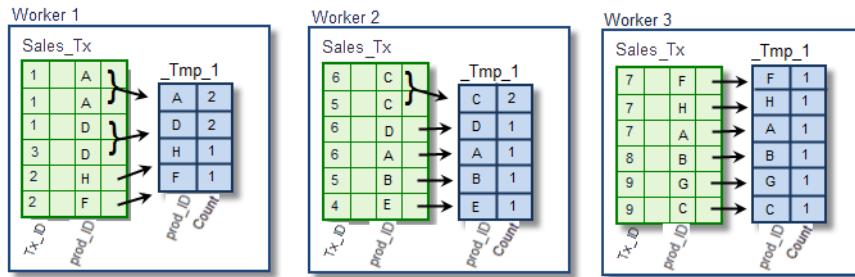
Here is an illustration of the dataset before the query is issued.



The Sales\_Tx\_Fact table distributed by Tx\_ID across 3 workers. Both the columns for Tx\_ID and Product\_ID are shown.

When you issue this query, you simply get a result set. Behind the curtain, however, Aster Database performs several actions to optimize performance of this type of query.

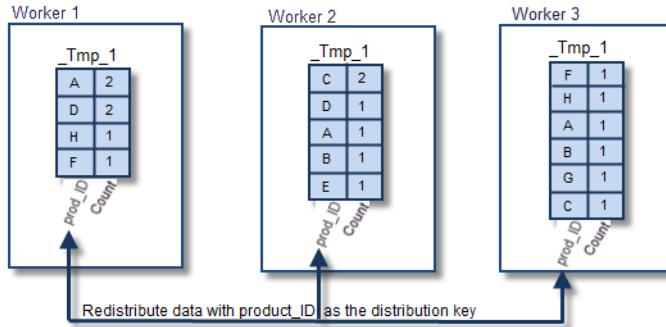
- 1 Aster Database first performs a GROUP BY operation on product\_id locally. This happens on each worker node simultaneously. Aster Database then performs aggregation locally again simultaneously across all workers. In our example, this means the planner gets the count of the number of sale transactions for each product. This result set is stored locally on the worker in a temp table.



Each worker performs a GROUP BY on product\_ID locally and then does a count(1) storing the results in a temp table. This "pre-aggregating" is one way nCluster speeds queries and maximizes parallelism across the cluster.

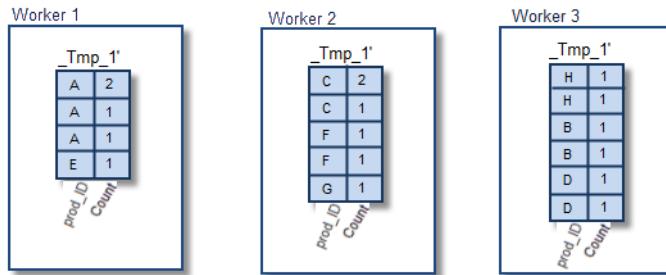
- 2 Then, Aster Database redistributes this temp table by the last column or expression in the GROUP BY clause. In our case, there is only one column (product\_id) thus the temp table is redistributed on the product\_id. This ensures that all the pre-aggregated sets of data that form a particular group end up on a specific worker.

The first part of this step:



nCluster repartitions these pre-aggregated temp tables by the GROUP BY column, product\_ID. This will gather all the same product\_ID totals to the same worker

The second part of this step:



With the repartitioning, nCluster creates new temp tables and stores the newly partitioned data. Now that all product\_IDs are grouped onto the same local worker, nCluster can simply sum the counts for each group.

- 3 Then, Aster Database does a sum over the groups on the workers and streams the results up to the queen to assemble and return the result set.

You can see this is a smart way to do this because it minimizes network traffic by pre-aggregating locally.

In query tuning, it is important to understand which column or expression Aster Database will redistribute on as it creates its plan and prepares to execute the query. If Aster Database uses a good column, where the data has good distribution, the performance will be good. If Aster Database uses a poor column (one that has bad skew), however, this can cause poor performance.

One very important item to be aware of, in a query with a GROUP BY clause, where the GROUP BY clause does not contain the distribution key, Aster Database always redistributes by the last column or expression in the GROUP BY clause.

If your “GROUP BY on non-distribution key” query is running slowly, validate which column Aster Database is redistributing on. If it is a column with known skew or low cardinality, this is likely the cause. Often, you can improve the performance by simply changing the GROUP BY clause order so that the best column for redistributing is listed *last* in the clause.

For example, let’s say you have a query:

```
SELECT product_id, COUNT(1) FROM sales_tx_fact  
GROUP BY product_id, basket_id;
```

Based on what you've learned above, you know that when Aster Database executes this query, it is going to redistribute the pre-aggregated data by `basket_id`, because this is the last column in the GROUP BY clause. Let's say, however, that `basket_id` has a lot of skew because, when a transaction doesn't include a `basket_id`, it simply puts a "-1" in that column. And, you have a disproportionately large number of "`basket_id = -1`" records in the system. Aster Database will not perform optimally on this query because it is redistributing on a column that has skew.

This is very easily resolved by simply swapping the order of the GROUP BY columns so that `product_id` comes last. The `product_id` column does *not* suffer from data skew, so Aster Database will perform optimally with this query:

```
SELECT product_id, COUNT(1) FROM sales_tx_fact  
GROUP BY basket_id, product_id;
```

Note that now `product_id` is the last column in the GROUP BY clause and thus, Aster Database will redistribute on `product_id`. This will have better performance because `product_id` is a good column without skew.

### Counting DISTINCT values in a non-distribution key column

When you perform a COUNT DISTINCT (with/without a GROUP BY in your query), if the definition of a distinct item includes the table's distribution key column, then Aster Database can divide the task and distribute it among the workers for faster performance. (This is also known as "pushing down" the task.) Performance may be somewhat slower when you count the distinct values in a column that is *not the distribution key*. This is a good example of a query that exhibits good performance in an SMP system, such as MySQL or Oracle, but runs more slowly in an MPP system like Aster Database. It is important to understand what slows the performance of such queries so that you can optimize them to run in Aster Database. For example, let's assume you want to count the distinct `product_id` values using the following query (remember that the `product_id` is *not* the distribution key of `sales_tx_fact`):

```
SELECT COUNT (DISTINCT product_id) FROM sales_tx_fact;
```

Each worker issues a local query:

```
SELECT COUNT (product_id) FROM sales_tx_fact GROUP BY product_id;
```

In other words, locally at each worker, Aster Database performs a SELECT COUNT with a GROUP BY. Next, Aster Database collects the distinct values from each worker and ships them over the network by hashing on the expression in the GROUP BY list, so that all the duplicate values are guaranteed to be placed on the same vworker (in a temporary table). Aster Database performs a COUNT DISTINCT on that data, and finally it calculates the global sum of the distinct values at the queen.

## Redistributed queries

Use the inner subquery to shrink the size of a relation before the data is redistributed on a different distribution key.

For example, this query:

```

SELECT
    i.product_id,
    SUM(s.sales_quantity*(i.unit_list_price-i.cost)-s.discount_amount)
        AS net_profit
FROM
    sales_tx_fact s,
    inventory_tx_fact i
WHERE s.product_id = i.product_id
    AND s.sales_date > '01-Jan-2011'
GROUP BY i.product_id;

```

can be rewritten as:

```

SELECT
    i.product_id,
    SUM(t.total_sold*(i.unit_list_price-i.cost)-t.total_product_discount)
        AS net_profit
FROM
(
    (
        SELECT
            s.product_id as product_id,
            SUM(s.sales_quantity) AS total_sold,
            SUM(s.discount_amount) AS total_product_discount
        FROM sales_tx_fact s
        WHERE s.sales_date > '01-Jan-2011'
        GROUP BY s.product_id
    ) t,
    inventory_tx_fact i
WHERE t.product_id = i.product_id
GROUP BY i.product_id;

```

In the rewritten query above, the GROUP BY serves to speed up the query because it ensures the grouping of rows is performed in a distributed fashion on the workers before the product\_id values only are passed to the queen for filtering of the distinct values.

## Avoid per-row processing

Avoid per-row function invocation. Instead, group rows together to shrink the data size, and then apply functions on the smaller number of values.

## Operations on native data types

Avoid doing JOINS, GROUP BYs, and WHERE clauses on non-native data types (e.g., numeric, varchars, char arrays). Instead, always try to run these operations on native data types (ints, floats, doubles, char) and then convert to non-native data types.

## Sorting vs. hashing

When tuning the workers, it's useful to keep in mind the duality of sorting and hashing. Whatever can be done using sorting can be done using hashing, too. What you choose (or more precisely, what you would like the worker instance to choose) depends on the characteristics of the operation, available memory, and so on. If you lived in a world in which the estimates were always correct, then you would always use hashing if that was possible. But when there is a misestimation, hashing fails more miserably than sorting does.

## Favor GROUP BY over DISTINCT

If you can express the same using GROUP BY or DISTINCT, use GROUP BY. This gives the Aster Database query planner more options to choose from.

# Read the EXPLAIN Plan

Before we discuss the advanced techniques in “[A Step-by-Step Approach to Query Tuning](#)” on [page 227](#), it is important to understand how to read query plans in Aster Database, and how to generate them using the EXPLAIN command.

## Introduction to EXPLAIN

EXPLAIN is a database command that, given a query, prints the execution plan that the Aster Database query planner and the Postgres query planners have chosen for that query. Running EXPLAIN provides hints that can help you tune your queries, and the EXPLAIN output can also be helpful for debugging multi-step transactions because it performs a syntax check of each SQL statement.

EXPLAIN shows the estimated cost (startup cost and total cost) for each of the phases in the query’s execution plan. Phases in the plan are organized into tasks and sub-tasks called “nodes.” Cost values are shown for each node. The cost shown for a parent node includes all the costs of its children, grandchildren, and so on.



**Tip!** When Aster Database chooses a query plan, the decisions that divide the overall plan into phases are *not* cost-based, but the choice of each execution within a phase *is* cost-based.

EXPLAIN also estimates the size of the result set and shows the network data transfers involved (if any), and the purpose of each. Pay close attention to the estimated result set sizes. Misestimating these is one of the planner’s most common errors.

Note that PostgreSQL also supports a command called “EXPLAIN ANALYZE” that actually runs the query being analyzed and reports statistics about the actual runtime cost in order to judge the accuracy of the planner’s estimates. EXPLAIN ANALYZE is not supported in Aster Database, but Teradata Global Technical Support (GTS) can perform detailed performance profiling if you have a query you suspect is not performing well. Contact Teradata Global Technical Support (GTS) for details.

## The EXPLAIN Plan

To see the EXPLAIN plan for a query, enter the keyword “EXPLAIN” before your SQL statement in ACT (or other query tool, like Aqua Data Studio), and then issue the statement. Aster Database prints the EXPLAIN plan and does not run the query. EXPLAIN works on SELECT statements and other DML statements, but not on DELETE statements.

## How does EXPLAIN work?

There are two levels of EXPLAIN used with Aster Database. The main plan is the parallel execution plan, often referred to as the “queen explain plan.” This is the plan that the queen will execute to complete the query. Next are the Postgres execution plans, often referred to as the “vworker explain plans.” These are the plans the vworkers will follow in order to execute the queries that have been sent to them by the queen. You can examine the queen-level plans yourself. If you suspect a problem in an individual worker, Teradata Global Technical Support (GTS) can check vworker-level explain plans for possible inefficiencies in processing.

## What happens behind the scenes?

When you run a query in Aster Database, the cluster components do the following:

- 1 Queen parses the query and does syntax checking.
- 2 Queen creates the parallel execution plan.
- 3 Queen sends the SQL to the vworkers to process.
- 4 vworkers send back the results.
- 5 Queen assembles and returns the finished query results.

## Tips for using EXPLAIN effectively

Tricks for making the plan readable in ACT:

- 1 EXPLAIN prints wide columns of output that are too wide to read easily on most screens. A quick way to view more readable output is to switch the formatting of ACT before you run EXPLAIN. To do this, type `\x` at the ACT prompt. This turns on 'expanded output,' which pivots the results of the display to show each column on a new row, providing a cleaner view. See “Formatting-related commands in ACT” in the Aster Database User’s Guide.
- 2 Always invoke ACT with the `-h <queen hostname>` argument. (Use the queen’s actual hostname or IP address; never use "localhost" as the queen hostname, even when running EXPLAIN from the queen node.) If you fail to provide the queen’s hostname or IP address, you get a very verbose explain plan. With `-h`, it gives you a “local” plan which provides a better picture of the real plan.
- 3 When you generate an EXPLAIN plan in an ACT session that is running directly on the queen, it provides more detail than you would get in an ACT session running on your local workstation.

## Read an EXPLAIN plan

Aster Database EXPLAIN produces the following columns of output:

- Number: This is the number of the high-level phase.
- Statement: The actual statement being executed at the individual workers.
- Result Table: If the statement being executed generates output that will be used later, this goes into the table named here.

- Operation Type: This column describes the kind of operation being performed. The operation can be one of the following:
  - Pre-condition
  - Actual statement execution
  - Repartition tuples <reason for redistribution> (Column <n>) - This indicates that the rows ("tuples") are being shuffled across different workers. The value *n* is the column number (one-based) which is being used as the distribution key for this operation. The reason for redistribution will be one of:
    - Repartition tuples and populate table
    - Repartition tuples for subselect
    - Repartition tuples for aggregation
    - Repartition tuples to satisfy join
  - Broadcast tuples <reason for broadcast> - This indicates that the result set of rows ("tuples") is being sent to all the workers. The following are the cases that could cause a broadcast of rows:
    - Broadcast tuples and populate table
    - Broadcast tuples for subselect
- Location: This column indicates where this phase is being executed. This can state either Queen, Workers, or AnyWorker. A value of "Workers" means the phase runs on all workers, and "AnyWorker" means it runs on any single worker.
- Query Plan and Estimates: This column displays the vworker-level query plan and cost estimates for the corresponding phase, and you read it from the bottom up. Note that all costs are relative. The basic unit of cost is a decimal value where one unit represents the cost of fetching 8 KB of sequential data from the disk. The cost of transferring 1 KB of network data is set to 1. The first line of the output displays the summary of the entire query. Here is how to interpret the query plan and estimates in the EXPLAIN output for phase one:

```
localCost=0.00..34.00 rows=9600 width=4 networkCost=37
```

The cost of getting the first row is 0. The cost to read all the rows is 34.00. The number of rows returned will be 9600. The average width of the row is 4 bytes, and the network cost to transfer these rows from the workers to the queen is 37.  
From the second line onward, the output is from the node where the query is going to be executed. If the query is one that will run on multiple workers, then we display the statistics from the slowest worker. You can see which scan, sort, or join algorithms will be applied at the nodes.
- Data Size Distribution (in bytes): This column gives an estimate of the mean and standard deviation of data coming from worker nodes. This is only applicable for queries that require transfer of data, either from the worker nodes to the queen node or among the workers.

## A simple example

Let's look at a simple example of EXPLAIN in action:

```
retail1=> EXPLAIN SELECT COUNT(*) FROM sales_tx_fact;
```

Typing this command will print the parallel execution plan that the queen will use to process the query. The displayed plan also shows the detailed plan to be executed by the vworkers.

## Overall plan

Here's what the plan looks like, when run directly on the queen:

```
retail1=> EXPLAIN SELECT COUNT(*) FROM sales_tx_fact;
Number | Phase      | Description
-----+-----+
1 | TransferPhase| Statement: SELECT count( 1 ) AS "_c5" FROM ( SELECT 1 AS "_c2" FROM "public"."sales_fact" AS "projInp_" ) AS "aggregateInp_";
: Source: Workers Partitioned ;
: Destination: Queen ;
: Table: _tmp_0 ;
: PersistType: Transient ;
:
2 | Query        | Statement: SELECT sum(  "_c5" ) AS "count(1)" FROM "_tmp_0" AS "aggregateInp_";
: Location: Queen ;
:
```

### Step 1 of the plan

The details of Step 1 (Transfer Phase) of the plan are:

- Statement: The SQL that is sent to each vworker
- Source: In this case, the vworkers
- Destination: This is an aggregate query with no intermediate steps, so the destination is the queen.
- Table: There is a temporary table (\_tmp\_0) built on the queen to store the results.
- Persistent: Transient indicates that the table \_tmp\_0 will be dropped when the query completes.

### Step 2 of the plan

Step two of the plan is:

```
SELECT sum(  "_c5" ) AS "count(1)" FROM "_tmp_0" AS "aggregateInp_"
```

This is the query that the queen will execute to get the result set to send back to the user. When the query is completed, the temp table \_tmp\_0 will be dropped.

## EXPLAIN plan for a large query

The Queen's explain plan for a large query looks like this example:

```
1 | TransferPhase| Statement: SELECT BIG QUERY;
: Source: Workers Partitioned on column(s) [transaction_id] ;
: Destination: Workers Partitioned on column(s) [customer_id] ;
: Table: _tmp_0 ;
: PersistType: Transient ;
:
2 | TransferPhase| Statement: SELECT BIG QUERY ;
: Source: Workers Partitioned ;
```

```
: Destination: Workers Partitioned on column(s) [transaction_date] ;
: Table: _tmp_1 ;
: PersistType: Transient ;
:

...
[repeats five times]

8 | TransferPhase|FINAL BIG QUERY ;
: Location: Queen ;
```

This query has seven UNION ALL operators. Note that the queen sends the seven different queries to each of the vworkers. There is also data motion (redistribution), after which the final query is processed at the queen.

## Summary

In summary, when you examine the explain plan output for a problematic query, you should ask:

- 1 Are any of the phases taking an inordinately long amount of time? If so, consult the checklist in “[Top Tips for Analysts](#)” on page 210 to see how the query might be rewritten.
- 2 Is the network cost of the phases too high? The network often is the biggest bottleneck, so queries should be written to minimize network traffic. For tips, see “[Optimize Queries for Parallel Execution](#)” on page 214
- 3 Is the standard deviation of data really high? This could indicate a large data skew. If you suspect data is distributed across workers very unevenly or in a way that does not match your typical join operations, see “[Distribute Your Data with Joins in Mind](#)” on page 187.
- 4 Are there signs of dynamic data skew or planner skew? If so, see “[Find data skew](#)” on page 242.

# A Step-by-Step Approach to Query Tuning

This section presents a step-by-step approach to finding and eliminating bottlenecks that slow query execution.

## Eliminate unnecessary data movement

Even the fastest network can be a bottleneck. Data movement (called “redistribution”) is typically the most expensive operation, and reducing it will give you the most bang for the performance-tuning buck. The Aster Database process used to move data between workers, called Inter-Connect Express (ICE) is optimized to be as fast as possible. Still, if your query shuffles a lot of unnecessary data between workers, it is not going to perform as well as it could. This is the fundamental reason for the Aster Database planner’s existence: to reduce network utilization as much as possible. To identify areas of potential unnecessary redistribution, look at the Aster Database EXPLAIN plan and search for phases labeled, `Repartition tuples`. These are the clauses where redistribution happens. If you have designed your data model well, there should be only minimal redistribution for most queries.

## The three types of queries in which redistribution may happen

There are three types of queries in which redistribution can happen:

- 1 Joining two or more distributed tables (INNER JOIN, JOIN, OUTER JOIN, NATURAL JOIN).
- 2 Aggregating on a fact table over a column or expression (GROUP BY, DISTINCT)
- 3 Compounding multiple queries together (e.g. UNION, INTERSECT)

When performance tuning your queries, pay attention to these types of queries and identify whether there is unnecessary redistribution happening.

## Some operations cannot be done locally, per worker

Redistribution occurs when joins or aggregations cannot be executed locally. This is either due to poor data modeling or the query is the 20% uncommon-case query that requires a redistribution. Think of redistribution as a “prerequisite step” for query execution where the data required for the join is moved so that it is distributed on the attribute/expression in the join or the aggregation. We’ll use the following query as an example:

```
SELECT product_id FROM sales_tx_fact GROUP BY product_id;
```

We know that `sales_tx_fact` is distributed on the `tx_id` column. This GROUP BY clause is on column `product_id`. If you recall what we discussed in “[Use GROUP BY and COUNT DISTINCT on the distribution key](#)” on page 217, you know that Aster Database will redistribute `sales_tx_fact` on `product_id` to perform the GROUP BY.

The planner will take the following steps to execute the query above:

- 1 Aster Database will perform a GROUP BY locally on the `product_id` column on each worker and store the result set in a temporary table. This is a pre-aggregated set of `product_ids`. By pre-aggregating locally first, Aster Database reduces the amount of data that needs to be redistributed.
- 2 Aster Database will redistribute the temporary table of pre-aggregated `product_ids`, distributing on the `product_id` column. At the completion of the operation, each worker will have all of the data for a specific set of `product_ids`. In other words, it is not possible for *Worker A* to have `product_id=1` and *Worker B* to have `product_id=1`.
- 3 Lastly, each worker will perform a local sum over the records in the newly distributed table and pass those results to the queen.
- 4 The queen assembles the result set and streams to the user.

In the example above, there is no other way to write the query. This query with the data model we are using requires redistribution. Remember the rule “Thou Shalt Not Move Big Data.” Even though you cannot optimize the query above, the Aster Database planner is smart enough to optimize for you, behind the scenes, by doing multiple local aggregations before redistribution or passing data to the queen.

## When the query requires redistribution

If redistribution is indeed required, first check to see whether the fundamental goal of the planner is satisfied: Has the planner been able to reduce network utilization? Draw a block

diagram that shows what data needs to go where for the query. Try to deduce the minimal amount of data that would be required to satisfy the query. For example, “SELECT product\_id FROM sales\_tx\_fact GROUP BY product\_id;” does NOT need to have the entire sales\_tx\_fact table redistributed on product\_id so that the GROUP BY can be executed locally. It is enough if only the unique product\_id values are redistributed.

Second, make sure that your query isn’t returning unnecessary data that could slow the query. For example, if all you need is the product\_id, then writing the query as “SELECT \* FROM sales\_tx\_fact GROUP BY product\_id;” will cause the entire fact table sales\_tx\_fact to be redistributed, which can be significantly more data and cause it to run significantly slower.

### **Tip: GROUP BY uses the last column**

When the planner has to redistribute for a GROUP BY aggregation, then it always picks the last column/expression! Make sure that column/expression produces values with a good amount of cardinality (is not a Boolean value, for instance).

### **Why is the planner doing that?**

If the Planner is not doing what you expect, contact Teradata Global Technical Support (GTS) for help in understanding the query plan.

## **Examine scan-type operations**

### **Types of scans**

For scan type of operations, there are typically only three possible operation types by which a table can be accessed:

- Sequential scan (the most generic and scalable scanning option, applicable to data of any size)
- Index scan (only if indexes are available AND the query specifies the column being indexed AND the Postgres optimizer thinks that the query’s filtering on the column will select out < 10-20% of the base fact table) (Note: Again, what matters is what Postgres thinks, so remember: ANALYZE ANALYZE ANALYZE!)
- Bitmap index scan (if multiple indices are available AND the query filters on columns on which these indices have been constructed AND the Postgres optimizer thinks that the query’s filtering on these columns will select out <10-20% of the base fact table).

For information on using indexes and testing their effects on query speed, see the section, “Index Your Tables” in the Aster Database Hands-On Guide To Data Modeling.

### **Scan hints: enable\_seqscan, enable\_bitmapscan, etc.**

Scan operations are controlled by the Postgres parameters/hints of enable\_seqscan, enable\_bitmapscan, random\_page\_cost, and occasionally effective\_cache\_size. The default settings for these are:

- enable\_seqscan='on'
- enable\_bitmapscan='on'

- `random_page_cost=40`
- `effective_cache_size=2GB`

You should not change the planner hint values, except within a transaction; always set them inside a `BEGIN; ... END;` transaction block.

In a production setting, you will rarely turn off `seqscan`, and you would rarely tune `random_page_cost` or `effective_cache_size` (ideally, this last parameter should have been set right at the time of deployment based on the estimated available memory). For testing though, you may want to change `random_page_cost` and `seqscan` temporarily.

In a production setting, the `random_page_cost` setting should be truthfully set to a value indicated by the disks' hardware statistics. The value should be proportional to 'the ratio of (random IOPS)/(sequential IOPS) for that disk'. In practice, it is hard to estimate these numbers. A good initial `random_page_cost` value that is frequently used in production clusters is 40. This is the default setting.

Sometimes, you may wish to temporarily set the `random_page_cost` setting to "1" to experiment with plans. For example, you might do this to force the planner to choose an index scan instead of a sequential scan, just to see how long the query takes using the index scan.

To change the planner hint values within a transaction, use the syntax shown below:

```
BEGIN;
SET enable_seqscan TO 'off';
SET enable_indexscan TO 'on';
SET random_page_cost TO '1';
SELECT ...
END;
```

Note: These planner variables can only be changed on a live cluster inside a transaction. The changes live for the duration of the transaction. If you need to make a more permanent change, please contact your Aster Database DBA or Teradata Global Technical Support (GTS).

The full set of hint parameters is as follows:

```
SHOW client_encoding;
SHOW cpu_index_tuple_cost;
SHOW cpu_tuple_cost;
SHOW effective_cache_size;
SHOW enable_bitmapscan;
SHOW enable_hashagg;
SHOW enable_hashjoin;
SHOW enable_indexscan;
SHOW enable_mergejoin;
SHOW enable_nestloop;
SHOW enable_seqscan;
SHOW maintenance_work_mem;
SHOW random_page_cost;
SHOW seq_page_cost;
SHOW statement_timeout;
SHOW work_mem;
SHOW cluster_status;
SHOW default_schema;
SHOW search_path;
SHOW enable_statement_atomicity;
```

### **Estimates of filter effectiveness**

How does the query planner estimate how much data a filtering condition would filter out? In other words, given a condition like "WHERE a > 10" how does the planner estimate that <10% of the base fact data set is selected and so it is time to use the index on 'a'? In general, the planner never "knows" but it tries to make an informed guess - informed by the frequency distribution of values in that column. In most cases, this is good enough. In odd cases, the planner's estimate may be wrong, so always look for discrepancies between the estimated and the actual row count for each filtering operation.

### **Primary key indices are ignored for planning**

Primary key indices are NOT used for querying.

### **Never load into a table with indexes**

Index maintenance is an expensive operation and is kicked off every time you load into a table. For this reason, do not perform large-scale loads into/updates in/deletions from a table that has an index or indices. Consider dropping the index, doing the desired load/update/delete, and then recreating the index.

## **Examine join operations**

There are three ways of doing a join between two or more tables:

- 1 Nested loop joins work well when Postgres thinks both the tables are really small OR one table is much, much smaller than the other AND the second table has an index on the joining column.
- 2 Hash joins work well when neither table is small but the smaller table's joining column can be used to construct a hash table that Postgres thinks will fit in `work_mem` amount of memory.
- 3 Merge joins work well if the tables are large and Postgres thinks they cannot fit in `work_mem` amount of memory.

For information on tuning Aster Database to use efficient join types, see "["Understand Table Joins" on page 232.](#)

## **Examine aggregation operations**

For aggregation type of operations, there are only two different ways in which it can be carried out:

### **Aggregation operation types**

- Hash aggregation (works for not too large tables when Postgres thinks the unique values of the column/expressions being GROUPed on can fit in `work_mem` memory)
- Sort aggregation (works for all large tables whose unique column values Postgres thinks cannot be fit into `work_mem` memory)

### The `work_mem` value matters, again

In choosing the aggregation operation type, the most important Postgres parameter is again `work_mem`.

Be aware that if Postgres chooses a hash aggregation because it estimates that the hash table will fit in `work_mem` memory, but the hash table turns out not to fit in main memory, then disk swapping will occur on the affected worker nodes. The same advice applies for the sort aggregation method.

### Why does the planner think that?

In the sections above, we have repeated the advice that, in choosing a query plan, the planner must rely on its knowledge of the tables and cluster. In other words, “What matters is what the planner thinks.” If you suspect poor plans are being chosen, it’s a good idea to find out why the planner thinks the way it thinks. Often you will find that the more complicated a query is, the more Postgres estimates incorrectly. In such cases, unroll the query (e.g. by removing sub-selects and making them temporary tables and then analyzing them).

## Understand Table Joins

The performance of table joins is key to overall query processing performance. There are several join types that are commonly executed by the optimizer:

- hash joins
- merge joins
- nested loop joins

In the sections that follow, we describe these three join types and provide tips on tuning Aster Database to choose join types that work well for your queries.

### Hash joins

#### Hash joins

A hash join is typically faster than a merge join, given enough memory. This is the join method most typically observed. The Postgres optimizer picks the smaller table for constructing the hash table by looking at the unique distribution of values in the smaller table. If the table hasn’t been analyzed, then the Postgres optimizer could mistakenly assume that the hash table will fit in memory whereas in fact, the table could be requiring more memory than available RAM. In such cases, there will be Swap activity happening. That is the surest sign that the estimates are wrong.

#### EXPLAIN plan showing a hash join

The explain plan shown below is from the vworker:

```
tpcc=> EXPLAIN SELECT * FROM warehouse JOIN district ON (d_w_id= w_id) ;  
QUERY PLAN
```

```
Hash Join (cost=1.02..2.26 rows=10 width=176)
Hash Cond: (district.d_w_id= warehouse.w_id)
-> SeqScan on district (cost=0.00..1.10 rows=10 width=91)
-> Hash (cost=1.01..1.01 rows=1 width=85)
-> SeqScan on warehouse (cost=0.00..1.01 rows=1 width=85)
```

### Notes on hash joins

- Hash join is the join method most typically observed.
- A hash join tends to be the more efficient join type to use when the *joining column of the smaller of the two tables* will fit into a hash table in the allocated (`work_mem`) amount of memory. A hash join is efficient because it takes place in a single pass, whereas sorting in a merge join may require more than one pass.
- The optimizer picks the smaller table for constructing the hash table by looking at the unique distribution of values in the smaller table.
- It's important that tables' statistics are up to date. Make sure you run ANALYZE after any significant change to a table's contents. Otherwise, the optimizer may mistakenly guess that a column's contents will fit into an in-memory hash table when, in fact, they will not. In such cases the hash table spills to virtual memory, and performance is likely to be poor. If this happens, you will see disk swap activity on the worker node. That is a sign the plan's estimates might be wrong.

## Merge Joins

### Merge joins

A merge join is the most scalable and widely usable method of joining. That also typically makes it the slowest.

The merge join is implemented by sorting both the tables on the columns being joined and then streaming the top few rows from each table to do the join. This makes for a very low memory utilization during the join operation. But the sorting step requires memory, and again misestimation causes problems. If Postgres thinks a table sorting will fit in memory, then it will use a quicksort algorithm. Else, it will use an external disk-based sorting algorithm (using at most `work_mem` amount of memory at any point during the sort).

### EXPLAIN plan that shows a merge join

```
tpcc=> EXPLAIN SELECT * FROM order_line JOIN item ON (i_id = ol_i_id) ;
QUERY PLAN
-----
Merge Join (cost=58862.12..67288.43 rows=301231 width=142)
Merge Cond: (item.i_id= order_line.ol_i_id)
-> Index Scan using pk_itemon item (cost=0.00..3659.26 rows=100000 width=72)
-> Materialize (cost=58861.10..62626.49 rows=301231 width=70)
-> Sort (cost=58861.10..59614.18 rows=301231 width=70)
Sort Key: order_line.ol_i_id
-> SeqScan on order_line(cost=0.00..6731.31 rows=301231 width=70)
```

Can only be used for equality join conditions. This is only practical way to do join on large tables that do not fit in memory, and this works great for all cases where the tables are large and the optimizer thinks they cannot fit in `work_mem` amount of memory.

## Merge join: Details

Merge join is the most scalable and widely usable method of joining.

- That also typically makes it the slowest if any of the other two approaches are possible.
- The merge join is implemented by sorting both the tables on the columns being joined and then streaming the top few rows from each table to do the join.

This makes for a very low memory utilization during the join operation.

However, the sorting step requires memory, and again bad estimates cause problems.

If the optimizer thinks a table sorting will fit in memory, then it will use a quicksort algorithm.

Else, it will use an external disk-based sorting algorithm (using at most `work_mem` amount of memory at any point during the sort).

## Nested Loop Joins

Nested loop joins are the fastest but can't always be used. Since the default statistics for a table that has not been analyzed tend to produce table size estimates that are too small, and since nested loop joins are the joins of choice for smaller tables, the Postgres planner, by default, tends to pick nested loops. For this reason, all Aster Database deployments ship with the `enable_nestedloop` parameter set to off. Turn it on with care, and only after you have done an EXPLAIN ANALYZE directly on the worker Postgres instances! (Aster Database does not support EXPLAIN ANALYZE, but Postgres does, so you can run it by connecting directly to a Postgres instance.)

Nested loop joins are very useful for performing a star-schema join between a large fact table and a very small dimension table (either the dimension table is itself small or the dimension table is being filtered down to a small set of rows). In such cases, consider creating an index on the large fact table on the column being joined with the very small dimension table. Turn on nested loops, and then check to see that the Postgres optimizer is scanning the very small dimension table first and then using its joining column values as probes into the large fact table using the index.

### Example EXPLAIN plan for a nested loop join

```
tpcc=> EXPLAIN SELECT * FROM order_line JOIN item ON (i_id= ol_i_id) WHERE ol_o_id= 1;
QUERY PLAN
-----
Nested Loop (cost=0.00..7849.57 rows=102 width=142)
 -> Index Scan using pk_o_lineon order_line(cost=0.00..7092.11 rows=102 width=70)
Index Cond: (ol_o_id= 1)
 -> Index Scan using pk_itemon item (cost=0.00..7.41 rows=1 width=72)
Index Cond: (item.i_id= order_line.ol_i_id)
```

### Notes on nested loop joins

Nested loop joins:

- are least efficient of join in theory, fast to produce first record;
- work great when the optimizer thinks both the tables are really small;

- work great when one table is much, much smaller than the other AND the larger table has an index on the joining column.

## Tuning join operations

### Making joins run fast

For all join operations, the most important cluster property to consider is the amount of physical memory in the machine, number of vworkers in that physical worker.

Another important measure is the complexity of the query (number of GROUP BY/DISTINCT aggregations, number of UNION operations, number of JOINs). The more the number of such operations, the lesser the `work_mem` value should be.

### The `work_mem` parameter

When tuning Aster Database's preferences for join types, the most important Postgres parameter to worry about is `work_mem`. Postgres picks among the three operation types above based on the `work_mem` setting in relation to the amount of memory it thinks is needed for each of the operation types.

### Advanced tip: One-column cardinality tracking

Postgres only keeps track of value distributions one column at a time. As a result, it often does a poor job estimating the joint cardinality distribution of two columns. The way this affects you is when you join do a snowflake join on two columns between a fact table and its extension or two fact tables (e.g. SESSIONID and USERID). For such joins, Postgres will most likely assume that each of the two columns you are joining on will filter some columns and together they will filter a lot, but in practice those two columns could be very highly correlated to each other (e.g. no two USERID values will have the same SESSIONID value, and so each (SESSIONID,USERID) is unique).

## Summary of join tuning tips

When you tune your join operations, keep in mind:

- Remember: It's what Postgres "thinks" that matters. So ANALYZE, ANALYZE, ANALYZE!
- The most important parameter to worry about is `work_mem`.
- The optimizer picks among the three operation types we have discussed based on the `work_mem` setting in relation to the amount of memory it thinks is needed for each of the operation types.
- For all join operations, the most important cluster parameter to think about is amount of physical memory in the machine (and the number of vworkers in that physical worker).

# Use Linux Utilities to Isolate Query Performance Issues

While a query is executing, we can use Linux tools including *top* and *vmstat* to view how the query is using system resources. Before you do this, make sure you have tracked down the issues at a cluster-wide level as explained in “[Use Ganglia to find the overworked worker](#)” on page 211. After you have tracked down the over-busy workers, you can use the tools explained in the sections below:

- *vmstat*: Run *vmstat* on the worker node to report virtual memory statistics real-time. See “[System statistics: vmstat](#)” on page 236.
- *top*: Run *top* on the worker node to view running Linux processes. See “[CPU Monitoring with TOP](#)” on page 238.
- Use *du* and *nc\_relationstats* to find data skew and processing skew. See “[Find data skew](#)” on page 242 and see “[Find processing skew](#)” on page 242.

## System statistics: **vmstat**

The UNIX *vmstat* command is useful for finding out how busy a worker node is. Specifically, *vmstat* prints statistics showing memory usage, disk paging, I/O wait times, and cpu activity.

### **vmstat syntax**

```
vmstat [-a] [-n] [delay [ count ]]  
vmstat [-f] [-s] [-m]  
vmstat [-S unit]  
vmstat [-d]  
vmstat [-p disk partition]  
vmstat [-V]
```

### **vmstat summary output**

The *-s* switch displays a summary table of various event counters and memory statistics.

```
vmstat -s  
16436312 total memory  
3521688 used memory  
979208 active memory  
1499132 inactive memory  
12914624 free memory  
814396 buffer memory  
1211316 swap cache  
32872552 total swap  
0 used swap  
32872552 free swap  
177877 non-nice user cputicks  
304 nice user cputicks  
148176 system cputicks  
55251857 idle cputicks  
64128 IO-wait cputicks  
0 IRQ cputicks  
58252 softirqcputicks  
1662615 pages paged in
```

```
24676376 pages paged out
0 pages swapped in
0 pages swapped out
20026534 interrupts
46977906 CPU context switches
1255653923 boot time
227604 forks
```

## vmstat example

Example:

```
vmstat -a -n 5 4
-a = Active/Inactive Memory switch
-n = Display the header 1 time
5 = Delay between system output
4 = Number of iterations
procs-----memory-----swap-----io-----system-----cpu-----
r b swpdfree inactactive siso bi boin csus syid wa
0 0 0 12914316 1498940 979484 0 0 3 44 36 7 0 0 0 99 0
0 0 0 12914932 1498968 979552 0 0 0 50 410 886 0 0 0 100 0
0 0 0 12914488 1498980 979936 0 0 0 78 373 965 0 0 0 100 0
1 0 0 12913988 1498928 980900 0 0 0 181 167 432 0 0 0 100 0
```

## vmstat: VM mode reporting fields

Field Descriptions for VM Mode

### Procs

- r: The number of processes waiting for run time.
- b: The number of processes in uninterruptible sleep.

### Memory

- swpd: the amount of virtual memory used.
- free: the amount of idle memory.
- buff: the amount of memory used as buffers.
- cache: the amount of memory used as cache.
- inact: the amount of inactive memory. (-a option)
- active: the amount of active memory. (-a option)

### Swap

- si: Amount of memory swapped in from disk (/s).
- so: Amount of memory swapped to disk (/s).

### IO

- bi: Blocks received from a block device (blocks/s).
- bo: Blocks sent to a block device (blocks/s).

### System

- in: The number of interrupts per second, including the clock.
- cs: The number of context switches per second.

CPU These are percentages of total CPU time.

- us: Time spent running non-kernel code. (user time, including nice time)
- sy: Time spent running kernel code. (system time)
- id: Time spent idle. Prior to Linux 2.5.41, this includes IO-wait time.
- wa: Time spent waiting for IO. Prior to Linux 2.5.41, shown as zero.

### **vmstat: What are we looking for?**

Use vmstats in conjunction with Ganglia and the 'top' command to get an idea of system performance.

- Specifically look for swapping (si= swap in, so = swap out)
- Look for CPU wait time. (CPU wa)

## **CPU Monitoring with TOP**

The 'top' program provides a dynamic real-time view of a running system.

- It can display system summary information as well as a list of tasks currently being managed by the Linux kernel.
- The types of system summary information shown and the types, order and size of information displayed for tasks are all user configurable and that configuration can be made persistent across restarts.
- Run 'top' from the Linux command line. Press 'h' at any time to toggle online help.

### **TOP: Header report fields**

The 'top' command displays a variety of information about the processor state. The display is updated every five seconds by default, but you can change that with the 'd' command-line option or the interactive command, 's'.

When you run top, it displays the following information in the uppermost portion of your console:

- Up (uptime) – This line displays the time the system has been up, and the three load averages for the system. The load averages are the average number of processes ready to run during the last 1, 5, and 15 minutes. This line is just like the output of the 'uptime' command. The uptime display may be toggled by the interactive command, 'T'.
- Tasks / processes – Shows the total number of processes running at the time of the last update. This is also broken down into the number of tasks which are running, sleeping, stopped, or undead. The processes and states display may be toggled by the interactive command, 't'.
- Cpu(s) – Shows the percentage of CPU time in user mode, system mode, 'niced' tasks, iowait and idle. ('Niced' tasks are only those whose nice value is positive.) Time spent in niced tasks will also be counted in system and user time, so the total will be more than 100%. The processes and states display may be toggled by the interactive command, 't'.

- Mem – Statistics on memory usage, including total available memory, free memory, used memory, shared memory, and memory used for buffers. The display of memory information may be toggled by the interactive command, 'm'.
- Swap – Statistics on swap space, including total swap space, available swap space, and used swap space. The contents of this field and the Mem field are just like the output of the 'free' command.

In the remainder of your console, 'top' displays the following report fields:

- PID: The process ID of each task.
- PPID: The parent process ID of each task.
- UID: The user ID of the task's owner.
- USER: The user name of the task's owner.
- PRI: The priority of the task.
- NI: The nice value of the task. Negative nice values are higher priority.
- VIRT: Virtual Image (kb) The total amount of virtual memory used by the task. VIRT = SWAP + RES.
- SIZE: The size of the task's code plus data plus stack space, in kilobytes, is shown here.
- TSIZE: The code size of the task. This gives strange values for kernel processes and is broken for ELF processes.
- DSIZE: Data + Stack size. This is broken for ELF processes.
- TRS: Text resident size.
- SWAP: Size of the swapped out part of the task.
- DSize of pages marked dirty.
- LC: Last used processor.
- RSS: The total amount of physical memory used by the task, in kilobytes, is shown here.
- SHARE: The amount of shared memory used by the task is shown in this column.
- STAT: The state of the task is shown here. It's one of:
  - S for sleeping
  - D for uninterruptible sleep
  - R for running
  - Z for zombies
  - T for stopped or traced.
  - These states are modified by trailing < (less-than sign) for a process with negative nice value, N for a process with positive nice value, W for a swapped out process.
- WCHAN: This shows the address or the name of the kernel function the task currently is sleeping in.
- TIME: Total CPU time the task has used since it started.
- %CPU: The task's share of the CPU time since the last screen update, expressed as a percentage of total CPU time per processor.
- %MEM: The task's share of the physical memory.

- **COMMAND:** The task's command name, which will be truncated if it is too long to be displayed on one line. Tasks in memory will have a full command line, but swapped-out tasks will only have the name of the program in parentheses (for example, “(getty)”).

### **TOP: Worker Node Example**

Running 'top' with no parameters creates a display that refreshes (by default) every three seconds.

```
top -07:01:51 up 5 days, 12:17, 1 user, load average: 0.38, 0.29, 0.16
Tasks: 178 total, 2 running, 176 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.2% us, 0.2% sy, 0.0% ni, 99.3% id, 0.0% wa, 0.0% hi, 0.3% si
Mem: 6046832k total, 4075068k used, 1971764k free, 156340k buffers
Swap: 11807348k total, 0k used, 11807348k free, 3510632k cached
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
22485 beehive 20 0 654m 13m 11m S 74 0.2 0:04.58 postgres
22486 beehive 20 0 654m 13m 11m R 54 0.2 0:03.72 postgres
22495 beehive 20 0 78360 11m 1656 S 23 0.2 0:00.62 IWTServerExec
5968 beehive 20 0 31900 1404 572 S 4 0.0 0:20.92 postgres
22497 beehive 20 0 75288 8772 1188 S 4 0.1 0:00.02 IWTServerExec
22509 beehive 20 0 10668 1296 892 R 4 0.0 0:00.02 top
5293 beehive 20 0 268m 24m 1628 S 2 0.4 142:59.20 python
1 root 20 0 2612 580 492 S 0 0.0 0:04.98 init
```

### **TOP: Queen Example**

```
top -07:06:13 up 9 days, 18:58, 2 users, load average: 0.01, 0.02, 0.00
Tasks: 237 total, 1 running, 235 sleeping, 0 stopped, 1 zombie
Cpu(s): 0.7% us, 0.2% sy, 0.0% ni, 99.0% id, 0.0% wa, 0.1% hi, 0.1% si
Mem: 6046832k total, 6006724k used, 40108k free, 786540k buffers
Swap: 12096936k total, 228k used, 12096708k free, 3591048k cached
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
6848 beehive 20 0 85584 23m 6608 S 1 0.4 0:00.54 queenExec
14312 beehive 20 0 33588 7576 2236 S 1 0.1 12:04.42 netmonExec
6861 beehive 20 0 10672 1436 968 R 1 0.0 0:00.02 top
13364 beehive 20 0 36472 14m 2024 S 1 0.2 7:49.17 LogServer
15623 beehive 20 0 3796 672 568 S 1 0.0 0:06.98 sadc
4898 root 20 0 2460 356 288 S 0 0.0 3:47.30 supervise
14914 beehive 20 0 78016 6616 1336 S 0 0.1 0:11.00 python
1 root 20 0 2612 576 492 S 0 0.0 0:07.16 init
```

Note some of the different processes on the Queen that do not execute on the Worker node.

### **TOP: What are we looking for at the queen?**

On the queen, look for the 'QueenExec' process run by user 'beehive'. It should be among the most resource-consuming processes, consuming most of the CPU:

| PID   | USER    | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+   | COMMAND   |
|-------|---------|----|----|------|-----|-----|---|------|------|---------|-----------|
| 18582 | beehive | 20 | 0  | 0    | 0   | 0   | Z | 10   | 8.0  | 0:00.00 | QueenExec |
| 18821 | root    | 20 | 0  | 0    | 0   | 0   | Z | 0    | 0.0  | 0:00.00 | dnscache  |

### **TOP: What are we looking for at the worker?**

On the worker node, look for the Postgres processes. There should be one for each vworker on that worker node. When you view the 'top' output during query execution, if you see fewer running Postgres processes than you have vworkers on the node, this may indicate processing

skew. Seeing fewer Postgres processes than vworkers indicates that some vworkers have completed processing a given query before others, which may be a sign of skew.

Taken together, the Postgres processes should consume almost all available CPU. A properly configured Aster Database should be CPU bound when processing its typical large queries, rather than I/O bound or network bound.

Look for the IceServer process on current versions of Aster Database, or the IWTServerExec process on pre-4.5 versions of Aster Database. This is the process that shuffles data between vworkers.

Look for high amounts of swap activity. The 'Swap' field appears near the top of your console:

```
Mem: 6046832k total, 6006724k used, 40108k free, 786540k buffers
Swap: 12096936k total, 228k used, 12096708k free, 3591048k cached
```

## TOP: CPU stats

While running 'TOP', pressing the '1' key will toggle the distinct CPU usage by processor.

```
top -08:10:29 up 5 days, 13:26, 1 user, load average: 0.00, 0.02, 0.00
Tasks: 167 total, 1 running, 166 sleeping, 0 stopped, 0 zombie
Cpu0: 0.3% us, 1.0% sy, 0.0% ni, 95.7% id, 0.0% wa, 0.0% hi, 3.0% si
Cpu1: 0.0% us, 0.0% sy, 0.0% ni, 100.0% id, 0.0% wa, 0.0% hi, 0.0% si
Cpu2: 0.0% us, 0.0% sy, 0.0% ni, 100.0% id, 0.0% wa, 0.0% hi, 0.0% si
Cpu3: 0.0% us, 0.0% sy, 0.0% ni, 100.0% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 6046832k total, 4066752k used, 1980080k free, 156516k buffers
Swap: 11807348k total, 0k used, 11807348k free, 3518552k cached
```

Look at the wait (wa) percentage. Long wait times can indicate swapping. One CPU running at a higher use percent ('us') than the rest can indicate processing skew.

## TOP output sorted by virtual memory usage

Sorting by "Virtual Memory" usage lets us see how much CPU the Postgres execs are using. These should not total more than the available CPU.

```
top -09:10:06 up 5 days, 14:25, 1 user, load average: 0.75, 0.18, 0.06
Tasks: 178 total, 3 running, 175 sleeping, 0 stopped, 0 zombie
Cpu(s): 37.5% us, 7.4% sy, 0.0% ni, 55.0% id, 0.1% wa, 0.0% hi, 0.0% si
Mem: 6046832k total, 4140276k used, 1906556k free, 156596k buffers
Swap: 11807348k total, 0k used, 11807348k free, 3545360k cached
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
27854 beehive 20 0 668m 32m 17m R 85 0.6 0:15.28 postgres
27853 beehive 20 0 668m 32m 17m R 83 0.6 0:15.52 postgres
27855 beehive 20 0 654m 8368 6220 S 1 0.1 0:00.06 postgres
27881 beehive 20 0 652m 4988 3304 S 0 0.1 0:00.00 postgres
27882 beehive 20 0 652m 4936 3248 S 0 0.1 0:00.00 postgres
27883 beehive 20 0 652m 4952 3264 S 0 0.1 0:00.00 postgres
5962 beehive 20 0 650m 30m 30mS 0 0.5 0:03.40 postgres
5963 beehive 20 0 650m 29m 29mS 0 0.5 0:03.52 postgres
5964 beehive 20 0 650m 9.9m 9516 S 0 0.2 0:11.90 postgres
```

## Find data skew

### Finding skew with UNIX du

The Linux du command is used to look for data skew on a worker node. Its syntax is:

```
du [OPTIONS] [file systems]
```

Data on the Aster Database workers is stored in directories with names like /primary/w5z (vworker number 5), /primary/w12z (vworker number 12), and so on. Collectively, we refer to these directories as the "w\*z" directories. We can examine these directories to check for data skew. For example, to show space usage for all the virtual workers on a node, you type:

```
du -sh /primary/w*
```

To sort your results, pipe them through 'sort':

```
du -sh /primary/w* | sort
```

Your results might look like this:

```
119M    /primary/w14z
119M    /primary/w30z
120M    /primary/w10z
120M    /primary/w26z
121M    /primary/w34z
124M    /primary/w18z
129M    /primary/w42z
130M    /primary/w22z
130M    /primary/w2z
131M    /primary/w46z
134M    /primary/w6z
196M    /primary/w38z
```

Note that vworker number 38 has 60 MB more data than the lowest vworker!

### Finding skew with the nc\_relationstats function

The nc\_relationstats function is very useful in detecting skew. For more information on the usage of this function, see the *Teradata Aster Big Analytics Appliance 3H Database User Guide*.

## Find processing skew

If we don't find any data skew and we have a very long running query, we have to look for processing skew. To figure out processing skew:

1 Start your long-running SQL statement.

2 On the queen, run:

```
ps -ef | grep "postgres: <Aster Database user name>"
```

where "Aster Database user name" is the name of the logged in SQL session, such as the default user, "beehive."

3 Find the queendb process that corresponds to the start of your transaction. Note the timestamp to see when the process started. For this example, let's say the username is "beehive," and the timestamp is "16:09".

- 4 Check all the worker nodes to see how many vworkers are still processing as part of the cluster-wide execution of the user statement.

### Processing skew: Checking the workers

Execute this statement at the Queen:

```
if [ -e /root/cluster-management/hosts ]; then
    workers='cat /root/cluster-management/hosts | grep -v loader | awk '/^
*[^#]/ {if ($1 != "sysman") {print $NF}}``;
else
    workers='cat /home/beehive/cluster-management/hosts | grep -v loader |
awk '/^ *[^#]/ {if ($1 != "sysman") {print $NF}}``;
fi;

for i in $workers; do echo WORKER $i; ssh$i"ps -ef" |grep"postgres:
beehive" |grep"16:09" |grep -v idle; done;
```

Each vworker that is actively executing will show up as one “postgres” process. The vworkers that are done would not be shown. Note that if there are multiple user transactions by this user that started at the same minute, then this method will show activity for all those transactions that started in the same minute.

### Processing skew: Possible causes

Here are some things to look for when investigating processing skew:

- Tables with a distribution key on a column that when filtered reduces the cardinality. For example, a table with a distribution key on a DATE column. In this example, filtering on the date column might reduce the number of vworkers involved in the query.
- Skew caused by data redistribution. For example, suppose many of our orders are from anonymous customers, and let's say we have chosen to give all anonymous customers a customer\_id value of “-1”. In this case, all such line items end up on the same vworker!
- Skew caused by the data redistribution.

### Processing skew: Finding it

Look at the EXPLAIN plan for the query. If the plan says a particular table's rows will be redistributed on a particular column, then we should find out whether our query is reducing the cardinality of the column on which we are redistributing. For example, in the query below, we are looking for the skew scenario where one or a couple of the top customerid values have orders of magnitude more rows than the next most popular customerid values:

```
SELECT customerid, count(*)
  FROM lineitem
 GROUP BY customerid
 ORDER BY 2 DESC
 LIMIT 20;
```

## Additional Resources

Refer to EXPLAIN in Aster Database documentation for understanding Aster Database-level Explain output.

- The site, [http://wiki.postgresql.org/wiki/Using\\_EXPLAIN](http://wiki.postgresql.org/wiki/Using_EXPLAIN), provides links to some detailed explanations of explain plans.
- <http://www.postgresql.org/docs/8.4/static/using-explain.html> provides a clear introduction to EXPLAIN.
- [http://wiki.postgresql.org/images/4/45/Explaining\\_EXPLAIN.pdf](http://wiki.postgresql.org/images/4/45/Explaining_EXPLAIN.pdf) does a very good job at how to interpret and understand the Postgres EXPLAIN output.
- For tuning tips, apart from the regular PostgreSQL links on <http://www.postgresql.org/docs/current/interactive/performance-tips.html>, one of the most useful collections of links for PostgreSQL tuning (and therefore Aster tuning) is [http://wiki.postgresql.org/wiki/Performance\\_Optimization](http://wiki.postgresql.org/wiki/Performance_Optimization).
- [http://www.postgresql.org/files/documentation/books/aw\\_pgsql/hw\\_performance/0.html](http://www.postgresql.org/files/documentation/books/aw_pgsql/hw_performance/0.html) gives high-level overview of understanding generic database system performance. (It is dated 2003 but much basic material remains relevant!)
- <http://neilconway.org/talks/executor.pdf> - Starting on slide 18, this presentation helps you understand various join algorithms in PostgreSQL, and, starting on slide 33, aggregations. Joins and aggregations are the most common types of SQL operations you will need to tune.

## Checklist for Query Tuning

Follow this checklist to address query performance problems:

- 1 Check for bottlenecks. See [Use Ganglia to find the overworked worker \(page 211\)](#).
- 2 Run ANALYZE. See [Run ANALYZE regularly to ensure the most optimal query plans \(page 212\)](#)
- 3 Run VACUUM. See [Run VACUUM to ensure queries achieve best performance \(page 212\)](#)
- 4 Make sure you're using transactions. [When updating or adding data to Aster Database, wrap statements in a transaction \(page 213\)](#).
- 5 Make sure you've written your queries in a way that allows Aster Database to parallelize the work as much as possible. See [Optimize Queries for Parallel Execution \(page 214\)](#).
- 6 Check the EXPLAIN plan to make sure Aster Database has chosen a good plan for running the query. See [Read the EXPLAIN Plan \(page 223\)](#).
- 7 If you suspect Aster Database has chosen the wrong join technique, see the section [“Understand Table Joins” on page 232](#).
- 8 If you suspect too much data is being shuffled among workers, see [“A Step-by-Step Approach to Query Tuning” on page 227](#).

- 9 If none of the techniques listed above solve the problem, try to isolate the problem using the Linux tools described in “[Use Linux Utilities to Isolate Query Performance Issues](#)” on [page 236](#), and contact Teradata Global Technical Support (GTS) for help.

## CHAPTER 4 User Management

This section introduces topics related to the management of database users.

This section shows:

- [Manage Users and Privileges](#)
- [User Authentication](#)
- [Schema Search Path](#)
- [System Internal Users](#)

## Manage Users and Privileges

Users in Aster Database are authenticated before they can access the various database components. A user has access to the various database components based on the privileges that they have been granted. The different authentication modes and the privileges that can be granted and revoked are explained in detail below.

### Access Control

Limits on users' ability to read from and write to databases are governed as follows:

- Aster Database security on database objects is managed through GRANT and REVOKE statements.
- GRANT statements grant privileges on database objects to one or more roles or individual users.
- Object level security authorizations are stored locally in systems tables on the coordinator.

Users' rights to use the AMC are also managed with GRANT and REVOKE.

### Default Roles and Users

As installed, Aster Database contains two default administrative users and roles, and a default database, beehive.

**Default users:**

- `db_superuser`: The user `db_superuser` (default password: `db_superuser`) has the powerful `db_admin` role and can access all database objects in every possible way without any restrictions. This account it should be used only when needed and with care.
- `beehive`: The user `beehive` (default password: `beehive`) owns the default database, also called `beehive`. By default, the `beehive` user has no administrative rights.

Important! Immediately after you install Aster Database, you should change the default password of `db_superuser` to one that is more secure.

**Default roles:**

- `db_admin` is Aster Database's superuser role. A member of this role will automatically be an Aster Database Administrator. *Only superusers can create and alter databases and roles!* The `db_admin` role does *not* provide AMC access. Note that the `db_admin` role presents a special case when configuring workload management. See [Taking Roles into Account \(page 288\)](#) for more information.
- `catalog_admin` is Aster Database's standard administrative role. This role has a minimal set of administrative rights. The `catalog_admin` role has the privilege to view all system tables. (However, the `catalog_admin` role does not have unrestricted access to everything in the database. For example, the catalog administrator cannot arbitrarily modify user tables unless explicitly granted permission to do so.) The `catalog_admin` role does *not* provide AMC access.
- `amc_admin` and similar roles determine what actions the user can undertake in the AMC.

The default administrative roles `db_admin` and `catalog_admin`, as well as the default administrative user `db_superuser` cannot be dropped or altered in any way.

If you're logged in a superuser, you can retrieve information about the current set of users, roles, and databases in your cluster from the Aster Database system tables. Refer to the *Teradata Aster Big Analytics Appliance 3H SQL and Function Guide* for more on system tables.

## User Name Rules

Database user names must not contain:

- Whitespace (including space, tab, and newline characters)
- Colon :, backslash \, or single-quote ' (apostrophe) characters
- Control characters (including ASCII 0-31 and 127).
- Multibyte characters.

## Password Rules

Database user passwords must follow certain rules, beginning in Aster Database 5.0.2. These password rules apply to new installations of and upgrades to Aster Database 5.10.00.01. If any database users were created with a disallowed password before upgrading, their ability to login

will not be affected in this release. However, any new passwords (for new or existing users) must follow the new rules.



**Notice!** These password rules are not enforced when a user password is created or changed. To avoid any problems that could occur upon future upgrades, make sure that users only have passwords that are in compliance.

These password rules apply regardless of the authentication method. If you are automatically generating passwords, ensure that only passwords that follow these rules can be generated. If you use any tools which automatically generate passwords, you may need to modify them to choose only Aster-supported passwords.

- 1 Password length must be at least 1 character and most 128 characters.
- 2 Password must consist of ASCII characters (foreign language or multibyte characters are not supported).
- 3 Password must not contain:
  - Whitespace (including space, tab, and newline characters)
  - Colon :, backslash \, or single-quote ' (apostrophe) characters
  - Right parenthesis (, left parenthesis ), double quote " and back-quotes ` .
  - Control characters (including ASCII 0-31 and 127).

Passwords are case sensitive.

## Add Users

New users in Aster Database are defined through the CREATE USER command:

```
CREATE USER name [ [ WITH ] option [ . . . ] ] PASSWORD 'password'
```

See CREATE USER in the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference* for the list of options.

The following example demonstrates how to add a new user to Aster Database with the name “ryan” in the group “marketing” with specified password “ryan123.”

```
CREATE USER ryan IN GROUP marketing PASSWORD 'ryan123'
```

When you create new users, databases, schemas, and tables, you will typically need to grant some combination of the following privileges to give users access to database objects and contents:

- GRANT CONNECT allows the user to connect to a database.
- GRANT CREATE on a database gives the user/role the right to create new schemas in the database. Granting CREATE on a database does not confer the right to create tables. To do that, you must do the following:
  - GRANT CREATE on a schema gives the user or role the right to create new tables and objects in the schema.
  - GRANT USAGE on a schema gives the user or role the right to access objects in the schema.

- Use GRANT SELECT, INSERT, UPDATE, DELETE, or ALL on a table to give the user or role appropriate access to a table.

## Database Users and Roles

Aster Database manages database access permissions using the concepts of users and roles. A role can be thought of as a group of database users. Users own database objects and can assign privileges on those objects to other users or roles to control the level of access to objects (e.g. table access). Furthermore, it is possible to grant membership in a role to another role or user, thus granting to the role/user the use of privileges assigned to this role. Once granted membership, a user or role is considered to be a “member” of this role group.

Note that database users are separate from UNIX system users. In practice it might be convenient – but not required – to maintain a correspondence between the two.

Database users are global across an Aster Database installation (and not per individual database). To create user “theadon” with password ‘5t4g0l33’, use the SQL command CREATE USER:

```
CREATE USER theadon PASSWORD '5t4g0l33';
```

The user name must follow the rules for SQL identifiers: either double-quoted or without special characters. The password must follow the [Password Rules](#).

To remove an existing user, use the DROP USER command, as in:

```
DROP USER theadon;
```

If you are unable to drop a user, you may need to revoke its privileges before dropping the user.

In order to bootstrap the database system, a freshly initialized system always contains one predefined user, “beehive.” To create more users, you first have to connect as this initial user. Every connection to the database server is made in the name of some user, and this user determines the initial access privileges for commands issued on that connection. The username to use for a particular database connection is indicated by the client that is initiating the connection request in an application-specific fashion.

## Example of Creating a User and Connecting with ACT

Create a new user from within ACT:

```
beehive=> CREATE USER theadon PASSWORD '5t4g0l33';
beehive=>\q
```

The user can now connect to Aster Database from ACT via the following ACT command line options:

```
$ act -h 10.50.43.100 -U theadon -w 5t4g0l33
```

In this example, the ACT client accesses:

- the IP address 10.50.43.100 (specified via the -h option)
- for the user theadon (specified via the -U option)
- with the password 5t4g0l33 (specified via the -w option).

## User Attributes

A database user may have a number of attributes that define its privileges and interact with the client authentication system. These are described below:

### Database and User Creation Privileges

A user must be explicitly given permission to create databases. To grant such rights, type

```
GRANT rolename TO username;
```

where `rolename` is the name of a role with database and user creation privileges. If you like, you can use Aster Database's powerful `db_admin` role for this as shown below. Notice! the `db_admin` role is a powerful administrator role!

```
GRANT db_admin TO username;
```

A user with `db_admin` privilege can alter and drop other users and roles, as well as grant or revoke membership in roles.

A user's attributes can be modified after creation with `ALTER USER`.

### AMC Usage Rights

A user's privileges in the Aster Database AMC depend on his or her role membership.

### Role Membership

You may find it convenient to group users to ease management of privileges. That way, privileges can be granted to, or revoked from, a group as a whole. In Aster Database this is done by creating a role that represents the group, and then granting membership in the group role to individual user roles.

New roles in Aster Database are defined through the `CREATE ROLE` command:

```
CREATE ROLE name [ [ WITH ] option [ . . . ] ] PASSWORD 'password'
```

See `CREATE ROLE` in the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference* for a list of options.

To set up a group role, first create the role:

```
CREATE ROLE rolename;
```

Then create one or more users:

```
CREATE USER username;
```

Once the group role exists, you can add and remove members using the `GRANT` and `REVOKE` commands:

```
GRANT rolename TO username, . . . ;
```

and

```
REVOKE rolename FROM username, . . . ;
```

There isn't any real distinction between group roles and non-group roles, so you can grant membership to other group roles, too. The only restriction is that you can't set up circular membership loops. Member roles automatically have the privileges of the group roles to which they belong.

As an example, suppose we have the following roles and privileges.

```
CREATE USER jstrummer PASSWORD 'm4gn1f1c3nc3';
CREATE ROLE admin ;
GRANT admin TO jstrummer;
```

Immediately after connecting as user *jstrummer*, a database session will have use of privileges granted directly to *jstrummer* plus any privileges granted to *admin*, because *jstrummer* "inherits" the privileges of *admin*.

You destroy a group role in the same manner with which you destroy any other role, using the DROP ROLE command:

```
DROP ROLE name;
```

If you are unable to drop a role, you may need to revoke privileges before dropping the role.

Any memberships in the group role are automatically revoked, but the individual members (users or roles) are not otherwise affected. So if DROP ROLE admin occurred, user *jstrummer* would no longer be a member of the group role "admin" since it was dropped. But *jstrummer* as a user would still exist and not be affected.

Note: Any objects owned by the group role must first be dropped or reassigned to other owners. Any permissions granted to the group role must be revoked.

## Privileges

When an object (such as a database, schema, or table) is created, it is assigned an owner. The owner is the user that created the object. For most kinds of objects, only the owner can do anything with the object initially. To allow other users and roles to use the object, privileges must be granted. There are many different privileges, including: SELECT, INSERT, UPDATE, DELETE, and CREATE. For more details, see the *Teradata Aster Big Analytics Appliance 3H SQL and Function Reference*.

To assign privileges, use the GRANT command. In Aster Database, privileges can be granted only at the database or table level. Note that *rolename* can be either a role group or user. Here's a subset of the GRANT syntax supported in Aster Database:

```
GRANT { { SELECT | INSERT | UPDATE | DELETE }
      [, ...] | ALL [ PRIVILEGES ] }
      ON [ TABLE ] tablename [, ...]
      TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ] [ CASCADE ]

GRANT { { CREATE | CONNECT } [, ...] | ALL [ PRIVILEGES ] }
      ON DATABASE dbname [, ...]
      TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { { CREATE | USAGE } [, ...] | ALL [ PRIVILEGES ] }
      ON SCHEMA schemaname [, ...]
      TO { username | GROUP rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

To revoke a privilege, use the REVOKE command.

Note: The special privileges of an object's *owner* – the right to modify or destroy the object – are always implicit and cannot be granted or revoked.

Note: Some types of objects can be assigned to a new owner with an object-appropriate ALTER command. A user/role can reassign ownership of an object only if she is both the current

owner of the object (or a member of the owning role) and a member of the new owning role. You cannot change the owner of a view.

## Set Up Read-Only Access for a User

If you want a user group in a deployment to be able to SELECT but not add or alter data or tables in the public schema, you can revoke the default rights to the public schema and grant limited rights as shown here. Let's assume the user's who we want to give read-only access are in a group called, "ANALYSTS":

By default, all users in Aster Database have read/write access to schema public, so first we must revoke that:

```
REVOKE ALL ON SCHEMA PUBLIC FROM PUBLIC;
```

Next, we grant back the rights we want the group to have. For example, let's assume we want them to be able to select from table1:

```
GRANT USAGE ON SCHEMA PUBLIC TO ANALYSTS;  
GRANT SELECT ON PUBLIC.TABLE1 TO ANALYSTS;
```

# User Authentication

## Authentication Methods

You configure your Aster Database installation to authenticate SQL and AMC users by one of the following authentication methods:

- Local password authentication: Aster Database validates the username and password against the user's record in its local repository on the queen (with backup on the cluster). Passwords stored here are masked. The Aster Database user accounts are *not* shared with the operating system user accounts and vice versa. This is the default. If you have activated LDAP or AD authentication and wish to switch back to password authentication, see ["Return to Password-Only Authentication" on page 263](#).
- Lightweight Directory Access Protocol (LDAP) authentication: Aster Database passes the username and password to the LDAP server for authentication. Your user accounts must be stored in an LDAP-compatible directory server such as Active Directory or OpenLDAP. See ["Set Up LDAP Authentication" on page 253](#).
- Active Directory authentication: This uses the LDAP mechanism discussed in the preceding bullet point.



**Notice!** The syntax for configuring LDAP has changed beginning in Aster Database 5.0. When upgrading to Aster Database 5.x or later from any version earlier than Aster Database 5.0, the installer migrates your LDAP settings to the new format automatically. The pg\_hba.conf formatting will be changed to entries that look like this example:

```
local all all ldap ldapserver="10.50.1.90" ldapport="389"  
ldapprefix="cn=" ldapsuffix=",CN=Users,DC=asterengqa,DC=com"
```

## Single Sign-On

Aster Database also supports single sign-on against Active Directory (AD). For this, Aster Database relies on a Quest/Vintel Authentication Services (QAS) tool running on the queen to authenticate a user against AD. For instructions, see “[Set Up Active Directory Authentication with Single Sign-On](#)” on page 257. (Note! To use implement this, you must buy and install the QAS tool from Quest Software.)

## Set Up LDAP Authentication

Users connecting to Aster Database can be authenticated using LDAP. (See “[User Authentication](#)” on page 252 for a list of the authentication types you can use.)

### Prerequisites for Setting Up LDAP Authentication

Four pieces of information are required before you can enable LDAP authentication in Aster Database:

- 1 Address of your LDAP server (for example, "ldap.example.com").
- 2 LDAP prefix string (for example, "AD-ENT\" for Windows Active Directory Server, "cn=" for other Windows LDAP servers, or "SAMAccountname=" for other operating systems).
- 3 LDAP suffix string (for example, ",ou=employees,ou=people;")

When a user tries to log into Aster Database after enabling LDAP, the following LDAP query is sent to the LDAP server:

```
<ldap_prefix><username><ldap_suffix>
```

where <username> is the person’s Aster Database user name, which must also match the username of his or her LDAP user account.

Using the example settings shown above, if a user named “jstrummer” tries to log in, the query sent to port 389 on ldap.example.com will be:

```
cn=jstrummer,ou=employees,ou=people;
```

If there are user accounts in both Aster Database and LDAP with the username “jstrummer”, and if the provided password matches the LDAP password of that user, the authentication succeeds.

### Enable LDAP Authentication

- 1 Make sure the Aster Database queen machine’s DNS settings allow it to communicate with the LDAP server. If necessary, edit /etc/hosts and /etc/resolve.conf.
- 2 You will configure LDAP for Aster Database using the script `ConfigureNCluster.py`. But running the script requires restarting the cluster to apply every change made, so you should first check your settings using the `ldapsearch` tool provided on the queen:
  - a Change the working directory to the directory containing `ldapsearch` on the queen. This should be similar to:

```
# cd /home/beehive/toolchain/x86_64-unknown-linux-gnu/5.10/bin/  
ldapsearch
```

- b If you will be using secure LDAP, make sure your `LDAPTLS_CACERT` environment variable is set correctly to the CA signed certificate:

```
# LDAPTLS_CACERT=root.pem
```

- c Run a LDAP search to verify that secure LDAP is working between the queen and the LDAP server outside of Aster Database, for example:

```
ldapsearch -b "DC=AsterData,DC=local" -h 10.1.1.11 -p 389
-D "CN=jstrummer,CN=Users,DC=Asterdata,DC=local" -w s3cr3t
"givenname=chakradhar"
```

or to test for secure LDAP:

```
ldapsearch -b "DC=AsterData,DC=local" -H ldaps://10.1.1.11 -s base
-D "CN=jstrummer,CN=Users,DC=Asterdata,DC=local" -w s3cr3t
"givenname=chakradhar" -x
```

In the above commands:

`-b` stands for searchbase as the starting point for the search instead of the default.

`-D` stands for the Distinguished Name binddn to bind to the LDAP directory.

In the example, we are authenticating as jstrummer with s3cr3t and then searching for user whose givenname is chakradhar.

But please note that the `ldapsearch` tool does not do an LDAP search to authenticate the user! The example only passes `"givenname=chakradhar"` to verify that the bind actually succeeded.

- 3 If the `ldapsearch` tool succeeds, you are ready to continue. Otherwise, adjust the parameters and run it again until you can connect verify LDAP connectivity.
- 4 Perform a soft shutdown on the cluster. Log in to the queen as root user and run the following command:

```
# ncli system softshutdown
```

- 5 Change the working directory to where the Aster Database configuration utility is located:
- 6 Run the configuration utility, to turn on LDAP authentication in Aster Database. After you perform this step, all users except beehive and db\_superuser will be authenticated against LDAP only:

```
# ./ConfigureNCluster.py --auth_type=LDAP --ldap-server="ldap-server"
--ldap-prefix="ldap-prefix" --ldap-suffix="ldap-suffix"
```

where:

- `ldap-server` is the address of the LDAP server which will be used to authenticate the user. For example:

```
--ldap-server="ldap.example.com"
```

- `ldap-prefix` is the label that appears before the username, as stored in LDAP. This parameter can be omitted if there is no prefix. For example, if your store Joe Strummer's account under the username `cn=jstrummer`, then your prefix is `cn=` (including the equals sign):

```
--ldap-prefix="cn="
```

- **ldap-suffix** is the label that appears after the username (often called an “ldap user suffix”), concluding with a semicolon on most systems. This parameter can be omitted if there is no suffix. For example:

```
--ldap-suffix=",ou=employees,ou=people;"
```

Combining the above examples, the command appears as follows:

```
# cd /home/beehive/bin/lib/configure
```

```
# ./ConfigureNCluster.py --auth_type=LDAP  
--ldap-server="ldap.example.com" --ldap-prefix="cn="  
--ldap-suffix=",ou=employees,ou=people;"
```

- 7 Soft restart the cluster by running the following as root user on the queen node:

```
# ncli system softrestart
```

After the restart, client authentication (for all users except beehive and db\_superuser) will be done through your LDAP server. User beehive and user db\_superuser always authenticate using their local Aster Database passwords.

- 8 Use the following setup instructions, depending on the type of LDAP authentication you wish to use:
  - [Set up Regular \(Unsecured\) LDAP](#)
  - [Set up Secure LDAP](#)

## Disable LDAP Authentication

To revert the LDAP configuration to the default (authentication through Aster Database only):

- 1 Perform a soft shutdown on the cluster. Log in to the queen as root and run the following command:

```
# ncli system softshutdown
```

- 2 Change the working directory to where the Aster Database configuration utility is located:

```
# cd /home/beehive/bin/lib/configure
```

- 3 Issue the command:

```
./ConfigureNCluster.py --auth_type=PASSWD
```

- 4 Restart the cluster:

```
# ncli system softstartup
```

## Set up Regular (Unsecured) LDAP

- 1 [Enable LDAP Authentication.](#)
- 2 [Verify the LDAP Setup.](#)

## Set up Secure LDAP

- 1 [Enable LDAP Authentication.](#)
- 2 On the queen, open the file /home/beehive/data/queenDb/pg\_hba.conf in a text editor and add "ldaptls=1" at the end of all lines that have an LDAP configuration entry. Such lines can be easily located by looking for lines that have the entry - `ldapsuffix=""`

- 3 Export the certificate from the certificate authority (CA) using the following steps on your Active Directory server:
  - a Open Control Panel > Administrative Tools > Certification Authority. The Certification Authority dialog box appears. You should see your CA listed in the left pane along with a + box to expand/collapse it.
  - b Expand it, and then click the Issued Certificates sub-item. In the right pane you should see a list of certificates.
  - c Right click on the CA you want to export and choose Open.
  - d Click on Details tab and click on the Copy the file button.
  - e You will see the Certificate Export Wizard. Click on Next.
  - f Choose the format to export. For use with OpenSSL (OpenLDAP), choose base-64 encoded X.509 format. The File to Export page appears.
  - g To save the certificate file to the default location, in the File Name text box, type a name for the certificate, for example sample-ca.cert
  - h Click Next. The Completing the Certificate Export Wizard page appears.
  - i Review the certificate information. Click Finish.
- 4 Copy the CA certificate file to the directory /home/beehive/data/queenDb on your Aster Database queen. Make sure the file owner is beehive and everybody has read permission on the file:

```
# chmod a+r sample-ca.cert
```
- 5 Edit /etc/profile to add the following lines at the end of the file:

```
LDAPTLS_CACERT=/home/beehive/data/queenDb/sample-ca.cert
export LDAPTLS_CACERT
```
- 6 Log out from the queen and then log back-in. This applies the setting you made.
- 7 [Verify the LDAP Setup](#).

## Verify the LDAP Setup

- 1 If you have changed the LDAP settings, soft restart the cluster by running the following as root user on the queen node:

```
# ncli system softrestart
```
- 2 Create a database user in the format firstname.middleinitial.lastname
  - a [Log In to ACT](#) as db\_superuser.
  - b Create the user. There should be a user with the identical username in your LDAP directory. If the username includes a dot, then it should be enclosed within double quotes as in this example.

```
#beehive=> CREATE USER "firstname.middleinitial.lastname" PASSWORD
'your-ldap-password';
CREATE USER
#beehive=> grant connect on database beehive to
"firstname.middleinitial.lastname";
GRANT
```

- 3 Try logging in as the user you created, to verify that LDAP authentication is working:

```
#act -U "firstname.middleinitial.lastname"  
Password for firstname.middleinitial.lastname:<ldap password>
```

- 4 If the login is successful, LDAP authentication is working.

## Guidelines for Using LDAP Authentication

To use LDAP authentication in Aster Database, each user must have two corresponding user accounts: one in LDAP and one in Aster Database. The usernames must match, and the user account in Aster Database must have connect privileges to the databases the user will use. If the user will use the AMC, you must grant an AMC-capable role to his or her Aster Database user account. Stated more explicitly, this means:

- 1 If an existing Aster Database user does not also have an account in LDAP, he or she will *not* be able to log in to Aster Database after LDAP is enabled! Create user accounts in LDAP for all Aster Database users.
- 2 If an LDAP user does not also have an account in Aster Database, he or she will *not* be able to log in to Aster Database! Even with LDAP authentication enabled, every Aster Database user must also have an account in Aster Database.
- 3 If the user's Aster Database user name does not match his or her global LDAP user name, he or she *cannot connect to Aster Database*. To fix this without creating new user accounts, create an alias in the global LDAP server that matches the Aster Database user name.

## Troubleshooting LDAP Authentication

If LDAP has been configured incorrectly, it could result in one or more vworkers not coming up as **Active**, but rather as **Failed**. To back out of this scenario, you must revert to your non-LDAP configuration using [Disable LDAP Authentication](#).

Then re-add the node(s) to the cluster one at a time, using the `-clean` option:

```
# ncli system addnode worker <worker_IP> -clean
```

## Set Up Active Directory Authentication with Single Sign-On

Users connecting to Aster Database can be authenticated using Active Directory (AD), and this authentication mechanism can provide single sign-on. By “single sign-on,” we mean that a logged-in Windows user can invoke Aster Database ACT, and he will be automatically logged into Aster Database with his Windows user account. (See [“User Authentication” on page 252](#) for a list of the authentication types you can use.)

### Requirements and Limitations

- Authentication across multiple forests is not supported.
- For single sign-on (“SSO”) to work, you must install Quest Authentication Services (“QAS,” formerly known as Vintela Authentication Services) version 4.0.3 on your queen. Please check with Teradata Global Technical Support (GTS) to find out which QAS version/OS version combinations are supported.

**To enable Active Directory integration**

- 1 Perform a soft shutdown on the cluster before you install QAS. Log in to the queen as root user and run the following command:

```
# ncli system softshutdown
```

- 2 Make sure the Aster Database queen machine's DNS settings allow it to communicate with the AD domain controller. To do this, you may need to edit the /etc/hosts and /etc/resolv.conf files.

In the queen's resolv.conf, add the DNS server's IP address as a name server entry (in our example setup, the DNS server and domain controller are the same machine and IP address). These settings enable the Linux machine to detect the domain controller.

You may also need to edit /etc/hosts to provide the full host name. The domain name is derived from the fully qualified host name by extracting the portion of the name that comes after the first dot ('.').

In the example text that follows, we assume the following settings:

- hostname of the queen node: cqueen
- domain name that Aster Database is part of: asterengqa.com
- queen IP address: 10.50.55.100
- name server IP address: 10.50.1.90

Edit the queen's /etc/hosts file so that it contains the queen's fully qualified domain name. For example, the relevant lines might look like this:

```
root@cqueen:~# head -2 /etc/hosts
127.0.0.1      localhost
10.50.55.100   cqueen.asterengqa.com    cqueen
```

The output of hostname command should show the hostname of the queen:

```
root@cqueen:~# hostname
cqueen
```

The output of dnsdomainname or domainname command should show the domain name correctly:

```
root@cqueen:~# dnsdomainname
asterengqa.com
```

Edit the resolv.conf to set the search list to the domain name and the IP address of the name server:

```
root@cqueen:~# cat /etc/resolv.conf
search asterengqa.com
nameserver 10.50.1.90
```

Use the ping or nslookup command to see if the above changes have taken effect. If the above changes have been applied, you should be able to ping the domain controller with its hostname:

```
root@cqueen:~# ping asterengqa.com
PING asterengqa.com (10.50.1.90) 56(84) bytes of data.
64 bytes from 10.50.1.90: icmp_seq=1 ttl=127 time=0.194 ms

--- asterengqa.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

```
rtt min/avg/max/mdev = 0.194/0.194/0.194/0.000 ms
```

- 3 Install QAS version 4.0.3 on the Aster Database queen. Follow the installation instructions provided by Quest for this. The main steps are:
  - a Copy the QAS installer bundle to your Aster Database queen and unpack it there. In this example, we unpack it in /tmp/vas
  - b Run the VAS preflight.sh script as Domain Administrator against the domain name to check if the prerequisites for VAS installation are met. For example:

```
# /tmp/vas/preflight.sh -u Administrator asterengqa.com
```
  - c Working as root on the queen, run the QAS installer by typing:

```
# /tmp/vas/install.sh
```

By default, QAS will be installed in /opt/quest. Teradata Aster strongly recommends that you accept the default QAS installation location.
  - d When prompted by the QAS installer, provide the QAS license as specified in the QAS documentation. If you don't have this now, you can add it later, but be aware that you cannot perform AD authentication until the license is installed.
  - e When prompted by the QAS installer, provide the name of the AD domain it will join. This is the Active Directory domain that holds your user accounts. Domain names typically carry a domain suffix like ".COM" as in, for example, "COMPANY.COM".
  - f When prompted by the QAS installer, provide the name and password of an AD user who has permission to create computer accounts (the user *must* have the *Create Computer Object* permission) in Active Directory. We refer to this as the "Aster Database service account" in AD.
  - g During VAS installation on queen ensure that self-enrollment is set to *off*.

At the end of this step, in the **Active Directory Users and Computers** MMC snap-in of the domain controller, you should see an entry for your queen node (cqueen in this example). This entry appears in the **Computers** node of the tree, under your domain (the asterengqa.com domain in this example).

- 4 Working as user beehive, run the QAS vastool utility to create a service account in AD for Aster Database.

```
# sudo /opt/quest/bin/vastool -u Administrator service create Aster Database/<DNS name of Aster Database queen machine>
```

As an example:

```
sudo /opt/quest/bin/vastool -u Administrator service create Aster Database/cqueen.asterengqa.com
```

When prompted, type the password for the Aster Database service account.

After the utility has created the account, you will see a message like this:

```
Service Aster Database/<DNS name of Aster Database queen machine> created successfully, keytab located at /etc/opt/quest/vas/ncluster.keytab.
```

- 5 Make user *beehive* the owner of the keytab file:

```
$ chown beehive:beehive /etc/opt/quest/vas/ncluster.keytab
```

- 6 Sync the coordinator's clock with the domain. The clocks must be in sync for authentications to succeed. You can use NTP for this, or you can use QAS's `vastool timesync` utility. Notice: If the AD server is in another time zone, running `vastool timesync` will cause the clock of the Aster Database queen to be moved to that time zone! To use `vastool`, working as root, type this:

```
# /opt/quest/bin/vastool timesync
```

- 7 Ensure the file `libvas-gssapi.so` exists at the following location:

```
# /opt/quest/lib64
```

If the file does not exist in the above location, please refer to QAS documentation for the location of 64-bit version of the file `libvas-gssapi.so` and create a symbolic link using the following command as user root:

```
# ln -s <full path of the file> /opt/quest/lib64/libvas-gssapi.so
```

- 8 On the DNS server, using the Windows *DNS* snap-in, navigate to the **Forward Lookup zones** node in the tree, open that node, and find your domain. Click on your domain to select it. Right click it, and choose the command **New Host (A or AAAA)...** to create a host entry for the Aster Database queen (`cqueen`) under your domain (`asterengqa.com` in this example). Test that your settings are working. From another host in the same domain, ping the queen hostname to see if the hostname resolves correctly. For example:

```
testuser10@vintela-ux:~$ ping cqueen
PING cqueen.asterengqa.com (10.50.55.100) 56(84) bytes of data.
64 bytes from 10.50.55.100: icmp_seq=1 ttl=63 time=0.394 ms
64 bytes from 10.50.55.100: icmp_seq=2 ttl=63 time=0.317 ms
64 bytes from 10.50.55.100: icmp_seq=3 ttl=63 time=0.378 ms

--- cqueen.asterengqa.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 10013ms
rtt min/avg/max/mdev = 0.317/0.363/0.394/0.033 ms
```

- 9 Run the Aster Database configuration utility to turn on AD authentication in Aster Database. After you perform this step, all users *except* beehive and db\_superuser will be authenticated against AD only. As root, run this:

```
# /home/beehive/bin/lib/configure/ConfigureNCluster.py --auth_type=GSS
```

- 10 On the Aster Database queen firewall, open the following ports to enable GSS communication. Working as root user on the queen node, open this file in a text editor: `/home/beehive/config/firewall.rules.user`. If the file does not exist, create it now. Add these entries to the file:

```
ufw allow 389
ufw allow 3268
ufw allow 88
ufw allow 53
ufw allow 123
```

- 11 Soft restart the cluster. Run the following as root user on the queen node:

```
# ncli system softrestart
```

After the restart, client authentication (for all users *except* beehive and db\_superuser) will be done through Active Directory. User beehive and user db\_superuser always authenticate using passwords only.

- 12 Check your firewall settings by logging in as root user on the queen and typing:

```
$ ufw status
```

In the output, you should see the required port numbers from the list above with an Action entry of ALLOW and a From entry of Anywhere.

**Next Step:**

Read the section, “[Active Directory Authentication](#)” on page 261

## Active Directory Authentication

Once you have turned on Active Directory (AD) authentication in Aster Database, all users (except beehive and db\_superuser) are authenticated against AD only. In other words, users authenticate to SQL databases using their AD username and password.

### **Create Users in an AD-Authenticated Aster Database**

To use AD authentication in Aster Database, each user must have two corresponding user accounts: one in Active Directory and one in Aster Database. The usernames must match, and the user account in Aster Database must have connect privileges to the databases the user will use.

To create a user account in Aster Database for a user who will authenticate using AD:

- 1 Use ACT to log into Aster Database as an administrator (a user with db\_admin rights).
- 2 Create the user account. Make sure the username matches his or her AD username. Only the username is used, not the domain name. For example, assume you have a user with the qualified AD username “ASTERDATA.COM/mjones”. To create this account in Aster Database:

```
CREATE USER mjones PASSWORD '1991e30m3'
```

- 3 Give the user at least connect privileges to the databases he or she will use. For example, you might type, in ACT:

```
GRANT ALL ON DATABASE na-sales-2010 TO na-dbusers;  
GRANT na-dbusers TO mjones;
```

Important things to note in this procedure:

- **Usernames:** To successfully AD-authenticate into Aster Database, the user’s username must adhere to the rules spelled out in the section, “[Rules for Active Directory Login Names for Connecting to Aster Database](#)” on page 262.
- **Passwords:** You’ll notice we specified a password. You must specify a password, even though it will not be used under normal circumstances. With AD authentication enabled, Aster Database ignores this password and delegates authentication to AD, instead. However, you should still use a secure password. In the event that Aster Database is switched back to local password authentication, this password becomes this user’s password.
- **Roles and groups:** To show a complete example, we put the user in a role with rights to a database. Note that Aster Database does *not* support AD group mapping to Aster Database roles in databases. Regardless of the authentication method you use, you must always manage users’ rights to specific Aster Database databases by creating roles in Aster Database and granting those roles to users.

### **Rules for Active Directory Login Names for Connecting to Aster Database**

Rules for usernames in Aster Database:

- 1 The username must start with a letter or an underscore; the rest of the string can contain letters, digits, and underscores.
- 2 The character limit is 63 chars. Longer entries are truncated.
- 3 The username cannot contain any of the following characters:

' " \

Rules for usernames in Windows:

- 1 If the username will be used as a Windows local logon name, it must be unique on the machine.
- 2 If it will be used as a global Windows logon name, then it has to be unique in the domain.
- 3 For every user account, Windows also creates a Microsoft Windows NT 4.0 logon name, which is usually the first twenty characters of the username. This shorter name, too, must be unique to the workstation or domain, depending on its scope of use.
- 4 The username cannot contain any of the following characters:  
| = , " + \* : ; ? [ ] / \ < >
- 5 Windows logon names can contain all other special characters, including underscores, dashes, and periods. Spaces are allowed, but strongly discouraged.

### **Log in to an AD-Authenticated Aster Database**

On Windows client machines that are part of the AD domain, when you log into Aster Database through Aster Database client tools such as ACT and ODBC, you need *not* specify a username and password. Typing your username and password is not needed because the Aster Database client obtains the current user's (logged-in user's) AD credentials from the Windows operating system and uses them to connect to the Aster Database queen. For the authentication to succeed, the user must have accounts set up as shown in “[Create Users in an AD-Authenticated Aster Database](#)” on page 261.

For Linux client machines, which also must be part of the AD domain, there are two modes:

- 1 **VAS Mapped User Mode:** If a Linux client is running QAS in *VAS Mapped User mode*, the user logs in to the machine with his or her local Linux username and Windows AD password. For the Aster Database authentication to succeed, Aster Database must contain a user account with a username that matches the Linux username of the logged-in Linux user who is trying to connect. That user must have CONNECT privileges to the database for which the connection is being attempted.

For example, if user Mick Jones has the Linux username *mjones* on the local machine where he's invoking the client, but his AD username is *MickJones*, then, for his authentication attempt to succeed, his username in Aster Database must be *mjones* (as created from the SQL prompt with CREATE USER).

The Aster Database client, when invoked by the user, obtains the logged-in user's AD credentials through QAS and uses them to connect to the Aster Database queen. For the authentication to succeed, the user must have accounts set up as shown in “[Create Users in](#)

[“Create Users in an AD-Authenticated Aster Database” on page 261.](#)

- 2 QAS Standard VAS Mode: If a Linux client machine is running QAS in *Standard VAS mode*, the user logs in to the machine with his or her AD username and password. For Aster Database authentication to succeed, Aster Database must contain a user account with a username that matches the AD username of the logged-in user who is trying to connect. That user must have CONNECT privileges to the database for which the connection is being attempted.

The Aster Database client, when invoked by the user, obtains the current user’s AD credentials through QAS and uses them to connect to the Aster Database queen. For the authentication to succeed, the user must have accounts set up as shown in [“Create Users in an AD-Authenticated Aster Database” on page 261.](#)

## Return to Password-Only Authentication

If your cluster currently uses AD or LDAP authentication, and you wish to switch back to Aster Database local authentication, do this:

- 1 Perform a soft shutdown on the cluster. Login to the queen as root user and run the following command:

```
# ncli system softshutdown
```

- 2 Execute the following command on the queen as root user to configure Aster Database to use local authentication:

```
# /home/beehive/bin/lib/configure/ConfigureNCluster.py --  
auth_type=PASSWD
```

- 3 If you were using GSS authentication, you must close the firewall ports that were opened for GSS. Working as root user on the queen node, open the following file in an editor: `/home/beehive/config/firewall.rules.user`. Either delete the following entries from the file, or put a hash symbol in front of each line to comment it out:

```
# ufw allow 389  
# ufw allow 3268  
# ufw allow 88  
# ufw allow 53  
# ufw allow 123
```

- 4 Soft restart the cluster. Run the following as root user on the queen node:

```
# ncli system softrestart
```

After the restart, client authentication will be done using Aster Database-stored passwords.

## Schema Search Path

Schema-qualified table names can be tedious to write, and analysts and engineers often prefer not to hardwire a particular schema name into their queries. As a result, tables are often referred to by unqualified names that consist of just the table name. When Aster Database encounters a table name that is not qualified with a schema name, Aster Database determines which table is meant by following a search path. The search path is an ordered list of schemas to search. The first matching table in the search path is taken to be the one wanted. If there is

no match in the search path, an error is reported, even if matching table names exist in other schemas in the database (but not in the search path).

You can specify a schema search path for a transaction or session (using SET), or as the default for a user (using ALTER USER).

The first schema named in the search path is called the current schema. Aside from being the first schema searched, it is also the schema in which new tables will be created if the CREATE TABLE command does not specify a schema name.

## Show Current Schema Search Path

To show the current search path, use the following command:

```
SHOW search_path;
```

The first schema in the search path that exists is the default location for creating new objects. When objects are referenced in any other context without schema qualification (table modification, data modification, or query commands) the search path is traversed until a matching object is found.

## Adding to the Schema Search Path

To put a new schema in the search path, use the SET search\_path command, as shown in this example:

```
CREATE SCHEMA myschema;
CREATE TABLE myschema.mytable (
    ...
);
SET search_path TO myschema,public;
```

After doing this, we can access the table without schema qualification:

```
DROP TABLE mytable;
```

Since myschema is the first element in the path, new objects would by default be created in it.

*Remember:* The search\_path must contain *all* the schemas in which you wish to search for database objects. For example, if we write:

```
SET search_path TO myschema;
```

...then we will not find tables in the public schema unless we explicitly qualify the table name as in, “public.mytable”.

## The search\_path Variable

```
search_path = string
```

The search\_path variable specifies the order in which schemas are searched when an object (table, view, etc.) is referenced by a simple name with no schema component. When there are objects of identical names in different schemas, the one found first in the search path is used. An object that is not in any of the schemas in the search path can only be referenced by specifying its containing schema with a qualified (dotted) name.

The value for search\_path has to be a comma-separated list of schema names. It is an error to specify a schema which does not exist.

When objects are created without specifying a particular target schema, they will be placed in the first schema listed in the search path.

Other effects can be obtained by altering the default search path setting, either globally or per-user.

## Setting the Schema Search Path for a Session or Transaction

This can be done on a session basis using the SET command:

```
SET {session | transaction} search_path { TO | = } { value }
```

## Setting a User's Default Schema Search Path

You can set a default search path for a user so that, when the user starts a session, the specified search path will overriding any global settings. To do this:

```
ALTER USER username SET search_path { TO | = } { value }
```

The default search path takes effect the next time the user logs in.

# System Internal Users

A system internal user is a specific type of Aster Database user, which has access to the local database instance on each node. The system internal users are:

- \_bee\_internal - the login used by loader and export
- \_bee\_sysman - the login use for logical backups
- \_bee\_sys - the system login used for inter-cluster operations

## Change System Internal User Passwords

As an added security measure, you may choose to change the password for the system internal users. The system internal user passwords must conform to the same [Password Rules](#) as for all Aster Database users. To change the system user passwords, you must use the arcpasswd script. Do not change the system user passwords directly on the PostgreSQL instances on the worker nodes. Once the password changes are complete, the new passwords are automatically applied to future node additions as well as to nodes that were down and rejoin the cluster at a later time.

### Procedure to Change System Internal Passwords

- 1 Log in to the queen as root.
- 2 Navigate to /home/beehive/bin/exec/  

```
# cd /home/beehive/bin/exec
```
- 3 Run the script arcpasswd, passing the name of the system user whose password you want to change:

```
# ./arcpasswd _bee_internal
```

- 4 Enter the current and new passwords for this user when prompted (by default, the password for each user is the same as the user id):

```
Enter current password for _bee_internal:  
Enter new password for _bee_internal:  
Re-enter new password for _bee_internal:
```

```
Password for _bee_sys successfully changed.
```

```
To change passwords for other users re-run this utility.  
When all desired passwords have been changed, restart the cluster  
for these changes to take effect. To accomplish this, enter:  
ncli system softrestart
```

Note that passwords are not echoed on screen as you type them.

- 5 Repeat the procedure to change the password for \_bee\_sysman and \_bee\_sys users.
- 6 Restart the cluster for the password change to take effect on all nodes:

```
# ncli system softrestart
```

- 7 Activate the cluster:

```
# ncli system activate
```



## CHAPTER 5 Workload Management

### Overview

Workload Management (WLM) for Aster Database includes two main areas of consideration: Admission Control and Resource Management.

Administrators have the option to enforce admission control by setting “Admission Limits” to determine when and if tasks (transactions, jobs, or queries) are allowed to be admitted into the system for processing. This is especially important if you have a particular type of query upon which other transactions depend. As an example, if you have call center or point of sale transactions that depend on other transactions, the administrator has the ability to control:

- Which tasks are allowed to be admitted.
- When tasks are allowed to run.
- How many tasks of a particular workload type may run concurrently.

Administrators control resource management by creating “Workload” settings and classifying different database process types into named profiles, called “workload policies” and applying a “service class” designation to each named workload. This allocates resources to particular workload types so that low-priority workloads (transactions, tasks, jobs, or queries) have a minimal impact on higher-priority workloads when they run concurrently.

This chapter explains how to set and control overall Workload Management and covers the following topics:

- [Admission Control](#)
- [Resource Management](#)
- [Best Practices](#)
- [Troubleshooting](#)

### Admission Control

Beginning in Aster Database 5.10, admission control enables you to create “Admission Limits” to set and manage predictable task execution, based on a particular expected workload.

Admission limits are configured using either the AMC or the command line to set specific admission limits and to set the global admission threshold or limit.

Admission limits are created using an arbitrarily ordered list of rules to apply admission limits to a particular transaction, executable, or query. These rules define the maximum number of queries of a specific type (those that match a specific predicate) that are allowed to run concurrently. Every admission limit is tied directly to one predicate (which must be a valid SQL WHERE clause) and requires each task (transaction, job, or query) to pass all predicates and admission limit counts before being admitted into the system.

The side effect of this is that a global admission threshold or limit can be set. The `ncli qos setconcurrency <concurrency>` command sets and then displays the maximum query concurrency. See “[ncli qos Section](#)” on page 179 for more details.

This setting can be used as a global admission threshold to hold all tasks under a certain concurrency limit. If the number of running tasks is at or above the set `nc_qos_concurrency` value, no new tasks are admitted to the system. These tasks are not denied, but rather queued. If this global admission threshold is not reached, admission limits determine if and when a task is admitted into the system.

## Predicate Evaluation

During predicate evaluation, every predicate is evaluated (or checked) against the “context” at that particular moment in time for that session. When a predicate check is matched, the number of tasks against that limit is checked to see if the admission limit is exceeded.

- If the limit is 0, the transaction is denied or terminated.
- If the limit is reached, the transaction is left in the queue.
- If it passes all of these checks it is admitted to the system.

For example, when administrators want to restrict certain users from submitting queries (during specific business hours) the limit for that user would be set to 0 and all queries for that user would be denied.

Admission control is performed at the transaction level, but is based on the “context” at that particular moment in time for that session. When a user makes a connection, it is a continuous session until that connection is ended. Within a session, one or more transactions may happen.

- Sessions are the high level container;
- Within a session there are transactions;
- Within a transaction, there are one or more statements.

The context of all three (session, transaction, and statement) at that particular moment in time constitutes the “context” against which predicates are evaluated.

A transaction that starts with a BEGIN can only be evaluated against that statement and it implies one or more statements may follow before the transaction progresses to the COMMIT, ABORT, or END phase. However, a single statement is still its own transaction and can be evaluated based on attributes within the statement when it is outside of an explicit transaction.

Statements in the same transaction will all execute once the transaction is admitted, however multiple transactions in the same session must each pass their own admission routine.

Note! Because predicate evaluation uses the “context” at that particular moment in time for that session (which includes some values from the statement) a statement of “BEGIN” or “END” may not work as one would expect against the table name checks or SQL-MR function names. This means that in the context of admission limits, transactions with multiple statements may not be evaluated as expected.

## Admission Limits via the Command Line Interface (ncli)

The ncli tool enables you to connect with the QosManager to set, edit, remove, or show the admission limits and to query the admission queue. For example, the command `ncli qos showadmissionqueue` displays a user-facing string to explain why a task is still in the queue and not yet admitted. A task might be in the queue because of the global limit or because of a particular admission limit. For a list of the available commands, see “[ncli qos Section](#)” on [page 179](#).

## Managing Concurrency

Administrators can use Admission Limits to control concurrency. This can be useful during upgrade or system maintenance windows when you need to block users from accessing the system.

### Block User Queries

To use Admission Limits to block user queries:

- 1 Log in to the AMC as an administrator (such as db\_superuser).
- 2 From the menu, select Admin > Configuration> Admission Limits.
- 3 Click the New Admission Limit button.
- 4 Fill in the admission limit information as follows:
  - a Provide a unique name such as "Block User Queries".
  - b Use != '\_bee\_sysman' for the predicate.
  - c Set the concurrency limit to 0.
  - d Click OK.
- 5 Now queries from users other than \_bee\_sysman will fail with the error "Admission denied". Note that any transactions that are currently running will continue to execute unless they are individually cancelled.

### Re-enable User Queries

To undo the [Block User Queries](#) changes and bring the cluster back on-line:

- 1 Log in to the AMC as an administrator (such as db\_superuser).
- 2 From the menu, select Admin > Configuration> Admission Limits.
- 3 Find the Admission Limit you created previously to block user queries in the list.

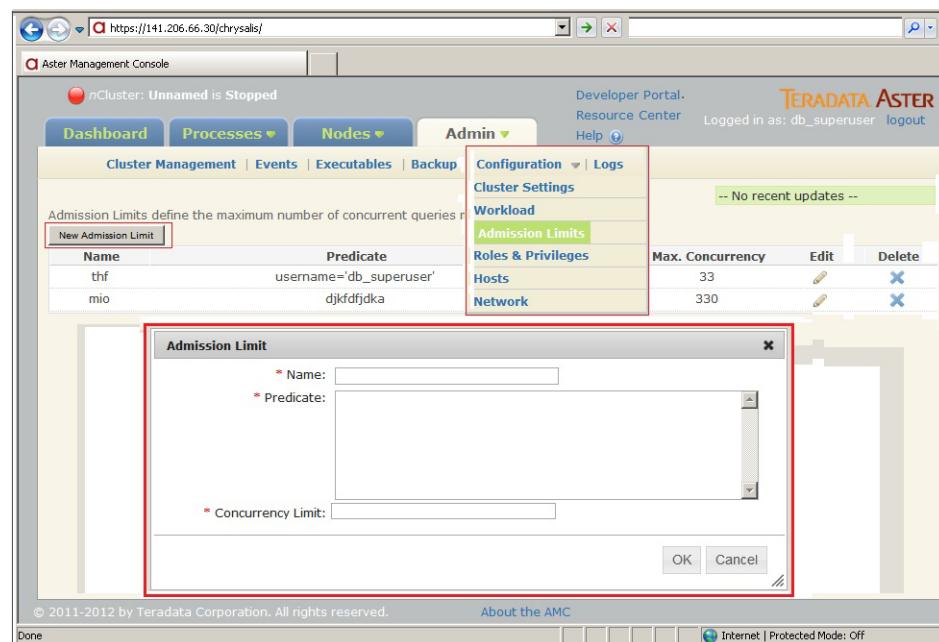
- 4 Click the X to delete it. Confirm deletion when prompted.

## Configure Admission Limits using the AMC

Open the Aster Database Management Console (AMC) in your browser.

- 1 To set Admission Limits, navigate to **Admin > Configuration > Admission Limits** tab of the AMC.
- 2 In the Admission Limits screen, click the **New Admission Limit** button. The Admission Limit window appears.
- 3 In the Admission Limit window, set the three required fields and click **OK**:
  - **Name:** A unique name for the Admission Limit title
  - **Predicate:** This must be a valid SQL WHERE clause
  - **Concurrency Limit:** This number sets the maximum concurrency limit

Figure 14: The Admission Limits Tab



The Admission Limits screen now contains the newly created admission limit.

To modify or remove a specific Admission Limit, click the **Edit** or **Delete** button.

## Resource Management

Administrators control resource management by specifying “Workload” settings and classifying different database process types into named profiles, called “workload policies” and applying a “service class” designation to each named workload. This allocates resources to particular workload types and is configured through the AMC.

In contrast to Admission Limit settings, workload management settings are created as an ordered list where the first match is the one applied. Processes are assigned a workload upon submission and re-evaluated periodically. Based on the Workload Policy and associated Service Class, the priority and resource allocation are assigned to the process. See “[Setting Up Workload Rules in the AMC](#)” on page 276 for information on using the AMC to configure workload management.

Administrators create rules that group database operations into *workloads*, such as:

- Physical backups
- ETL operations
- All queries generated by members of the Sales department
- Reports against the table “daily\_summary”
- Administrative operations

Administrators create rules, known as “[Service Classes](#)” on page 273, to assign to each workload. The service class determines the importance with which queries of a particular workload type will be run. The share of system resources allocated to a service class is a function of its:

- Priority - determines the priority of a workload and is a first-level control on admission to the queue for processing and resource usage (CPU and disk I/O).
- Weight - determines the priority and resource (CPU and disk I/O) allocation for a workload by percentage and is a second-level control on admission to the queue and resource usage.
- Memory soft limit - the highest percentage of memory the workload can occupy when there is contention for memory in the system.
- Memory hard limit - the highest percentage of memory the workload can occupy, even there is no contention for memory in the system.

These rules instruct Aster Database to run each type of job (transaction, task, or query) with the expected level of urgency. Based on your rules, Aster Database assigns an initial level of importance to each job and, if warranted, re-ranks the job while it is running. For example, these rules can ensure high resource allocation for a newly added query of a given type but throttle down resources for that query if it runs so long that it is suspected of being a runaway query.

The “[Workload Policies](#)” on page 275 evaluate incoming database tasks and groups them with like operations, based on criteria you define. Administrators set workload policies by writing a standard filter, called a “predicate,” that groups like operations at processing time. A predicate’s syntax is identical to that of a SQL WHERE clause. The predicates can evaluate a large set of attributes of the job and its context, such as username, user group/role, time-of-day, elapsed time, database/table name, workload type (for example load, backup, batch query, or interactive query). For example, you might have a workload policy called “adminOps” that applies to any job that an administrator runs in Aster Database.



**Important!** There is always a default policy that is applied for any job that does not match one of the preceding workload policies. See “[Default Workload Policies](#)” on page 275 for more details.

## Workload Settings via the Command Line Interface (ncli)

The ncli tool enables you to connect with the QosManager to set, edit, remove, or show the workload settings, show all of the QoS-related data, service classes, and the session details. For example, the command `ncli qos showevalstats` displays predicate evaluation statistics. For a list of the available commands, see “[ncli qos Section](#)” on page 179.

## Service Classes

Service classes define how Aster Database divides the available hardware resources among all active statements and other activities (e.g., physical backups) in the system. Each service class has a *priority*, a *weight*, a *soft memory limit percent*, and a *hard memory limit percent*. These settings apply to all statements that map to that service class. Statements are mapped to workloads, which are then mapped to service classes.

### Priority

Priority is the first-level setting that governs admission to the queue for processing. That is, incoming requests are first evaluated and prioritized by their priority, and then by weight. These two settings also map to a per-node CPU share control and a per-node IO priority.

The priority value is an integer between 0 and 3, inclusive, that establishes, at the coarsest level, how important a job is. A higher priority value indicates a job of higher importance. You set a service class’s priority when you create the class in the Admin > Configuration > Workload > Service Classes tab of the AMC. The priority value will be one of the following: Deny, Low, Medium, or High. This is recorded as an integer code in the `priority` column of the `nc_qos_service_class` table.

The codes and their mappings are shown in the table below.

Table 5 - 1: Priority names in the AMC

| Priority | Name in AMC |
|----------|-------------|
| 0        | Deny        |
| 1        | Low         |
| 2        | Medium      |
| 3        | High        |

*Priority 0* (zero) indicates a job that will not be allowed to run, *priority 1* a very low-importance job, *priority 2* a medium-priority job, and *priority 3* a high-importance job. You can prevent a query from starting by having it map to a *priority zero* service class at the outset, but you cannot use priority zero to stop an already-running query. Priority levels 1, 2, and 3 can be applied to a running query.

Reasons for using priority 0 (deny) might be that you wish to disallow any queries against a particular table during certain hours when the daily sales reports are run. Or you may wish to block certain categories of users from running queries during peak hours.

## Weight

Within a priority level, the weight value dictates the ratio of resource allocation. For example, if two statements execute with the same *priority*, but with *weight* values of 80 and 20, the system will aim to allocate resources in a 4:1 ratio, with most of the resources allocated to the statement with higher weight. Allocation of I/O-related resources is less accurate than allocation of CPU shares, so in this example, the CPU share ratio would be very close to 4:1, while the disk I/O shares cannot be guaranteed as precisely.

## Priority Levels, Weight Values, and Resource Allocation

Given that weight values can vary between 1 and 100, within a single priority level you can allocate two orders of magnitude more resources to one class than the other. Furthermore, the lowest weight within a priority level (e.g., priority 2 and weight 1) receives approximately an order of magnitude more resources than the highest weight level in the next lower priority level (e.g., priority 1, weight 100). In summary, there is a significant difference in terms of the resources (CPU and disk) assigned to two statements executing at different *priority* levels, because the system ensures that most resources are allocated to the highest-priority job.

Unless you need the granularity of a very large number of workloads, your workload definitions can use just priority 2 and priority 3 for all jobs.

## Memory Soft Limit Percent

A memory soft limit is a best-effort limit on the amount of memory collectively used by all processes at a given node running under a service class. The limit is associated with the service class and therefore affects all queries running under all workloads that map to that service class. When there is no memory pressure and resources are available, a service class is allowed to utilize more memory per node than defined by its soft limit. When memory is needed, the system will strive to reduce the memory used by a service class to its soft limit, but the success of this operation is not guaranteed.

A soft limit is defined as a percentage of physical memory (RAM) on a per physical node basis. For each service class, the percentage is a value between 0% and 100%. Note that given the use of swap space, the total allocated percentage among all service classes is not limited to a value between 0-100%. An administrator assigns a soft limit to each service class and the cumulative percentage of all soft limits defined for all service classes may add up to more than 100%, again due to the use of swap. The system will not reject such high values.

Under extreme memory pressure, service classes using more memory than their soft limits are considered to be over quota. In such situations, queries and other activities executing under that service class may be canceled (for more information see “[Setting Up Workload Rules in the AMC](#)” on page 276).

## Memory Hard Limit Percent

A memory hard limit defines an upper bound to the amount of memory used by a service class on each physical node. The limit is associated with the service class and therefore affects all queries running under all workloads that map to that service class. Even when resources are available and the system is not under memory pressure, a service class will not be allowed to

consume more memory than defined by its hard limit (i.e. it is always considered over quota when trying to go beyond its hard limit).

Like a soft limit, a hard limit is defined as a percentage of physical memory (RAM) on a per physical node basis. Again because of swap, hard limits and/or their sum can be higher than 100%.

### Automatic Query Cancellation

Queries in a given service class will use swap space when the service class hard limit is reached, or when the node is under memory pressure. Queries are automatically cancelled by the system when they use substantial amounts of swap space. The allowed amount of swap space queries can use is governed by the following rules:

- Queries in service classes with a hard limit are canceled when the service class swap usage goes above 1 GB.
- Queries in service classes without a hard limit are canceled when the service class swap usage goes above 10 GB.
- Queries without an assigned service class fall under the default service class, which is required for WLM. See [Step 2: Create the Default Workload Policy \(page 278\)](#).

## Workload Policies

A workload policy defines a set of related activities (SQL queries being the most common) that share similar properties. Assigning these related activities to a service class allows them to be prioritized in a similar manner by Aster Database.

Before any statement starts executing, it is mapped to a workload policy. Workload policies contain a predicate attribute that specifies a boolean clause that evaluates to `true` for all activities that are to be mapped to that workload. You define this predicate to evaluate the attributes of the statement and its context. For instance, a workload containing all queries issued by user `beehive` could be defined with the predicate `username = 'beehive'`.

The workload policy then specifies the service class under which the statement will be executed. In the examples in this document we focus on SQL statements, but the mechanism described here also applies to other activities including physical backup and restore operations.

### Default Workload Policies

When in use, workload management requires that at least one default workload policy be defined. A default workload policy is one which matches any activity in the system and it is defined by a predicate of '`true`'. This ensures that any activity in the system can be mapped to at least one workload policy. Note that the default workload policy should appear last in the evaluation order, so that it will be applied only if none of the other workload policies apply.

Figure 15: Default Workload Policy setup

The screenshot shows the 'Workload Policies' section of the AMC. It lists five workload policies with the following details:

| Eval Order | Workload Policy Name | Predicate               | Status | Service Class    | Priority | Weight | Action |
|------------|----------------------|-------------------------|--------|------------------|----------|--------|--------|
| 1          | AdminOps             | username='db_superuser' | Active | AdminClass       | High     | 75     |        |
| 2          | BI Tools             | dbname='reports'        | Active | InteractiveClass | Medium   | 35     |        |
| 3          | Loads                | username='etl'          | Active | ETLClass         | Medium   | 30     |        |
| 4          | Backups              | username='backup'       | Active | BackupClass      | Low      | 25     |        |
| 5          | default              | true                    | Active | defaultClass     | Low      | 10     |        |

When you install Aster Database, no workload policies are provided. Before you define any other workload policy, you must first define the *default workload policy* as explained in “[Step 2: Create the Default Workload Policy](#)” on page 278.

## Setting Up Workload Rules in the AMC

The following sections describe the process of setting up workload rules in the AMC, including the creation of service classes and workload policies.

### Preparatory Steps

- Consider all the types of queries that users run on your Aster Database, and decide on a name for each query type.  
This named query type is called a workload in Aster Database, and it will be subject to a specific *workload policy* that you write. Make a note of the defining characteristics that determine a query's membership in each workload. For example, you might create a workload policy called “LOAD” defined either as all SQL COPY statements or all statements executed by the special username “etl”, whichever best fits your environment.
- Decide how many service classes you need.  
In most cases, it is enough to have only a few service classes.
- Decide on a name, priority, weight, memory soft limit, and memory hard limit for each.

Multiple workloads can point to a service class. A service class is not required to have exactly one workload that uses it.

### Step 1: Create Service Classes

- Log into the AMC as a user with *amc\_admin* or *node\_admin* rights.
- Click Admin > Configuration > Workload.

Figure 16: Accessing Workload Management in the AMC



3 Click the Service Classes tab.

Figure 17: The Service Classes Tab

The screenshot shows the AMC's main dashboard with tabs for 'Dashboard', 'Processes', 'Nodes', 'Admin' (selected), 'Cluster Management', 'Events', 'Executables', 'Backup', 'Configuration' (selected), and 'Logs'. Below the tabs, there's a 'Save & Apply Changes' button and two tabs: 'Workload Policies' and 'Service Classes' (selected). A message says 'Service classes provide the assignment of priority and weight.' Below is a table for 'New Service Class' with columns: Service Class Name, Priority, Weight, Memory Soft Limit Pct, Memory Hard Limit Pct, Edit, and Delete. The 'Service Class Name' column has 'InteractiveClass' entered.

4 Create one or more New Service Classes.

- Define the new Service Class Policy by completing the following fields:
  - Name in the Service Class Name field
  - Priority (deny, low, medium, or high)
  - Weight value between 0 and 100
  - Memory Soft Limit Percent value between 1 and 100
  - Memory Hard Limit Percent value between 1 and 100

Figure 18: Service Class Policy

The dialog box is titled 'Service Class Policy'. It contains the following fields:
 

- \* Service Class Name:
- Priority:
- \* Weight:  (Enter a number between 1 and 100)
- Memory Soft Limit Pct:  (Enter a number between 1 and 100 or leave as blank)
- Memory Hard Limit Pct:  (Enter a number between 1 and 100 or leave as blank)

 At the bottom are 'OK' and 'Cancel' buttons.

For information on the settings, see “Service Classes” on page 273.

- Click OK.
- Click Save & Apply Changes.

Note: You must click **Save & Apply Changes** every time a Service Class is added or edited.

- 5 Click Yes to apply all changes.

Figure 19: Save and Apply Changes to a Service Class

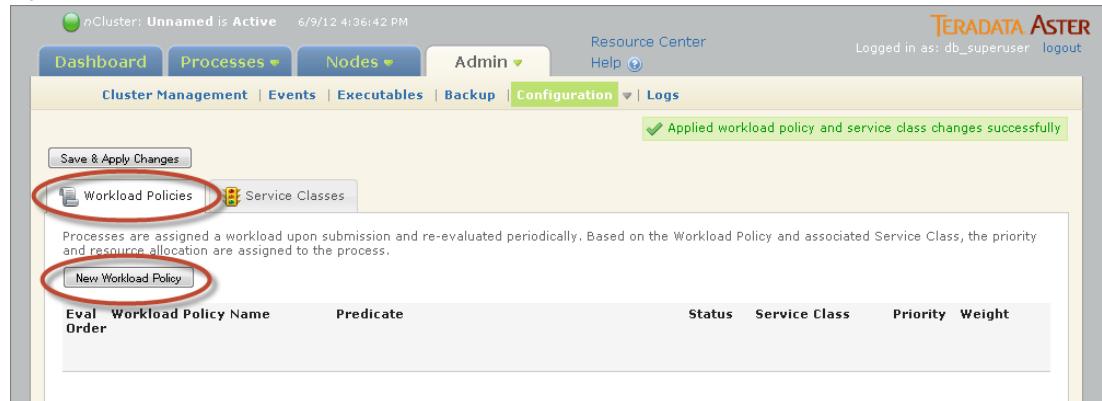


### Step 2: Create the Default Workload Policy

Before you define any other workload policy, you must first define the “[Default Workload Policies](#)” on page 275. This will be the policy used for any workload or which does not fit the predicate for any other policy. Follow the steps below to create the default workload policy:

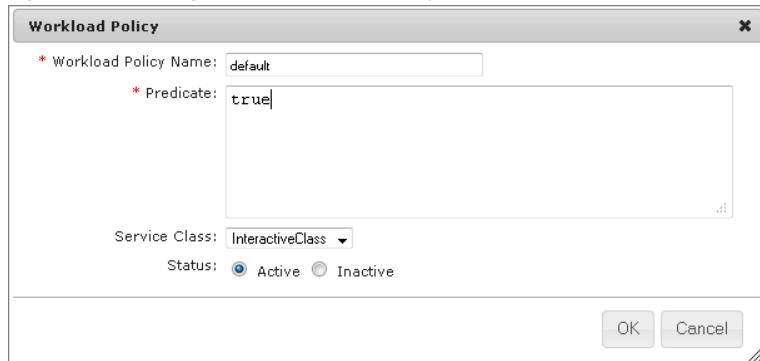
- 1 In the AMC, navigate to the Admin > Configuration > Workload.
- 2 Click the Workload Policies tab.

Figure 20: The Workload Policies Tab



- 3 Click the New Workload Policy button.
- 4 Give the policy a name (usually, “default” is best).
- 5 Type true in the Predicate field.
- 6 Choose a Service Class that will establish the running priority of default workloads.
- 7 Click Active.
- 8 Click OK.

Figure 21: Creating a new Workload Policy



- 9 Click the Save & Apply Changes button.

### **Step 3: Create the Workload Policies**

Follow the steps below to create workload policy definitions. Each has a name and a link to a service\_class that will determine its priority relative to other workloads. Each has a predicate clause that determines which submitted queries will be considered part of this workload policy. (We sometimes also call this the *QoS context* of the query, where QoS stands for “quality of service”.) Write the predicate as you would a WHERE clause. For a list of criteria you can use, see “[Workload Policy Predicate Attributes](#)” on page 282.

- 1 In the AMC’s Admin > Configuration > Workload > Workload Policies tab, click the New Workload Policy button.
- 2 Give the policy a name.
- 3 Type the query selection criteria in the Predicate field. See “[Workload Policy Predicate Attributes](#)” on page 282 and “[Workload Policy Predicate Syntax](#)” on page 284.
- 4 Choose a Service Class.
- 5 Click Active. Note that you can choose Inactive if you wish to make a workload inactive, without deleting it.
- 6 Click OK.
- 7 Click the Save & Apply Changes button. Do this after every workload policy you create.
- 8 Repeat the preceding step for all workload policies you wish to add.
- 9 Make sure you click Save and Apply Changes after adding the last policy.

### **Step 4: Set the Evaluation Order of Workload Policies**

At evaluation time, the first match wins. In other words, when a user submits a query, the first workload policy whose predicate matches the user’s query will be used. Because of this, you will always want the default policy to appear last in the evaluation order. Follow the steps below to set the workload policy evaluation order:

- 1 In the **Workload Policies** tab, click on a policy's row and drag it up or down.

Figure 22: Changing a Workload Policy's evaluation order

| Eval Order | Workload Policy Name | Predicate               |
|------------|----------------------|-------------------------|
| 1          | AdminOps             | username='db_superuser' |
| 2          | BI Tools             | dbname='reports'        |
| 3          | Loads                | username='etl'          |
| 4          | Backups              | username='backup'       |
| 5          | default              | true                    |

- 2 Repeat for the other policies, dragging each row to the right place in the order.
- 3 Click **Save and Apply Changes** to apply the sort order.

#### **Step 5: Edit or Delete Workload Policies and Service Classes**

You may edit or delete the workload policy definitions and service classes at any time. Note that doing so will cause Aster Database to re-evaluate resource allocation, admission, and position in the queue. This also holds true for any new rules based on elapsed time. The planner will re-evaluate the new workload settings for all queued and running processes, canceling or reprioritizing them as necessary to comply with the new rules.

Note: By creating a predicate of "true" and setting the admission limit, for example, to 50 or 100 all tasks would match to "true" then all tasks would be held under the limit.

- 1 Navigate to the AMC screen **Admin > Configuration > Workload**.
- 2 Click the appropriate tab, either **Workload Policies** or **Service Classes**.
- 3 Click the **Edit** icon, which looks like a pencil, or to delete the item, click the **X** icon.
- 4 Change the desired settings to the new values.
- 5 Click **OK**.
- 6 Click the **Save & Apply Changes** button. Do this after every workload policy or service class you edit.
- 7 Repeat the preceding step for all items you wish to edit.
- 8 Make sure you click **Save and Apply Changes** after editing the last item.

#### **Step 6: Test Your Workload Policies**

To test the policies you have saved, follow these steps:

- 1 In the AMC, working as a user with administrator rights, go to the **Processes > Processes** tab.
- 2 Open an ACT session, and run queries that meet the criteria of your rules.
- 3 In the **Workload Policy** column of the **Processes** tab, check that each test query matches the desired policy. You can click on the policy name to be reminded of its priority and weight.

Figure 23: Testing Workload Policies

The screenshot shows the Teradata Aster Admin interface with the 'Processes' tab selected. A tooltip is displayed over a table row for a query, providing details about the service class: 'Service Class: InteractiveClass', 'Priority: Medium', and 'Weight: 35'. The main table lists various database operations with their execution times and session IDs. A red oval highlights the tooltip area.

| ID      | Statement                              | User         | Status | Execution Time          | Type                 | Workload Policy | Priority    | Session ID  | Cancel |
|---------|----------------------------------------|--------------|--------|-------------------------|----------------------|-----------------|-------------|-------------|--------|
| 5443... | select * from reports_table;           | beehive      | ✓      | 01 sec 6                | BI Tools             | Medium          | 32426812... |             |        |
| 8195... | select * from reports_table;           | db_superuser | ✓      | 01 ms 6                 | QL AdminOps          | High            | 54717610... |             |        |
| 2046... | create table reports_table (id int,... | db_superuser | ✓      | 01 sec 6/11/12 10:39 AM | 6/11/12 10:39 AM SQL | AdminOps        | High        | 54717610... |        |
| 3946... | create tab...                          | db_superuser | ✗      | 01 sec 6/11/12 10:39 AM | 6/11/12 10:39 AM SQL | AdminOps        | High        | 54717610... |        |
| 8450... | create database reports;               | db_superuser | ✓      | 02 sec 6/11/12 10:38 AM | 6/11/12 10:38 AM SQL | AdminOps        | High        | 19383530... |        |
| 2880... | select * from sales_fact;              | db_superuser | ✓      | 01 min 6/11/12 10:37 AM | 6/11/12 10:38 AM SQL | AdminOps        | High        | 19383530... |        |
| 4015... | select * from sales_fact;              | beehive      | ✓      | 36 sec 6/11/12 10:35 AM | 6/11/12 10:35 AM SQL | default         | Low         | 63407629... |        |

## Workload Configuration Example

As an example, we will show the configuration of a scenario with the following workloads:

- 1 *Loading operations*, which are assumed to be executed using the 'etl' username;
- 2 *Interactive queries* against database 'reports', which only contains the tables needed for these queries;
- 3 *Backups*: Full and incremental physical backup operations;
- 4 *Statements executed by any system administrator*. We assume that all administrators have been granted the database role admin;
- 5 The *default workload*, which includes all other statements and activities.

In the Aster Database AMC, in the **Admin > Configuration > Workload > Service Classes** tab, you can define service classes for each of the workloads above. Seen in the system tables, these classes look like:

Table 5 - 2: Example Service Class definitions (nc\_qos\_service\_class)

| name             | priority   | weight | memory soft limit | memory hard limit |
|------------------|------------|--------|-------------------|-------------------|
| InteractiveClass | 3 (high)   | 60     | 60                | 75                |
| AdminClass       | 3 (high)   | 10     | 10                | 15                |
| EtlClass         | 2 (medium) | 50     | 50                | 65                |
| DefaultClass     | 2 (medium) | 30     | 30                | 35                |
| BackupClass      | 2 (medium) | 10     | 10                | 20                |

Note that:

- 1 We have set the interactive queries and the admin operations to have the highest priority (3) and defined the weight values so that interactive queries get approximately 6 times more resources than admin operations when both service classes are active;
- 2 We defined all other service classes with priority of 2, using the weights to allocate resources between ETL operations (loads), backup operations, and all other activities using a 5:3:1 ratio.

With the service classes defined, we can now (in the AMC's **Admin > Configuration > Workload > Workload Policies** tab) define the following workload policies. Seen in the system tables, the policies look like:

Table 5 - 3: Example workload policies

| evaluation_order | name        | active | predicate            | service_class_name |
|------------------|-------------|--------|----------------------|--------------------|
| 1                | adminOps    | t      | 'admin' = ANY(roles) | AdminClass         |
| 2                | interactive | t      | dbName = 'reports'   | InteractiveClass   |
| 3                | loads       | t      | userName = 'etl'     | EtlClass           |
| 4                | backups     | t      | activity = 'Backup'  | BackupClass        |
| 5                | default     | t      | true                 | DefaultClass       |

One important observation is that the *evaluation order* is the deciding factor when a given activity maps to more than one workload policy. In our configuration, for example, any activity performed by an admin will map to the "AdminClass," even if he or she is performing a load or executing an interactive query. If this behavior is not the intended one, you must modify the evaluation order values accordingly.

## Workload Policy Predicate Attributes

A workload policy governs only those queries that are mapped to it. Queries are mapped to workload policies using the evaluation order you specify (see "[Step 4: Set the Evaluation Order of Workload Policies](#)" on page 279). The first match between a query and the workload policy predicate results the query being mapped to that workload policy.

You build the workload predicate using the pre-defined WLM attributes listed in the table below. WLM attributes are SQL-typed values that contain information about the query itself or about the user or session that ran the query. When you write a predicate, the datatype of your test value(s) must match the datatype of the attribute being tested.

The WLM attributes and their associated values are assigned to the query when it is planned by the system. For example, the attribute 'userName' has its value assigned during connection establishment and contains the username provided by the user, while the attribute 'stmtType' contains the type of statement being executed (e.g., a SELECT or an INSERT statement).

Because WLM attributes are SQL-typed, you can use existing UDFs and SQL operators when specifying workload predicates. Like in a WHERE clause, any construct that is compatible with the attribute's type is accepted. WLM predicates may include AND and OR conjunctions, with as much complexity as desired. For example, the predicate:

```
stmtElapsedTime > '30 min' AND 'analyticsTeam'=ANY(roles)
```

returns true for any statement executed by any user on the analyticsTeam group but only after the statement has been running for more than thirty minutes.

The attributes you can evaluate in your predicate, listed in the following table, are also listed in the Aster Database system table, nc\_qos\_workload\_variables.

Table 5 - 4: Workload policy predicate attributes

| Name             | Type       | Description                                                                                                                                                                                                                                                                                                 | Example                                               |
|------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| activity         | VARCHAR    | Aster Database high-level operation type. The only valid value to specify is 'Backup' which matches Aster Database Backup operations.                                                                                                                                                                       | activity='Backup'                                     |
| clientIpAddr     | INET       | The IP address of the machine from which the client is connected. The example matches all client IP addresses in the subnet 192.1.1/24.                                                                                                                                                                     | clientIpAddr << '192.1.1/24'                          |
| connTime         | TIMESTAMP  | Time the connection to Aster Database was established.                                                                                                                                                                                                                                                      | (EXTRACT('HOUR' FROM connTime)) BETWEEN 8 AND 18      |
| currentTime      | TIMESTAMP  | The current time.                                                                                                                                                                                                                                                                                           | (EXTRACT('HOUR' FROM currentTime)) BETWEEN 8 AND 18   |
| dbName           | VARCHAR    | The name of the database the client is connected to.                                                                                                                                                                                                                                                        | dbName LIKE 'db_clicks_%'                             |
| is_ddl(stmtType) | BOOLEAN    | Returns true if the statement is a DDL. The following are DDLs: ALTER, DROP, GRANT, and REVOKE statements.                                                                                                                                                                                                  | is_ddl(stmtType)                                      |
| is_dml(stmtType) | BOOLEAN    | Returns true if the statement is a DML. The following are DMLs: CopyFrom, Insert, Update, Delete, and Truncate.                                                                                                                                                                                             | is_dml(stmtType)                                      |
| roles            | VARCHAR [] | The set of roles of which the current user is a member, either directly or indirectly. The example matches all users who have been granted the 'admin' role.                                                                                                                                                | 'admin'=ANY(roles)                                    |
| sqlmrFns         | VARCHAR [] | An array with the names of SQL-MR functions.                                                                                                                                                                                                                                                                | 'npath'=ANY(sqlmrFns)                                 |
| stmtElapsedTime  | INTERVAL   | The time elapsed since the current statement started executing. You can use this attribute in your rules to have Aster Database re-prioritize running tasks after they have been running for a specified amount of time. The example matches all statements during the first 10 minutes of their execution. | stmtElapsedTime < '10 min'                            |
| stmtStartTime    | TIMESTAMP  | The time the current statement started executing.                                                                                                                                                                                                                                                           | (EXTRACT('HOUR' FROM stmtStartTime)) BETWEEN 8 AND 18 |

Table 5 - 4: Workload policy predicate attributes (continued)

| Name          | Type       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Example                                             |
|---------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| stmtType      | VARCHAR    | The type of statement being executed. Its possible values are: Alter, AlterIndex, AlterRole, AlterTable, Close, Cluster, CopyFrom, CopyTo, CreateDatabase, CreateIndex, CreateRole, CreateSchema, CreateTable, CreateTableAs, CreateTempTable, CreateView, Declare, Delete, Drop, DropDatabase, DropIndex, DropRole, DropTable, Explain, Fetch, Grant, GrantRole, Insert, Lock, Reindex, Revoke, RevokeRole, Select, Set, Show, Transaction, Truncate, Update, and Vacuum. | stmtType LIKE 'Alter%'                              |
|               |            | Each of these maps to the SQL statement of the same name, except for “Transaction”, which maps to BEGIN, END, COMMIT, and ABORT statements. See also <code>is_ddl</code> and <code>is_dml</code> , above.                                                                                                                                                                                                                                                                  |                                                     |
| tableNames    | VARCHAR [] | An array with the schema-qualified names of all non-temporary tables involved in a statement. You can omit the schema name for tables in the public schema.                                                                                                                                                                                                                                                                                                                | 'sales_fact' = ANY(tableNames)                      |
| true          | BOOLEAN    | Useful for a default (catch-all) rule. Since a predicate of true will match all statements, you will typically want to place it last in the workload evaluation order.                                                                                                                                                                                                                                                                                                     | true                                                |
| txElapsedTime | INTERVAL   | The time elapsed since the current transaction started.                                                                                                                                                                                                                                                                                                                                                                                                                    | txElapsedTime < '10 min'                            |
| txStartTime   | TIMESTAMP  | The time the current transaction started.                                                                                                                                                                                                                                                                                                                                                                                                                                  | (EXTRACT('HOUR' FROM txStartTime)) BETWEEN 8 AND 18 |
| userName      | VARCHAR    | The username of a query connected to the database. This does not change during the established session.                                                                                                                                                                                                                                                                                                                                                                    | userName='etl_user'                                 |

## Workload Policy Predicate Syntax

QoS context attributes are SQL-typed, which allows you to use existing UDFs and SQL operators when specifying workload predicates. Like in a WHERE clause, any construct that is compatible with the attribute's type is accepted. For example, given that 'dbName' is an attribute with type VARCHAR, a workload policy defined with a predicate `dbName like 'db_clicks_%'` would match any statement executed while connected to any database with a name that starts with `db_clicks_`.

Below, we show some examples of predicates - that is, clauses that can be used within a predicate - that use an assortment of SQL-compatible expressions.

Table 5 - 5: Example workload policy predicates

| Predicate                                                                                      | Description                                                                                                                                        |
|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>clientIpAddr &lt;&lt; '192.1.1/24'</code>                                                | Client IP addresses contained within subnet 192.1.1/24.                                                                                            |
| <code>'admin' = ANY(roles)</code>                                                              | True for any user that has been granted the 'admin' role.                                                                                          |
| <code>stmtElapsedTime &lt; '10 min'</code>                                                     | Matches the execution of any statement for the first 10 minutes.                                                                                   |
| <code>stmtType like 'Alter%'</code>                                                            | Matches all ALTER statement types.                                                                                                                 |
| <code>stmtElapsedTime &gt; '30 min'</code><br>and<br><code>'analyticsTeam' = ANY(roles)</code> | True for any statement executed by any user that is part of the analyticsTeam group but only after the statement has executed for over 30 minutes. |

In addition to all functions and operators provided by PostgreSQL, Aster Database also introduces a set of functions, as listed below.

Table 5 - 6: Workload predicate functions

| Function            | Description                                                                                                                                                                                                                                                                                    |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>is_ddl</code> | Takes the statement type (a VARCHAR attribute) as argument and returns true if the statement is a DDL. The following stmtType values are considered DDLs: all ALTER, DROP, GRANT, and REVOKE statements. For example, the clause <code>is_ddl(stmtType)</code> would match all DDL statements. |
| <code>is_dml</code> | Takes the statement type as argument and returns true if the statement is a DML. The following stmtType values are considered DMLs: CopyFrom, Insert, Update, Delete, and Truncate.                                                                                                            |

## Resource Allocation and Service Classes

Each service class defines the unit for resource allocation in the system independently of how many queries - and indirectly, processes - execute under it. One consequence of this property is that if your configuration has two service classes whose priority/weight settings are identical, then if you run many concurrent queries, some of which match one class and some of which match the other, you will not see the same results as if you had run many concurrent queries that all matched just one class.

For example, consider the following simple service class configuration:

Table 5 - 7: Example Simple Service Class Configuration

| Service Class      | Priority | Weight | Memory soft limit | Memory hard limit |
|--------------------|----------|--------|-------------------|-------------------|
| InteractiveQueries | 3        | 30     | 30                | 35                |
| AdminStmts         | 3        | 30     | 30                | 45                |
| CEOQueries         | 3        | 10     | 10                | 20                |

Despite the fact that they have the same priority and weight settings, we have separated interactive queries from the statements issued by the administrator. If these are the only two service classes active in the system at a given point in time, in other words, no active statement maps to any other service class - the configuration above stipulates that each get the same share of resources, or 50% each. For the allocation of CPU resources this is the case even if the administrator is issuing a single SQL statement while 99 concurrent interactive queries are executing. Instead of receiving only 1% of the available resources, that single admin query will in fact receive roughly the same share as all the interactive queries put together! Note that I/O resource allocation is not as fine-grained as CPU time allocation, so Aster Database performs this in a best-effort manner.

Similarly, if only the CEO and the administrator have active statements executing in the system, all admin statements would collectively receive 3 times more resources than all CEO queries put together.

## Monitoring the Priority Execution of Queries using the AMC

The AMC allows you to monitor query execution, by showing the workload policy that was mapped to each query. In the AMC, you can view this in the **Processes** tab. Use a filter if needed to load the statistics for the desired queries, and look at the **Workload Policy** column and the **Priority** column.

Figure 24: Monitoring priority execution of queries in the AMC

| ID      | Statement                              | User         | Status                              | Execution Time | Submit Time      | Completion Time  | Type | Workload Policy | Priority | Session ID | Cancel                             |
|---------|----------------------------------------|--------------|-------------------------------------|----------------|------------------|------------------|------|-----------------|----------|------------|------------------------------------|
| 7910... | select * from sales_fact;              | beehive      | <span style="color:blue;">🕒</span>  | 12 sec         | 6/11/12 1:06 PM  |                  | SQL  | default         | Low      | 2027...    | <span style="color:blue;">✖</span> |
| 4245... | select * from "sales_fact" ;           | beehive      | <span style="color:red;">✗</span>   | 01 ms          | 6/11/12 1:06 PM  | 6/11/12 1:06 PM  | SQL  | default         | Low      | 2690...    |                                    |
| 7666... | select * from...                       | beehive      | <span style="color:green;">✓</span> | 01 sec         | 6/11/12 12:49 PM | 6/11/12 12:49 PM | SQL  | default         | Low      | 2690...    |                                    |
| 7682... | select * from...                       | beehive      | <span style="color:green;">✓</span> | 01 sec         | 6/11/12 11:32 AM | 6/11/12 11:32 AM | SQL  | default         | Low      | 3285...    |                                    |
| 5443... | select * from reports_table;           | beehive      | <span style="color:green;">✓</span> | 01 sec         | 6/11/12 10:40 AM | 6/11/12 10:40 AM | SQL  | BI Tools        | Medium   | 3242...    |                                    |
| 8195... | select * from reports_table;           | db_superuser | <span style="color:green;">✓</span> | 01 ms          | 6/11/12 10:39 AM | 6/11/12 10:39 AM | SQL  | AdminOps        | High     | 5471...    |                                    |
| 2046... | create table reports_table (id int,... | db_superuser | <span style="color:green;">✓</span> | 01 sec         | 6/11/12 10:39 AM | 6/11/12 10:39 AM | SQL  | AdminOps        | High     | 5471...    |                                    |
| 3946... | create tab...                          | db_superuser | <span style="color:red;">✗</span>   | 01 sec         | 6/11/12 10:39 AM | 6/11/12 10:39 AM | SQL  | default         | Low      | 5471...    |                                    |
| 8450... | create database reports;               | db_superuser | <span style="color:green;">✓</span> | 02 sec         | 6/11/12 10:38 AM | 6/11/12 10:38 AM | SQL  | AdminOps        | High     | 1938...    |                                    |

You cannot force re-prioritization of a running query, but you can write workload rules that will re-prioritize running and queued jobs. See the variable, *stmtElapsedTime* in “[Workload Policy Predicate Attributes](#)” on page 282.

## Common Mistakes

In this section we describe some of the most common mistakes related to workload management configuration.

## No Default Workload Policy

As discussed above, the system requires at least one default workload policy to be defined if any workload policies are defined (i.e., the nc\_qos\_workload table is not empty), so remember to add one.

## Incorrect Default Workload Evaluation Order

The default workload policy must be last in the execution order. If it were first in the execution order, every incoming query would be mapped to the default workload policy, because it would be evaluated first and it always returns 'true'.

## Overlapping Workloads

It is possible for a single statement to match multiple workloads, so sometimes the mapping may not happen as you expect (and still be correct). To ensure that mapping is working as intended, you can use the nc\_qos\_active\_workloads view to see the ordered list of active workloads. (See [“Active Workloads” on page 238](#) for more details.) .

## Invalid Predicates

The type of each QoS context attribute defines the expressions and operators that are allowed by the system. For instance, given that userName is of type VARCHAR, an expression such as `userName like 'daniel%'` is valid and should be accepted, while something like `userName < 100` will be rejected by the system with a message as shown below.

```
beehive=> insert into nc_qos_workload values (10, 'newWorkload', 't',
'userName < 100', 'defaultClass');

ERROR: Predicate error: operator does not exist: character varying <
integer (SQLSTATE: 42883)
beehive=>
```

## Improper Quoting

The predicate column in the nc\_qos\_workload table is of type VARCHAR, so you need to properly quote constants when inserting new entries into the table. For example, note how the given username is quoted below:

```
beehive=> insert into nc_qos_workload values (10, 'newWorkload', 't',
'userName='jsmith'', 'defaultClass');

INSERT 1
```

## Case Sensitivity

Values of type VARCHAR are case sensitive! Keep this in mind when defining predicates that use attributes of this type. For example, although the predicate `stmtType='alter'` will be accepted as a valid predicate, (that is, it's a valid expression using a VARCHAR attribute) it will *not* match any ALTER . . . SQL statements because Aster Database recognizes each operation only by the Aster Database constant used to represent it, which in this case is “Alter” with a capital “A” instead of the all-lowercase “alter”. Pay close attention to the case of the allowed predicate arguments listed in [“Workload policy predicate attributes” on page 283](#).

## Taking Roles into Account

When using workload management, remember that both roles and workload assignments play a part in determining which workload is assigned to a user. The 'db\_admin' role is a special case which has visibility to all database objects. Because of this, for any user with the 'db\_admin' role, the 'myrole' = ANY(roles) condition will evaluate as positive (where 'myrole' is a valid role in the database). This also applies for 'amc\_admin', and any other role which is in an administrative role.

Be aware that users with an administrative role (which includes 'db\_superuser' as a member of the 'db\_admin' group) have implicit access to all database objects and that any workload classification rules should take special care to account for this case. This means that any user with an administrative role has the ability to act as any role, can access all database objects, and includes implicit roles, not just the expected ones.

# Best Practices

- 1 When approaching WLM for the first time, you should make a list of the common workloads your own system processes. Write down the time of day when they are typically run, the expected service level (SLA), the priority, and the importance to your organization (revenue, customer satisfaction, etc.) This will help you in determining how to allocate your system resources among the different jobs to be processed in a typical day by your Aster Database cluster.
- 2 Create workload policies that are as simple as possible while minimizing the number of false positives and false negatives. For instance, if all bulk loads into a system are executed as user 'etl' and this user is used only for this purpose, a workload policy definition that uses "userName= 'etl'" will be more effective than one that matches all COPY statements, for instance, as not all COPY statements are executed as part of bulk load operations. While Aster Database provides a comprehensive set of QoS attributes to use in workload policy definitions, it is useful to think about the smallest subset that achieves the desired effect.
- 3 It may not be necessary to use all priority levels. There is a huge difference (an order of magnitude) in terms of resource allocation between priority levels (for example, between "priority 3, weight 1" and "priority 2, weight 100"), so you may not need to use all available priority levels. Unless you have a complex system and tens of workload policy definitions, it may not even be necessary to use more than one or two priority levels.
- 4 As mentioned in, [Taking Roles into Account \(page 288\)](#), users with the 'db\_admin' role (which includes 'db\_superuser' as a member of the 'db\_admin' group) have implicit access to all database objects and that any workload classification rules should take special care to account for this.
- 5 You may wish to test your own WLM settings. When testing WLM settings, first run the queries and time them before setting up the service classes and workload policies. You may want to run them several times to verify that the timings are correct. Then set up your service classes and workload policies, save and apply them, and perform a soft restart of the cluster to flush the cache before running the queries again to collect WLM timings.

## Troubleshooting

The following steps are recommended when troubleshooting WLM:

- 1 You can use log files to help troubleshoot problems with WLM. WLM uses the log file `/primary/logs/QosSlaveExec.log`
- 2 To view information on queries canceled by WLM, see [Setting Up Workload Rules in the AMC \(page 276\)](#).
- 3 To further improve diagnosability, several statistics are provided on a per-service class basis. See [Workload Management statistics tables \(page 240\)](#).

## CHAPTER 6 Export and Load Tools

Aster Database provides a number of tools for exporting and loading data. These include:

Aster Clients, which are documented in the *Aster Client Guide*:

- Aster Export
- Aster Loader Tool

Connectors for loading data from other components of the Teradata Unified Data Architecture<sup>TM</sup>:

- [Teradata-Aster Database Connector](#)
- [SQL-H: The Hadoop/HCatalog Connector](#)



## CHAPTER 7 Teradata-Aster Database Connector

### Introduction

The Teradata-Aster Database Connector (referred to in this document as “Teradata Connector” or simply “Connector”) is a high-performance, bidirectional connector to copy data between the Teradata database and the Aster Database. The Connector takes advantage of parallel processing on both databases to enable fast loading of large amounts of data. To do this, Aster Database creates multiple sessions with Teradata to achieve parallelism during export and load operations.

The Connector consists of two SQL-MR functions called “load\_from\_teradata” and “load\_to\_teradata”. These two functions use the parallel version of the Teradata Parallel Transport (TPT) API to move data between the two systems. The Connector uses Teradata FastLoad and FastExport to do the loading and exporting of data on the Teradata side.

This document contains the following sections:

- [Teradata-Aster Database Configuration Overview](#)
- [Hardware Best Practices](#)
- [Copying Data from Teradata to Aster Database](#)
- [Copying Data from Aster Database to Teradata](#)
- [Connector Argument Clauses](#)
- [Datatype Support](#)
- [AMC Support](#)
- [Performance Tips and Best Practices](#)
- [Troubleshooting](#)
- [Limitations](#)

# Teradata-Aster Database Configuration Overview

## Prerequisites

The Teradata Connector has the following prerequisites:

- Teradata version 12, 13, 13.10 or 14.0
- Network connectivity between Aster Database and Teradata for all nodes

## Configure networking

On the Teradata side, you need connectivity for all nodes and an assigned database name for any Teradata database(s) you will be accessing through the Connector. All the other network configurations in this section apply to Aster Database only.

Because the Teradata Import and Export operations are executed in parallel across Aster Database, every node in Aster Database will need to be able to access the source or target Teradata database(s) by name using DNS. Depending on the network configuration in use, this may require that the `/etc/hosts` and/or the `/etc/resolv.conf` files on each Aster Database node be edited to include the necessary entries to access these gateways. It is recommended that you manage these configurations centrally using the AMC, as described in the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*. You will make the settings once in the AMC, and they will be copied to all Aster Database nodes automatically.

The Teradata TPT client uses DNS to discover gateways to the Teradata database. Teradata calls these gateways “Communication Processors” or “cops”. Each Teradata database is given a database name, and all the cops have DNS names which use a very specific naming convention (e.g. `dbnamecop1`, `dbnamecop2`, ..., `dbnamecopn`).

The TPT client relies on the cop naming/numbering convention described above. When it is looking for database connections, it starts by appending “`cop1`” to the database name, followed by “`cop2`”, “`cop3`”, ... and “`copn`” until a Host Not Found exception is thrown. Because of this, when making host entries for Teradata nodes, when you enter the alias, you must include “`cop#`” at the end (e.g., if you will execute “`... load_from_teradata( ... TDPID('dbc') ... )`”, then enter a name like “`dbccop1`” as the alias.)

## Hardware Best Practices

### Building out the correct hardware configuration

#### Overview

The Teradata Connector requires that Aster Database and Teradata have access to each other over a network. Depending on expected usage and workload, the design of this network should follow some best practices, which are outlined below.

In summary, the network configuration can take these forms:

- 1 Gigabit Ethernet (GbE) networking using the existing LAN
- 1 GbE private network between Teradata and Aster Database
- 10 GbE private network between Teradata and Aster Database

### **Aster Database node ethernet adapters**

The Aster Database side of the system needs to enable all of the workers to connect to at least one Teradata Gateway (cop). On appliance 2, the Aster Database worker nodes typically have a 10 GbE adapter for communication between nodes within the cluster. These would be connected by a 10GbE switch. If there are additional 1GbE or 10GbE ports available on each node, these can be used to connect to Teradata using either the #1 or #2 configuration outlined above.

The preferred method for connecting Aster Database to Teradata is to purchase an extra 10 GbE Ethernet adapter for installation in each of the Aster Database workers nodes. Make sure to purchase the correct Ethernet Adapter for your environment (copper or optical).

On the Appliance, while you may be able to use the 10Gb node interconnect switch or the 1Gb Server Management switch that is in use for the Aster Database intra-cluster network with the existing adapters, it's not the recommended procedure. A separate Aster Database to Teradata connector network is preferred.

### **Teradata ethernet adapters**

On the Teradata side, you'll also need client Ethernet ports to use to connect to Aster Database. The performance of the data movement will depend on the effective bottleneck. The number of connections, and the speed of those connections, within the Teradata system is a big contributing factor, but other factors that need to be considered may be outside of the Teradata system. The best approach would be to review the load architecture to ensure there are no obvious bottlenecks (slow Client servers, etc).

In addition to the client Ethernet performance, be sure to consider the BYNET bandwidth of the Teradata system. This may become the limiting factor when data is received by nodes that don't own the AMP where the data resides. In this case, there is also a second hop needed for row re-distribution, that will decrease effective performance.

### **Ethernet switch needs**

Since the Aster Database and Teradata systems are disparate systems, and each can be configured in many ways, there are no fixed configuration rules for the network switches. The Appliance cabinet does not have any available space for additional switches.

If you are running on the Appliance, Teradata does have a plethora of switch options that can be used for smaller installations, but for larger installations you may need to rely on either your existing network infrastructure or on procuring the switch outside of the Teradata sales channel.

The switch will need to accommodate enough ports for *all* of the Aster Database worker ports *plus* however many Teradata nodes will be configured as Gateways. Remember that this might be a multiple of the number of Aster Database workers and/or Teradata AMPs, if more than

one port is used in each node for performance reasons. You may also want to build in redundancy to the switch infrastructure for high availability reasons.

## Configuring for performance

Once the desired network bandwidth between the systems is determined, the following guidelines will help you determine how many nodes on the Teradata side need to have Gateways.

### Maximum performance

Maximum performance is determined by how many connections Aster Database has to Teradata nodes. If fewer than all Teradata nodes are connected to Aster, peak performance will be limited. Conversely, if all Teradata nodes are connected to Aster, peak performance potential is greater.

So, if you have the infrastructure in place to support connecting all of the Teradata nodes to Aster Database, that will yield the best performance. However, if you can not easily put the infrastructure in place, you'll need to perform a cost/benefit analysis to see if it's worth the investment and effort to configure the network for maximum performance.

### Number of Teradata AMPs and Aster Database vworkers

Using the Teradata Connector is easiest when the number of Teradata AMPs matches the number of Aster Database vworkers. If this is not the case, you will need to use some additional arguments as shown in the example “[Using load\\_to\\_teradata when number of vworkers exceeds number of AMPs](#)” on page 300.

### Ethernet performance

For the Appliance, Teradata provides two main options for Ethernet connectivity – 1Gb Ethernet and 10Gb Ethernet.

- Teradata has a four port 1Gb Ethernet adapter for sale – this adapter is capable of running at full speed on all four ports.
- Teradata’s Dual 10Gb Ethernet adapter is often limited by the PCIe slot, so if considering this option, please consult the documentation on the server to understand how much actual performance should be expected.

### BYNET performance

When sizing a system, you'll want to know the variation of BYNET installed on the Teradata side. BYNET performance scales per node.

- 1Gb BYNET maxes out at about 150MB/second per node.
- BYNETv4 maxes out at about 330MB/second per node.

The total performance you can expect on the Teradata side is BYNET-performance multiplied by the number of nodes with Gateways.

## Maximum gateway sessions

The per Gateway connection limit is 600 sessions. If the implementation requires more than 600 sessions, the way around this is to define more Gateways in /etc/hosts (or DNS). The session count is defined by the Client, with a maximum session count equal to the number of AMPs.

The recommended number of sessions to use is an even multiple of the node count. For example, 4 to 8 sessions per node is a good starting point.

If more sessions are needed and there aren't enough nodes to configure a gateway on an additional node, you can assign a second gateway to a given node as long as a separate IP address is used.

# Copying Data from Teradata to Aster Database

The “load\_from\_teradata” SQL-MR function copies data from a Teradata table to an Aster Database table. The function is invoked on Aster Database. A SELECT statement is supplied in the QUERY clause of the function, to specify the data to be loaded.

## Syntax of load\_from\_teradata

Here is the full syntax for load\_from\_teradata:

```
load_from_teradata(
    ON mr_driver
    TDPIID('tdpid')
    QUERY('query')
    [USERNAME('username')]
    [PASSWORD('password')]
    [LOGON_MECHANISM('TD2' | 'LDAP')]
    [LOGON_DATA('mechanism-specific logon data')]
    [ACCOUNT_ID('account-id')]
    [TRACE_LEVEL('trace-level')]
    [NUM_INSTANCES('instance-count')]
    [PRESERVE_COLUMN_CASE('YES|NO')]
    [MAX_SESSIONS('max-sessions-number')]
    [QUERY_TIMEOUT('timeout_in_seconds')]
    [SPOOLMODE('NoSpool|Spool')]
    [SKIP_ERROR_RECORDS('yes' | 'no')]
) ;
```

For an explanation of what the clauses mean, and what parameters should be passed, see [“Connector Argument Clauses” on page 301](#).

## Using load\_from\_teradata

The load\_from\_teradata SQL-MR function must be invoked on a partitioned fact table in Aster Database. This is usually done by using a dummy table, which will be referred to as “mr\_driver”, created as follows:

```
CREATE TABLE mr_driver(
    c1 INT)
```

```
DISTRIBUTE BY HASH (c1)
;
```

For larger result sets, it's a good idea to capture the output from `load_from_teradata` to a table in Aster Database, to avoid the need to repeat the load if a query must be run again. There are two ways to do this:

1 Create a table in Aster Database to receive the data:

You can first create a table in Aster Database to receive the data from Teradata. The columns in the Aster Database table must exactly match the names and structure of the columns in the Teradata table(s) from which the data will be copied. That is, it must have the same column names, with the same or a compatible datatype in each column. But of course, the Aster Database table does not need to include any columns that will be omitted from the query that selects the data to be loaded. If the datatypes for a column do not match, an implicit datatype conversion will be performed. For more information on datatypes, see [“Datatype Support” on page 305](#). After creating the table, you may then select into it using the `load_from_teradata` function.

2 Use `CREATE TABLE...AS...SELECT` (CTAS):

Using CTAS will ensure that the data structure of the Aster Database table is compatible with that of the Teradata table(s) from which the data will be copied, for all the columns that are included in the `QUERY` clause. Because the connector creates the schema and does datatype conversion automatically, this is an excellent way to easily copy data, without having to analyze the schema and do datatype conversions manually.

Here is how CTAS works with the `load_from_teradata` function: During planning, the `load_from_teradata` function describes the Teradata query to determine the schema of the result set. This schema will then be converted into a compatible Aster Database schema, and the SQL-MR function will report that as the output schema of the `load_from_teradata` SQL-MR function. Datatype conversion is performed implicitly.

To see what implicit conversions will be applied, see [“Datatype Support” on page 305](#).

Before attempting a CTAS within `load_from_teradata`, you may want to ensure that there are no columns for which there is no corresponding datatype in Aster Database. To do this, see [“Unsupported Datatypes” on page 309](#).

This sample query will create a new table in Aster Database called `aster_target_table` which contains the same rows as the result of the Teradata query `'SELECT * FROM td_src_table'`.

```
CREATE TABLE aster_target_table DISTRIBUTE BY HASH(td_partition_col) AS
SELECT
*
FROM load_from_teradata(
    ON mr_driver -- empty Aster mr_driver table
    TDPID('dbc') -- Teradata hostname
    USERNAME('vmuser')
    PASSWORD('vmpass')
    QUERY('SELECT * FROM td_src_table'))
```

) ;



**Tip!** When using a CTAS (CREATE TABLE ... AS) or when specifying PRESERVE\_COLUMN\_CASE ('YES'), the Connector will preserve the case of Teradata column names. Remember that in Aster Database, names with any uppercase letters must be quoted. Therefore, you would need to refer to the column COL1 as "COL1" in any subsequent SQL statements. See the next section for details.

## Handling Upper Case Letters in Imported Table Column Names

Both Teradata and Aster Database provide case-sensitive treatment of table and column names, but the two systems differ in how they treat mixed-case names in a query. In both systems, you may create, for example, two distinct tables called "mytable" and "MyTable." In Aster Database, the way you query a table or column with an uppercase or mixed case name is different than in Teradata. In Aster Database, you must surround any uppercase or mixed case name in quotation marks (you use *double quotes*, by default) or it will be treated as if you had typed it in all lowercase characters. In Teradata, no such quoting is needed.

This difference can create confusion when you retrieve tables and columns from Teradata and use them in an Aster Database query, because if you specify PRESERVE\_COLUMN\_CASE ('YES') the case of the Teradata table and column names is preserved. As a result, if a retrieved table or column name contains any uppercase characters, you must double-quote that name in your Aster Database query.

To avoid confusion, you should make a habit of enclosing in double quotes all table and column names that you retrieve using `load_from_teradata`. For example, note the use of double quotes in this example:

```
SELECT "user_stats"."URL" AS dest_url, "td_query"."HEIGHT_I" AS ter_height
  FROM
    user_stats,
    load_from_teradata (
      ON
        EMPTY tdpid('teradata.ggw.cda')
        USERNAME('sjsonze')
        PASSWORD('L00pyd00')
        LOGON_MECHANISM('ldap')
        QUERY(
          'SELECT HEIGHT_I, PITCH_A, status_code '
          'FROM Saturn.Item_Heights ')
      ) td_query
 WHERE user_stats.user_height = td_query.ter_height;
```

## Copying Data from Aster Database to Teradata

The "load\_to\_teradata" SQL-MR function copies data from Aster Database to Teradata. The function is invoked on Aster Database. A SELECT statement is supplied in the ON clause, to specify the data to be loaded. The function outputs information about the data copied and any errors.

## Syntax of load\_to\_teradata

Here is the full syntax for load\_to\_teradata:

```
load_to_teradata(
    ON ('source query')
    TDPID('tdpid')
    TARGET_TABLE('fully-qualified table name')
    [ERROR_TABLES('error table'[, 'unique constraint violation table'])]
    [LOG_TABLE('table name')]
    [USERNAME('username')]
    [PASSWORD('password')]
    [LOGON_MECHANISM('TD2' | 'LDAP')]
    [LOGON_DATA('mechanism-specific logon data')]
    [ACCOUNT_ID('account-id')]
    [TRACE_LEVEL('trace-level')]
    [NUM_INSTANCES('instance-count')]
    [START_INSTANCE('start-instance')]
    [MAX_SESSIONS('max-sessions-number')]
    [QUERY_TIMEOUT('timeout_in_seconds')]
);
```

For an explanation of what the clauses mean, and what parameters should be passed, see [“Connector Argument Clauses” on page 301](#).

## Output of load\_to\_teradata

The load\_to\_teradata function has two outputs. For each output, one row is returned for each vworker instance on which the function was executed.

- `loaded_row_count`

The loaded\_row\_count output indicates the total number of rows that were loaded into the target Teradata table. Only one row will have the total row count, and the other rows will have a value of 0. If the connector succeeded in loading rows into Teradata, but failed to get statistics, this column will have the value of -1. Use `sum(loaded_row_count)` to obtain the total number of rows loaded.

The value returned is equal to (actual number of rows returned) modulo  $2^{32}$ . If the number of rows to be loaded is expected to be greater than  $2^{32}$  (4,294,967,295) or the row count is -1, please check the row count in the Teradata database by issuing a `SELECT COUNT(*)` on the target table.

- `error_row_count`

The error\_row\_count returns the number of rows in both of the Teradata error tables. To see a total number of errors, issue `sum(error_row_count)`.

## Using load\_to\_teradata

You must first create a table in Teradata to hold the data being loaded. This table must exist, be empty, and have a schema that's compatible with the data being exported from Aster Database. Note that because the table must be empty, you cannot make two consecutive load\_to\_teradata calls to the same target table. If you are making multiple consecutive calls to the function, you must use a different target table for each one. After the data is loaded into

the target tables, it can be consolidated into a single Teradata table in a separate SQL operation on Teradata.

If a datatype specified in the originating Aster Database schema does not match the datatype in the target Teradata table, implicit datatype conversion will be performed by *Teradata*. For Teradata, conversion rules are listed in the Teradata document “SQL Functions, Operators, Expressions, and Predicates” which may be found at <http://www.info.teradata.com/edownload.cfm?itemid=102320046>. After this table has been created, you can execute the `load_to_teradata` function.

The following code example includes a `SELECT` statement to access the output of the function and provide 1) a count of rows successfully loaded, and 2) a count of rows with errors. In the `ON` clause, there is a `SELECT` statement to indicate which rows are to be copied to Teradata.

```
SELECT
    sum_loaded_row_count,
    sum_error_row_count
FROM load_to_teradata(
    ON (SELECT * FROM aster_source_table)
    TDPID('dbc')
    USERNAME('vmuser')
    PASSWORD('vmpass')
    TARGET_TABLE('schema.td_target_table')
) ;
```



**Tip!** If a `load_to_teradata` operation is cancelled or aborted, the target table in Teradata may become locked. The Connector cannot load data into a locked table or a partially loaded table.

The following steps must be taken before executing the query again:

1 The locked target table and any error tables must be unlocked and then dropped.

2 The target table and any error tables must be recreated.

For the procedure for handling locked tables in Teradata, please refer to the Teradata FastLoad manual, found at <http://www.info.teradata.com/edownload.cfm?itemid=101320006>.

## Using `load_to_teradata` when number of vworkers exceeds number of AMPs

This example shows how to load data into Teradata when the number of Aster Database virtual workers exceeds the number of AMPs in Teradata. To find out the number of AMPs in Teradata:

To find out the number of Teradata AMPs, do the following:

- 1 Login to the Teradata database using the Teradata client, `bteq`.

```
$bteq
.logon dbc/UserID
password:
```

```
Select Count( distinct vproc) from dbc.AmpUsage;
*** Query completed. One row found. One column returned.
*** Total elapsed time was 1 second.
```

```
Count(Distinct(Vproc))
-----
2
quit;
```

- 2 The output displays the number of Teradata AMPs. In the above example the number of Teradata AMPs is 2.

The load\_to\_teradata function is invoked as many times as needed, to balance the data transfer between the Aster Database vworkers and the Teradata AMPs. Note that as a best practice, the functions should be run within the context of a single transaction to maintain data integrity, as shown in the example.

Perform the following steps when the number of vworkers exceeds the number of AMPs:

- 1 Determine the number of load\_to\_teradata() calls to make, using the following formula:

$$\text{CallCount} = \text{vworkerCount} / \text{AMPCount}$$

If the result is not an integer, round up to the nearest integer.

- 2 In Teradata, create the required number of target tables (e.g. if CallCount is 4, then create 4 target tables with names like target\_table1, target\_table2, target\_table3, etc.)
- 3 Create and execute the load\_to\_teradata() statements, as follows:

```
BEGIN;

SELECT (*)  
  FROM load_to_teradata (ON (SELECT * FROM aster_source_table)  
    TARGET_TABLE ('schema.target_table1')  
    TDPIP('...') USERNAME('...') PASSWORD('...')  
    START_INSTANCE('0')  
    NUM_INSTANCES('6'));  
  
SELECT (*)  
  FROM load_to_teradata (ON (SELECT * FROM aster_source_table)  
    TARGET_TABLE ('schema.target_table2')  
    TDPIP('...') USERNAME('...') PASSWORD('...')  
    START_INSTANCE('6')  
    NUM_INSTANCES('6'));  
  
...;  
  
END;
```

- 4 On the Teradata server, combine the data into a single table using SQL:

```
INSERT INTO target_table SELECT * FROM target_table1;  
INSERT INTO target_table SELECT * FROM target_table2;  
INSERT INTO target_table SELECT * FROM target_table3;  
...
```

## Connector Argument Clauses

### Common Clauses

The following clauses occur in both the load\_from\_teradata and load\_to\_teradata functions:

- TDPIP('tdpid')

The TDPID is the identifier for the Teradata database. It specifies the hostname, IP address, or database name of the Teradata Database to which the Connector should connect.

This clause is required for both load\_from\_teradata and load\_to\_teradata.

- `USERNAME ('username')`
- `PASSWORD ('password')`
- `LOGON_MECHANISM ('TD2' | 'LDAP')`
- `LOGON_DATA ('mechanism-specific logon data')`

These four clauses specify logon credentials for the Teradata database.

A different collection of these clauses, depending on the authentication mechanism, is required for both load\_from\_teradata and load\_to\_teradata.

The TD2 parameter selects the Teradata Connector authentication mechanism. For TD2, the USERNAME and PASSWORD clauses are required. As TD2 is the default authentication mechanism, the LOGON\_MECHANISM clause is optional when using TD2.

For LDAP, the LOGON\_MECHANISM and LOGON\_DATA clauses are required. See the Teradata Database Security Administration manual for full details on the required value of the LOGON\_DATA clause.

More information regarding logon credentials in Teradata is available at: <http://developer.teradata.com/connectivity/articles/logging-on-teradata-database-sessions-via-bteq>



**Tip!** For security, the values of the PASSWORD and LOGON\_DATA clauses will be obfuscated and will not appear in the Aster Management Console (AMC) or in Aster Database statistics tables.

- 
- `ACCOUNT_ID ('account-id')`

Specifies the Teradata accounting group with which to associate the work done on the Teradata database by the Connector.

This clause is optional.

- `TRACE_LEVEL ('trace-level')`

Specifies the TPT trace level to use.

This clause is optional and should only be used when instructed to do so by Teradata Global Technical Support (GTS).

- `MAX_SESSIONS ('max-session-number')`

Specifies the limit on the maximum number of sessions that can be established with Teradata AMPs.

This value is specified as an integer, and it must be greater than or equal to the number of TPT instances used.

The argument is optional and if not supplied, is not used at connection time.

- `QUERY_TIMEOUT ('timeout_in_seconds')`

Specifies the response time allowed in seconds.

The argument is optional and if not supplied, the default timeout is 30 minutes.

## load\_from\_teradata Clauses

The following clauses occur in only the load\_from\_teradata function:

- `ON mr_driver`

The load\_from\_teradata function requires a pre-existing `mr_driver` table to be present in Aster Database. This table is referenced in the `ON` clause of the `load_from_teradata` function. The structure of the `mr_driver` table is not important. It is simply used to fulfill the syntax requirements of the SQL-MR function.

- `QUERY ('query')`

The query is a `SELECT` statement, which is supplied within the function to specify the data to be loaded. Specifies the query to be executed on the Teradata database, the result of which will be returned by `load_from_teradata`. It is a good practice to use dollar signs for quoting around your query clause (e.g. `QUERY ( $SELECT * FROM emp$ )`).

This clause is required for `load_from_teradata`.

- `NUM_INSTANCES ('instance-count')`

Specifies the number of instances that should participate in the parallel TPT import into Aster Database. The value of this clause can exceed neither the number of virtual workers (vworkers) in Aster Database nor the number of AMPs (Access Module Processors) in Teradata.

This clause is optional. By default, the number of instances will be equal to the number of Aster Database virtual workers. See “[Interaction Between MAX\\_SESSIONS and NUM\\_INSTANCES](#)” on page 312.

- `PRESERVE_COLUMN_CASE`

Specifying 'YES' for this clause will result in keeping the existing case for column names when transferring to Aster Database. Specifying 'NO' for this clause will result in changing all table names to lowercase.

This clause is optional. The default value is 'NO'.



**Tip!** The optional `PRESERVE_COLUMN_CASE` clause was added to the syntax for `load_from_teradata` beginning in Aster Database version 5.0. The default value is 'NO', which results in all column names being loaded to Aster Database as all lowercase. In earlier releases, the Teradata column names were brought into Aster Database using `load_from_teradata` as supplied by the API (preserving case). Specifying 'YES' for this clause will result in the same behavior as in older versions (i.e. keeping the existing case for Teradata column names when transferring to Aster Database.)

- 
- `SPOOLMODE ('NoSpool|Spool')`

Specifies setting "Spool" or "NoSpool" for the Teradata attribute "TD\_SPOOLMODE".

The argument is optional, and if not supplied, the default value "NoSpool" is used. For details and information on when to use this argument, see “[NoSpool Mode Limitations and Functionality](#)” on page 314.

- `SKIP_ERROR_RECORDS ('yes' | 'no')`

Use this argument to specify what to do if errors are encountered when loading data from Teradata. When using `load_from_teradata`, the connector can encounter an error when parsing a row with data that is valid in Teradata but is not supported in Aster Database. If set to 'yes', the connector skips rows with parsing errors and continues loading. If set to 'no' the load is aborted upon encountering an error, and no rows are loaded to Aster Database. The default is 'no'. This clause is optional.

## load\_to\_teradata Clauses

The following clauses occur in only the `load_to_teradata` function:

- `ON source_query`

The source query is a `SELECT` statement, supplied within the function to specify the data to be loaded into Teradata.

- `TARGET_TABLE ('fully-qualified table name')`

Specifies the Teradata table into which the `load_to_teradata` function should load data. This table must exist, be empty, and have a schema that's compatible with the data being exported from Aster Database.

This clause is required for `load_to_teradata`.



**Notice!** When creating the Teradata target table, be aware of how Teradata FastLoad handles duplicate rows. By default, Teradata FastLoad does *not* load duplicate rows from the data source to the Teradata Database. (A duplicate row is one in which every field contains the exact same data as the fields of an existing row.). Duplicate rows *can* be inserted if the Teradata table is created with the NoPI option, so be sure to use it if you will be loading data which could include duplicate rows that you want to retain.

- 
- `ERROR_TABLES ('error table' [, 'unique constraint violation table'])`

Specifies the error tables to use for the load into Teradata. These tables must not exist in Teradata when the export to Teradata begins.

The clause can have either one or two values. The first value corresponds to the error table (i.e. "first error table" in the Teradata documentation.) The second value corresponds to the unique constraint violation table (i.e. "second errors table" in the Teradata documentation.)

The error table contains records rejected during the Acquisition phase of the export due to data conversion errors, constraint violations, or AMP configuration changes. The default name of this table is <TARGET\_TABLE>\_ET.

The unique constraint violation table contains records rejected during the Acquisition phase of the export due to violations of unique primary index constraints. The default name of this table is <TARGET\_TABLE>\_UV.

This clause is optional. If it does not exist, the default tables names are used.

- `LOG_TABLE ('table name')`

Specifies the name of the Teradata restart log table to use. In Teradata, the FastExport utility writes operational status information to the restart log table, so it can be restored after a system restart operation. This name is optionally schema-qualified; if the schema is

not specified, it defaults to the user's default logon database. If you receive the error “[Not Enough PERM Space Error](#)” on page 318, however, you should use a fully qualified table name for the log table. The table may already exist, but it is not required to be pre-existing. This clause is optional. If not specified, the log table defaults to “logtable” in the user's default logon database.

- `START_INSTANCE ('start-instance')`

This clause should be used only if the number of virtual workers in Aster Database exceeds the number of AMPs in Teradata. If START\_INSTANCE is used, you must also use NUM\_INSTANCES to specify the number of instances to use.

Specifies the beginning number of Aster Database virtual worker instances that should participate in the parallel load into Teradata.

This clause is optional. The default value of this parameter is 0. The maximum value of this parameter is one less than the number of Aster Database virtual workers.

For an example, see “[Using load\\_to\\_teradata when number of vworkers exceeds number of AMPs](#)” on page 300.

- `NUM_INSTANCES ('instance-count')`

This clause must be used if the number of vworkers in Aster Database exceeds the number of AMPs in Teradata and/or if START\_INSTANCE is used. If neither of these is the case, it should not be used. You must set this argument to a value equal to or lower than the total number of vworkers.

Specifies the number of Aster Database virtual worker instances that should participate in the parallel load into Teradata. The value of this clause can exceed neither the number of vworkers in Aster Database nor the number of AMPs in Teradata.

This clause is optional, unless START\_INSTANCE is used and/or the number of vworkers is greater than the number of AMPs in Teradata, in which case it is required. By default, the number of instances will be equal to the number of Aster Database virtual workers.

See these related topics:

[“Interaction Between MAX\\_SESSIONS and NUM\\_INSTANCES” on page 312](#)

[“Using load\\_to\\_teradata when number of vworkers exceeds number of AMPs” on page 300](#)

## Datatype Support

The following sections describe datatype support and provide some notes on datatype conversion. You may also need to consult the Teradata documentation for your database version.

Whenever transferring data from one database to another, ensure that both the source table and the destination table have the same structure in columns, constraints, datatypes, etc. It is also acceptable if they have corresponding datatypes or datatypes for which an implicit conversion may be performed. If the datatypes do not match exactly, there may be conversion rules and behavior that you need to be aware of. The following section addresses these:

## Aster Database to Teradata datatype conversions

When copying from Aster Database to Teradata, you may need to consult the Teradata documentation for your database version, for datatype conversion rules. For version 13.10, the documentation may be found here: <http://www.info.teradata.com/edownload.cfm?itemid=102320046>. The following notes are supplemental to the Teradata documentation:

### Converting from numeric datatype in Aster Database to date datatype in Teradata

Teradata allows some datatype conversions that are different from those in Aster Database. For example, inside the Teradata server, the SMALLINT 127 can be converted to the DATE "1900-01-27". The same types of conversions can occur when data is transferred from Aster Database tables to Teradata tables.

If a datatype specified in the originating schema does not match the datatype in the target table, implicit datatype conversion will be performed.

For Teradata, conversion rules are listed in the document "SQL Functions, Operators, Expressions, and Predicates" which can be found here: <http://www.info.teradata.com/edownload.cfm?itemid=102320046>.

The following excerpt is from page 833 of this document, for Teradata Release 13.10 B035-1145-109A:

Translation of Numbers to Dates: Although not recommended, you can explicitly convert numbers to dates. Teradata Database stores each DATE value as a four-byte integer using the following formula:

$(\text{year} - 1900) * 10000 + (\text{month} * 100) + \text{day}$

For example, December 31, 1985 would be stored as the integer 851231; July 4, 1776 stored as -1239296; and March 30, 2041 stored as 1410330.

The following values demonstrate how numeric dates are interpreted when inserted into a column. Note the translation of the third date, which was probably intended to be 1990-12-01.

These numeric values:

901201 1001201 19901201

Translate to these date values:

1990-12-01 2000-12-01 3890-12-01

Notice that this formula best fits two-digit dates in the 1900s. Because of the difficulty of using this format outside of the 1900s, dates are best specified as ANSI date literals instead.

### Converting from time datatype in Aster Database to char(N) datatype in Teradata where N is too short

The Teradata documentation has a "Time-to-Character Conversion" section, which says:

Time-to-Character Conversion: "When converting TIME to CHAR(N) or VARCHAR(N), then N must be equal to or greater than the length of the TIME value as represented by a character string literal."

Note that if N is less than the length of the TIME value as represented by a character string literal, the time is truncated. There is no error message if this occurs.

### **Converting from Date datatype in Aster Database to numeric datatype in Teradata**

This is documented in the section titled "DATE-to-Numeric Conversion" in the Teradata document "SQL Functions, Operators, Expressions, and Predicates" which can be found here: <http://www.info.teradata.com/edownload.cfm?itemid=102320046>.

### **Converting from char datatype in Aster Database to byteint datatype in Teradata**

Teradata will perform an implicit datatype conversion when you pass data of type char from Aster Database to a Teradata field of type byteint. For example, number characters from Aster Database will be converted to byteint in Teradata automatically. It should be noted that during this conversion, an empty string, or string of blanks is treated as zero (0) by Teradata.

## **Teradata to Aster Database datatype conversions**

This section covers datatype conversion when copying from Teradata to Aster Database.

Table 7 - 1: Supported Datatypes

| Teradata Datatype | Aster Datatype | Notes                                                                                |
|-------------------|----------------|--------------------------------------------------------------------------------------|
| bigint            | Bigint         |                                                                                      |
| byte [ (n) ]      | bytea          | Exporting a bytea to Teradata will create a varbyte of maximum length (64000 bytes). |
| byteint           | smallint       |                                                                                      |
| char [ (n) ]      | char [ (n) ]   |                                                                                      |
| date              | date           |                                                                                      |

Table 7 - 1: Supported Datatypes (continued)

| Teradata Datatype                   | Aster Datatype                      | Notes                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| decimal [(s [, p])]                 | numeric(s, p)                       | An Aster Database numeric must have a scale and precision to be exportable to Teradata, due to the different meanings of missing precision and scale in the two systems.<br><br>To export an Aster Database column that has no declared precision, you must cast the numeric column to a datatype that has an explicit precision. For example: |
|                                     |                                     | <pre>SELECT * FROM load_to_teradata ( ON (SELECT cast(NUMERICCOL as decimal(15,2)) FROM mydb.mytable) TDPID('tdpid') USERNAME('username') PASSWORD('password') TARGET_TABLE('fully-qualified table name') LOG_TABLE('optionally non-qualified table name') );</pre>                                                                            |
| float                               | double<br>precision                 |                                                                                                                                                                                                                                                                                                                                                |
| integer                             | integer                             |                                                                                                                                                                                                                                                                                                                                                |
| smallint                            | smallint                            |                                                                                                                                                                                                                                                                                                                                                |
| time [without<br>time zone]         | time<br>[without<br>time zone]      |                                                                                                                                                                                                                                                                                                                                                |
| timestamp<br>[without time<br>zone] | timestamp<br>[without<br>time zone] |                                                                                                                                                                                                                                                                                                                                                |
| varbyte (n)                         | bytea                               | Exporting a bytea to Teradata will create a varbyte of maximum length (64000 bytes).                                                                                                                                                                                                                                                           |

Table 7 - 1: Supported Datatypes (continued)

| Teradata Datatype | Aster Datatype | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| varchar (n)       | varchar (n)    | <p>Note that while Aster Database supports arbitrary length varchar columns, Teradata does not. In addition, the Teradata TPT API supports a maximum of 32000 bytes in a varchar column during data transfer.</p> <p>Due to this mismatch in limitations, if the number of bytes exceeds 32000 bytes in a varchar column during transfer, load_to_teradata will fail with an error.</p> <p>Therefore, the Connector has a limit of 32,000 bytes maximum in any varchar column. For example:</p> <p>If the data has all ASCII characters, the column can have up to 32000 characters.</p> <p>If the data has multibyte UTF-8 characters, in the worst case, the column can have up to 32000/3 characters (assuming all 3-byte UTF 8 characters).</p> |

## Unsupported Datatypes

Table 7 - 2: Teradata Datatypes Not Supported in Aster Database

| Teradata Datatype | Workaround                                                                                                   |
|-------------------|--------------------------------------------------------------------------------------------------------------|
| blob              | Extract a substring of up to size 64000 from the blob and cast to a character datatype.                      |
| clob              | Extract a substring of up to size 64000 from the clob and cast to a character datatype.                      |
| interval [...]    | Cast the interval to a character datatype.                                                                   |
| period [...]      | Convert a single period datatype column into its begin and end bounds using the BEGIN() and END() functions. |

Table 7 - 3: Aster Database Datatypes Not Supported in Teradata

| Aster Datatype | Workaround                                                                                                                               |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| interval       | Cast to a varchar.                                                                                                                       |
| ip4            | Cast to a varchar.                                                                                                                       |
| ip4range       | Convert a single ip4range column to its lower and upper bounds using the LOWER() and UPPER() functions. Export these columns separately. |
| uuid           | Cast to varchar or numeric.                                                                                                              |

## Converting Multibyte Characters to Unicode During a Load

The load\_from\_teradata function does not support loading characters encoded using the Japanese character sets Kanji 1 or Kanji Jis. This limitation applies to columns of the datatype char, varchar and longvarchar.

However, you can load these characters if you do a conversion within your SQL call to load\_from\_teradata, similar to this example:

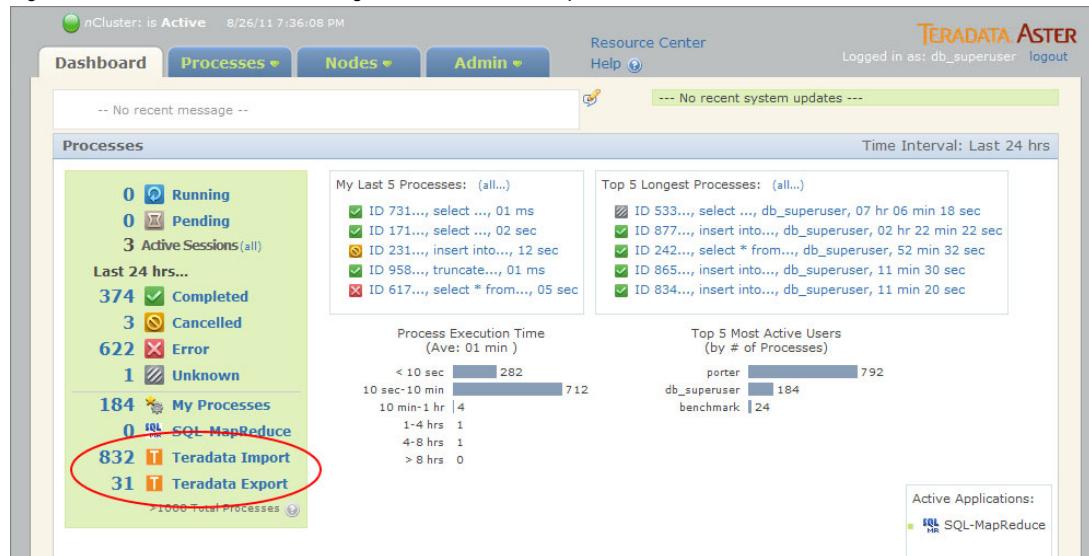
```
SELECT * from load_from_teradata(
    TDPID('...')

    ...
    QUERY ('SELECT TRANSLATE(<column-name>
        USING <charset:{kanjisjis,kanji1}>_to_unicode)
        AS <column-name> from <source-table>')
    )
    ...
);
```

## AMC Support

The AMC refers to Teradata - Aster Database Connector processes as statements of the type “Teradata Import” or “Teradata Export”. They are displayed as such on the AMC dashboard:

Figure 25: AMC dashboard showing Teradata Connector processes



When viewing processes in the AMC, you can specify “Teradata Import” or “Teradata Export” as the Statement Type in the Process Filter dropdown list:

Figure 26: AMC Process Filter dropdown list showing Teradata Connector processes

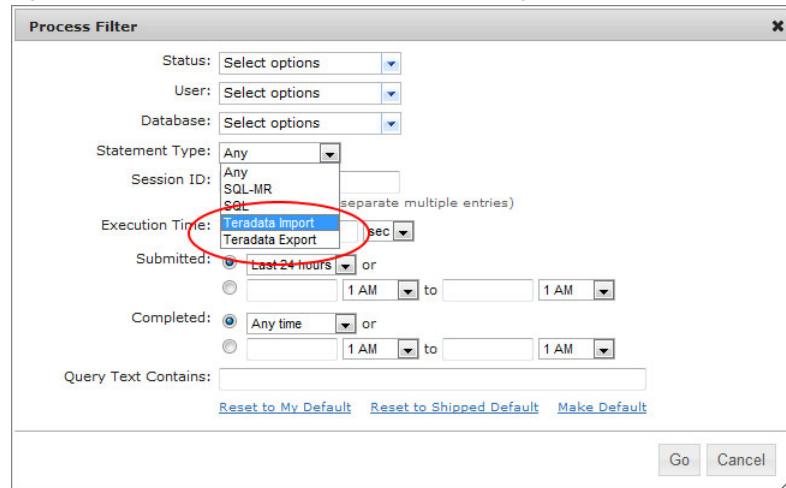


Figure 27: AMC Process list showing Teradata Import processes

| ID      | Statement      | User         | Status                                | Execution Time | Submit Time     | Completion Time | Type               | Workload Policy | Priority | Session ID | Cancel |
|---------|----------------|--------------|---------------------------------------|----------------|-----------------|-----------------|--------------------|-----------------|----------|------------|--------|
| 2316... | insert into... | db_superuser | <span style="color: orange;">S</span> | 12 sec         | 8/26/11 6:44 PM | 8/26/11 6:44 PM | Teradata PM Import | None            | None     | 3823...    |        |
| 1947... | insert into... | db_superuser | <span style="color: red;">X</span>    | 01 min 46 sec  | 8/26/11 6:43    | 8/26/11 6:44    | Teradata PM Import | None            | None     | 1165...    |        |
| 7360... | insert into... | db_superuser | <span style="color: green;">✓</span>  | 23 sec         | 8/26/11 6:41    | 8/26/11 6:41    | Teradata PM Import | None            | None     | 5575...    |        |
| 2690... | insert into... | db_superuser | <span style="color: red;">X</span>    | 04 sec         | 8/26/11 4:32 AM | 8/26/11 4:32 AM | Teradata AM Import | None            | None     | 6194...    |        |
| 5691... | insert into... | db_superuser | <span style="color: green;">✓</span>  | 13 sec         | 8/26/11 4:32 AM | 8/26/11 4:32 AM | Teradata AM Import | None            | None     | 2486...    |        |
| 1136... | insert into... | db_superuser | <span style="color: green;">✓</span>  | 02 min 04 sec  | 8/26/11 4:23 AM | 8/26/11 4:25    | Teradata AM Import | None            | None     | 2951...    |        |

## Performance Tips and Best Practices

- 1 Always ensure that there is enough disk space on the cluster to which the data will be copied, before beginning a data load operation.
- 2 Do not attempt to transfer data from an Aster Database column of variable length with no specified maximum (e.g. "VARCHAR") to a Teradata column with a specified maximum (e.g. "CHAR(30)" or "VARCHAR(30)"). This can cause unexpected behavior. If you attempt to do this, you may see an error like:

ERROR: Initiate() failed with status code 3798: A column or character expression is larger than the max size.

If this happens, and the Teradata tables remain locked, you may have to drop and recreate them before attempting to copy the data again.

- 3 There is a 64KB limit on the size of the records that can be transferred via the two functions, `load_from_teradata` and `load_to_teradata`.
- 4 You will find the Teradata documentation helpful, particularly the Teradata "Messages" manual, which provides an explanation and possible remedy for many error messages that come from Teradata. This manual makes it easy to search for error messages that have a numeric error code (e.g. the "2634" in the message "2634 Existing ERROR table(s) ..."). The Messages manual for Teradata version 13.10 can be downloaded here:

<http://www.info.teradata.com/edownload.cfm?itemid=102320024>

For other versions of Teradata, manuals can be downloaded by searching here:

<http://www.info.teradata.com/>

## Interaction Between MAX\_SESSIONS and NUM\_INSTANCES

For any `load_to_teradata()` or `load_from_teradata()` operation, Aster Database attempts to create multiple sessions to connect to Teradata, in order to achieve parallelism.

Teradata has the following controls that affect the number of sessions available for a particular request. The number of sessions available for Aster will be equal to the lesser of these two numbers:

- The number of Teradata AMPS : A maximum of one session is available per AMP.
- The number set by the Teradata Active System Management (TASM) rules: In Teradata 13.10 and later versions, TDWM can be used to throttle the number of sessions for a specific application. The Teradata server will select the number of utility sessions available according to Teradata Dynamic Workload Manager (TDWM) session rules. You can view this setting from Teradata Viewpoint or by executing the `tdwmdmp -a` utility.

The Teradata Connector uses the following parameters to set the number of vworkers that will connect to Teradata:

- MAX\_SESSIONS: The maximum number of sessions that can be used by Aster. The default setting is the maximum number of sessions available on Teradata.
- NUM\_INSTANCES: The number of Aster vworkers that will attempt to connect to Teradata. This parameter should equal to or less than the number of sessions available on Teradata for Aster connections. The default setting is the number of Aster vworkers.

Beginning in Aster Database 5.0.2, if the number of sessions offered by Teradata is less than the value of NUM\_INSTANCES, the Teradata Connector operation will fail with an error.

The following table describes the effect of assigning different values for the NUM\_INSTANCES and MAX\_SESSIONS parameters in the Teradata Connector:

Table 7 - 4: Effect of NUM\_INSTANCES and MAX\_SESSIONS parameters

| Item           | Value |     |     |     |
|----------------|-------|-----|-----|-----|
| Aster vworkers | 36    | 36  | 36  | 36  |
| Teradata AMPs  | 256   | 256 | 256 | 256 |

Table 7 - 4: Effect of NUM\_INSTANCES and MAX\_SESSIONS parameters

| Item                                    | Value                                                  |                                                |                                                |                                                      |
|-----------------------------------------|--------------------------------------------------------|------------------------------------------------|------------------------------------------------|------------------------------------------------------|
| TDWM Utility Session limit              | 50                                                     | 50                                             | 4                                              | 4                                                    |
| Effective session available on Teradata | 50                                                     | 50                                             | 4                                              | 4                                                    |
| MAX_SESSIONS                            | 72                                                     | 32                                             | 32                                             | 32                                                   |
| NUM_INSTANCES                           | Not specified*                                         | Not specified*                                 | 16                                             | 4                                                    |
| Result                                  | 50 sessions will be established from 36 Aster vworkers | Error: NUM_INSTANCES should be >= MAX_SESSIONS | Error: NUM_INSTANCES > effective session count | 4 sessions will be established from 4 Aster vworkers |

\* When NUM\_INSTANCES is not specified, the default is the number of Aster vworkers (36 in this example).

### When to use the NUM\_INSTANCES parameter

If the number of sessions available for Aster on the Teradata server is less than the number of vworkers, use the NUM\_INSTANCES parameter, set its value to be less than or equal to the number of available sessions. As you reduce this value, transfer time will increase because fewer vworkers will participate in the data transfer.

For example, if 360 GB of data is transferred using 36 vworker instances, each instance will handle about 10 GB of data. If the same amount of data is transferred using 4 vworker instances, each instance will handle about 90 GB data, which should take roughly nine times as long as in the first example.

### When to use the MAX\_SESSIONS parameter

If the Teradata server has a large number of AMPs, and does not limit the number of available sessions, Aster attempts to use all of the available sessions. This can consume a large number of resources on Teradata. In this case, MAX\_SESSIONS can be used to limit the number of sessions created between Aster and Teradata.

When TASM rules are configured to throttle the number of sessions allowed by the Teradata server, this parameter is not very effective.

On Teradata 13.0 or other systems where TASM is not configured, the MAX\_SESSIONS parameter can be used effectively to control the number of sessions between Aster and Teradata.

## NoSpool Mode Limitations and Functionality

The load\_from\_teradata function provides an option to control spooling behavior.

The syntax for the Teradata spool mode setting in the load\_from\_teradata connector is:

```
SPOOLMODE ('NoSpool | Spool')
```

where:

- Spool = Use Spool.
- NoSpool = Do not use Spool. This is the default setting.

The Teradata database (version 13.10 and higher) supports NoSpool . If the Teradata database version being accessed does not support NoSpool, it uses Spool mode instead.

The NoSpool mode exports the contents of a table as fast as possible without reading the table into a spool file or distributing the file to all AMPs before extracting it.

### Limitations and Functionality

The NoSpool mode applies only to simple SELECT statements.

The following are not supported by NoSpool:

- Access to nondata tables, such as SELECT DATE or SELECT USER
- USING modifier;
- Contains a SORT (ORDER BY), HAVING, or WITH clauses
- Joins
- Aggregations (Explain shows SUM step)
- TABLE functions
- Ordered-analytic (OLAP) functions
- Multiple SELECT statements or multistatement requests
- Statements with zero or more than one, retrieve or sampling step

The NoSpool mode only retrieves data from a single table, but the SELECT statement can be selective about which columns are exported and can constrain the job to a subset of rows.

Scalar expressions/functions are allowed. The Sample and partition eliminating constraints are supported.

### Disadvantages of the NoSpool Mode

- Locks are maintained during the entire export process.

- Data conversion errors previously detected during the spooling phase will not be detected until the block is read, which could occur any time during the export.



**Notice!** With Teradata version 13.10 and 14.10, there is a Teradata bug in FastExport related to the TD\_SPOOLMODE setting that could cause Teradata to crash. This can occur if the default configuration (NoSpool) is used for SPOOLMODE in the Teradata-Aster Database Connector. If using the Connector in the default SPOOLMODE (NoSpool) causes Teradata to crash, you can fix this by setting the SPOOLMODE argument in the Connector to Spool as follows:

```
SPOOLMODE ('Spool')
```

See Teradata Tech Alert NTA 2836 for details:

Wrong results or 7487 DBS Restart may occur for Fast Export job having SSQ with NOSPOOL option

<http://pc01.teradata.com/support/general/newcase.nsf/T/NTA+2836>

## Troubleshooting

If a data load fails, the approach for troubleshooting differs, depending which function you are using.

### Troubleshooting common to both Teradata Connector functions

- 1 For both functions, the most common problem is that the mapping of the schema is inconsistent between the two tables. Ensure that both tables have the same structure (columns, datatypes, etc.). If you are doing a datatype conversion as part of the transfer, check “Datatype Support” on page 305 and/or the Teradata document “SQL Functions, Operators, Expressions, and Predicates” which can be found here: <http://www.info.teradata.com/edownload.cfm?itemid=102320046>.
- 2 Data inconsistencies (constraint violations, hidden characters, failure of datatype conversion) can also cause problems, so check your data when in doubt.
- 3 In case of a Teradata database shutdown, a pre-existing query using the Teradata Connector on Aster Database will continue to wait for up to four hours for a reply from Teradata. The query will resume and run to completion upon a restart of the Teradata database. During the wait period, the query will show a status of “Running” on Aster Database.
- 4 During execution of the Connector functions, an error message may be returned from either Aster Database or Teradata. First, determine whether an error message is likely to be based on a problem on the Aster Database side or on the Teradata side. Aster Database errors have a format like:

```
ERROR: <Operation> failed with status code <codenumber>:  
<error_description>.
```

You may also see errors from Teradata. These will look similar to the following example:

```
ERROR: Teradata Database returned error in DBCHCL(CON) : MTDP:  
EM_NOHOST(224) : name not in HOSTS file or names database.
```

- 5 Ensure that in the Teradata setting TPT Export/Load utility session limit is set to a value greater than the number of Aster vworkers. This limit can be applied using the TASM rule setting. You should also be sure to use the NUM\_INSTANCES () argument in your query.

## **load\_from\_teradata Troubleshooting**

When a copy of data from Teradata to Aster Database fails, the entire function will fail. You should check the schema mapping and correct any data inconsistencies. Then delete all rows in the receiving table, if necessary, and attempt to run the function again. Note that using a CTAS to create the target table in Aster Database may solve any schema mapping problems that can occur when attempting to manually create the target table.

One common cause of query failure is the difference in how Aster Database and Teradata handle mixed case table and column names. See “[Handling Upper Case Letters in Imported Table Column Names](#)” on page 298.

### **Skip Rows with Data Errors using SKIP\_ERROR\_RECORDS**

When using load\_from\_teradata, the connector can encounter an error when parsing a row with data that is valid in Teradata but is not supported in Aster Database. For example, the Teradata database allows a CHAR column in a table to have null characters (characters with code 0). However, such data is not supported in CHAR columns in Aster. When running load\_From\_teradata on this table, the following error is returned:

```
Error: String (byte length=8) contains non-UTF8 character (starting with byte = 0x0)
```

Rows that produce this type of error can be bypassed by specifying the argument clause SKIP\_ERROR\_RECORDS('YES') in the load\_from\_teradata() function call.

### **Quoting for Literals Using Multiple Lines**

You can use the following characters to quote literals in Aster Database.

- two dollar signs \$\$
- backslash escaped quotes \ '
- two single quotes ''

This is useful when specifying the QUERY clause. For example:

```
CREATE TABLE aster_target_table DISTRIBUTE BY HASH(td_partition_col) AS
SELECT
*
FROM load_from_teradata(
    ON mr_driver -- empty Aster mr_driver table
    TDPIID('dbc') -- Teradata hostname
    USERNAME('vmuser')
    PASSWORD('vmpass')
    QUERY($$SELECT * FROM td_src_table$$
        $$WHERE date_col between '2012-06-05' and '2012-12-10';$$)
);
```

Notice that if you want to split the part of the query that is quoted into multiple lines, each line requires opening and closing quote of its own.

Here are some examples showing different valid ways to use quotes within queries:

- 1 A multi-line query of the form:

```
SELECT id FROM employees  
WHERE firstname LIKE 'Joe';
```

Can be dollar quoted within the query clause as follows:

```
QUERY($$SELECT id FROM employees $$  
$$ WHERE firstname LIKE 'Joe' $$)
```

- 2 The same query using backslash escaped quotes would look like:

```
QUERY('SELECT id FROM employees '  
' WHERE firstname LIKE \'Joe\' ' )
```

- 3 The same query with escaping using two single-quotes would look like:

```
QUERY('SELECT id FROM employees '  
' WHERE firstname LIKE ''Joe''' )
```

## **load\_to\_teradata Troubleshooting**

When copying data to Teradata, any rows that fail to load will be logged in the two error tables on Teradata. These are the error tables you specified in the `load_to_teradata` function. If you did not specify a name for one or both of the error tables, the default error table name(s) will be used (`<TARGET_TABLE>_ET` for the first error table and `<TARGET_TABLE>_UV` for the second error table).

Query the error table(s) to discover any problems that occurred during the data copy.

Error messages in Teradata have a numeric code that can be used to look up the error in the Teradata documentation. See the following Teradata documentation (for version 13.10), or the corresponding documents for the Teradata release you are using.

The Messages manual, which can be used to look up error messages, can be downloaded here:

<http://www.info.teradata.com/edownload.cfm?itemid=102320024>

The FastLoad documentation can be downloaded here:

<http://www.info.teradata.com/IndexedPdfs/101320006.pdf>

The FastExport documentation can be downloaded here:

<http://www.info.teradata.com/IndexedPdfs/101320004.pdf>

For other versions of Teradata and/or other manuals, search here:

<http://www.info.teradata.com/>

## **NUM\_INSTANCES Error**

If you observe the following error message, the message is incorrect:

```
"NUM_INSTANCES cannot exceed total task count of..."
```

This means that the `NUM_INSTANCES` argument has been set to a value higher than total number of vworkers. Set this argument to a value equal to or lower than the total number of vworkers.

## Copying Rows Containing varchar Data to Teradata Fails

When copying rows with a column of datatype varchar from Aster Database to Teradata, if the target Teradata column is of the type CHARACTER SET LATIN, the transfer will fail with ErrorCode 6706. This happens because all varchar columns in Aster Database are encoded in UTF-8.

The Teradata column will have the CHARACTER SET LATIN datatype if one of the following is true:

- the column itself is specified to be of “Latin” type
- the Teradata database was created with the default character set “Latin”

If this is the case, assuming a target table in Teradata named “td\_table”, any rows that failed to transfer would instead go into the error table “td\_table\_ET”. The error code logged would be ErrorCode 6706.

To resolve this issue, do one of the following:

- ensure that the default character set for the target Teradata database is UNICODE, or
- create all varchar columns with "CHARACTER SET UNICODE".

## Not Enough PERM Space Error

When using the log\_table argument clause, you may see a error message saying that there is “not enough PERM space available in the default database”.

This error may be returned under these conditions:

- 1 the Teradata database user has a default logon database defined, which is different from the database of the target table, and
- 2 The default database does not have enough PERM space available to store the log table.

If you see this error, you should specify a fully qualified log table name using the log\_table argument clause. Otherwise, an unqualified log table name is acceptable.

# Limitations

- 1 Export to Teradata must happen into a pre-created, empty table (this is a requirement for Teradata FastLoad).
- 2 Teradata Fastload will not load duplicate rows from your data source to the Teradata Database. (A duplicate row is one in which every field contains the exact same data as the fields of an existing row.) This is true even for MULTISET tables.
- 3 Note that while Aster Database supports arbitrary length varchar columns, Teradata does not. In addition, the Teradata TPT API supports a maximum of 32000 bytes in a varchar column during data transfer.

Due to this mismatch in limitations, if the number of bytes exceeds 32000 bytes in a varchar column during transfer, load\_to\_teradata will fail with an error.

Therefore, the Connector has a limit of 32,000 bytes maximum in any varchar column. For example:

- If the data has all ASCII characters, the column can have up to 32000 characters.
  - If the data has multibyte UTF-8 characters, in the worst case, the column can have up to  $32000/3$  characters (assuming all 3-byte UTF 8 characters).
- 4 Imports from and exports to Teradata are not resumable. If a connection times out, the SQL-MR function will be rolled back. Unlock any locked Teradata tables if necessary, drop and recreate the target table, and attempt the import or export again.
  - 5 In Teradata version 12, FastExport cannot support result sets with only a single row. Because of this, single row loads from Teradata version 12 using `load_from_teradata` will not work, and should not be attempted.

## CHAPTER 8 SQL-H: The Hadoop/HCatalog Connector

### Introduction to SQL-H

Apache Hadoop is an open source platform for storing and managing big data. Teradata Aster SQL-H is a software access method which provides a bridge that enables users to:

- 1 Easily analyze data stored in Hadoop through standard ANSI SQL and Aster's SQL-MapReduce framework, and
- 2 Copy data from Hadoop to Aster Database.

HCatalog is a table metadata service for Hive, Pig, etc. SQL-H provides deep metadata layer integration with the Apache HCatalog project. SQL-H enables business users to access the Hadoop data from Aster Database directly. Aster Database manages communication with Hadoop nodes through SQL-H to read data for SQL queries and SQL-MR functions.

This document contains the following sections:

- [Configuring Aster Database and HCatalog](#)
- [The load\\_from\\_hcatalog SQL-MR Function](#)
- [Display the HCatalog Metadata in ACT](#)
- [Using SQL-H](#)
- [SQL-H Argument Clauses](#)
- [Datatype and Storage Type Support](#)
- [Tips for Working with SQL-H](#)
- [Troubleshooting](#)
- [Limitations](#)

For Hadoop and HCatalog documentation from Hortonworks, see:

- [http://docs.hortonworks.com/CURRENT/index.htm#HDP\\_Documentation.htm](http://docs.hortonworks.com/CURRENT/index.htm#HDP_Documentation.htm)

For more general information on Hadoop and HCatalog, see:

- <http://incubator.apache.org/hcatalog/>
- <http://hadoop.apache.org/>

## Benefits of SQL-H

Some of the benefits of using SQL-H are:

- 1 Allows direct access to the HCatalog metadata from Aster Database. For example, you can list all databases and tables in HCatalog from Aster using the Aster Database Cluster Terminal (ACT).
- 2 Supports fetching data from HCatalog, and the automatic mapping of HCatalog datatype value to the Aster datatype value.
- 3 Supports partitions and partition pruning on HCatalog, to improve query performance when querying HCatalog from Aster Database. (Don't confuse HCatalog partitions with Aster Database's "logical partition" feature, which is described in "[Automatic Logical Partitioning](#) on page 39.)
- 4 Enables complex queries which include some data from Hadoop and some data from Aster Database through SQL.
- 5 Extends Aster Analytics library over data stored in Hadoop.
- 6 Provides a simple way to load Hadoop data into Aster Database tables - Extract, Transform, Load (ETL).
- 7 Creates views of HCatalog data within Aster Database, which enables Business Intelligence (BI) tools to generate queries against data in Hadoop.

## Configuring Aster Database and HCatalog

There is no special installation needed to enable using SQL-H. All required Hadoop packages and HCatalog jars for certified distributions are installed during the normal Aster Database installation or upgrade. However, you do need to set up SQL-H Configuration for each Hadoop cluster you will access.



**Tip!** If you are upgrading from a pre-5.10 version of Aster Database, you will need to perform the procedure in [Create a SQL-H configuration](#) for each Hadoop cluster, or any SQL-H queries will fail.

## Prerequisites

SQL-H comes with Aster Database, and does not require any additional installation. SQL-H has the following prerequisites:

- Aster Database version 5.0 or higher.
- One of the Hadoop distributions that is certified for your version of Aster Database. These are listed in [Certified Hadoop distributions](#).
- HCatalog (the version that comes with your Hadoop distribution).
- Network connectivity between Aster Database and Hadoop for all nodes.

## Certified Hadoop distributions

SQL-H is tested and certified on the following distributions and versions of Hadoop on the appliance (for unsecured clusters, only). Make sure to upgrade your version of Hadoop to match your version of Aster Database, if necessary:

Table 8 - 1: Aster Database and Hadoop SQL-H Compatibility

| Aster Database Version | Hortonworks Data Platform (HDP) Version | Hadoop Platform    |
|------------------------|-----------------------------------------|--------------------|
| 5.0                    | HDP 1.0                                 | appliance          |
| 5.0.1                  | HDP 1.1                                 | appliance          |
| 5.0.2                  | HDP 1.1                                 | appliance          |
| 5.10                   | HDP 1.1                                 | appliance          |
|                        | HDP 1.2*                                | commodity hardware |
| 5.10.00.01             | HDP 1.1                                 | appliance          |
|                        | HDP 1.3.2                               | appliance          |

\* To enable support for this version combination, issue the following ncli command on the queen node:

```
# ncli sqlh setversionconfig --version=HDP1.2
--hcat_jar_dir=/home/beehive/partner/hadoop/HDP1.2
--comment='Hortonworks Data Platform 1.2'
```

This will add 'HDP1.2' to the AMC dropdown list so you can add the SQL-H configuration for HDP1.2. Next, follow the steps in ["Create a SQL-H configuration" on page 324](#).

## Configure networking for SQL-H

To configure connectivity between Aster Database and Hadoop, there are two considerations:

- 1 Hostnames and DNS
- 2 Hadoop Ports



**Tip!** Ensure that you use the same hostname that is used for the installation on the Hadoop cluster. It is recommended to use the fully qualified domain name (FQDN) for all the Hadoop host machines.

### Hostnames and DNS

Every node in Aster Database must have access to all Hadoop nodes. On some versions of the appliance, this was not the case in the out-of-the-box configuration. If you have one of those versions, you will see the following error when attempting to use SQL-H:

```
ERROR: SQL-MR function LOAD_FROM_HCATALOG failed: Failed to read data
from hcatalog. task : 46. Details : Error in RPC response header:
AppError
```

To correct this, make sure that every Aster node can resolve the address of every Hadoop node, using the same format of the hostname as found in the Hadoop file <master>: /etc/hadoop/conf/slaves. This will typically be something like hdp001-xx. To do this:

- 1 Copy the file /etc/hadoop/conf/slaves from the Hadoop master to your Aster queen.
- 2 Compare the slaves file to the /etc/hosts file. If there are any hostname entries that are not present in the Aster hosts file, add them.
- 3 Copy the new /etc/hosts file from the queen onto each worker and loader node.

Alternatively, you may set up Host/DNS entries for Hadoop on all the Aster Database nodes through the AMC, as described in the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*.

## Example

Problem:

Aster appliance systems come with a /etc/hosts file with entries like this:

```
39.64.8.7 byn001-7
```

The Hadoop master (and data) nodes have entries like this:

```
39.64.8.7 hdp001-7 byn001-7
```

The Hadoop master node determines that the hdp001-x (from the /etc/hadoop/conf/slaves file on the name node) is the naming convention to give back to Aster Database. Aster then tries to resolve those names and cannot because they are not in the Aster /etc/hosts file.

Solution:

Update the Aster /etc/hosts file to use the same form as the Hadoop nodes. Include the hdp001-x and byn001-x names on the same lines.

For example, use this format in the Aster /etc/hosts:

```
39.64.8.7 hdp001-7 byn001-7
```

## Hadoop Ports

The following ports must be reachable from all Aster Database nodes. Make sure your firewall doesn't block them:

Table 8 - 1: Hadoop ports that must be reachable by all Aster nodes

| Host          | Port  | Comment            | Purpose                             |
|---------------|-------|--------------------|-------------------------------------|
| Hive Server   | 50111 | Templeton server   | Fetch table metadata                |
| Hive Server   | 9083  | Hive thrift server | Fetch table metadata and split info |
| Namenode      | 8020  | Namenode server    | Get file metadata                   |
| ALL Datanodes | 50010 | Datanode server    | Read HDFS files                     |

## Hardware configuration

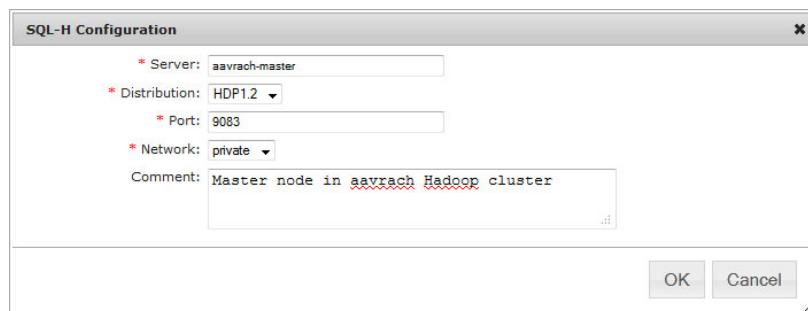
It is recommended that Aster Database and HCatalog/Hadoop reside on the same network switch for the best performance when transferring data.

In addition, some best practices have been developed for connectivity between Aster Database and Teradata. The general principles are the same for connecting to HCatalog. These best practices can be found in “[Hardware Best Practices](#)” on page 293.

## Create a SQL-H configuration

Each Hadoop cluster you wish to access must have a SQL-H configuration. You can either use the AMC or ncli to create a SQL-H configuration. To use the AMC:

- 1 Log in to the AMC as a user with the administrator role.
- 2 Navigate to Admin > Configuration > SQL-H Configuration.
- 3 Click on New SQL-H Configuration Entry.
- 4 Fill in the values for your Hadoop master node. Note that the Server value you specify here must be the same one you will use when invoking SQL-H:
  - a Server: The hostname of your Hive server. Note that if the Hive server and namenode have different hostnames, you must specify the Hive server hostname, not the namenode hostname.
  - b Distribution: The version of your Hadoop cluster.
  - c Port: The port on which the Hive server listens. This is generally port 9083.
  - d Network: The type of network - public or private. For information on how to set this, see “[Configure networking for SQL-H](#)” on page 322.
  - e Comment: A comment related to this configuration.



- 5 Click OK. This saves your SQL-H configuration.
- 6 You will see the entry you just made in the SQL-H Configuration list.

A screenshot of the Teradata Aster management interface. The top navigation bar shows 'nCluster: Unnamed is Active' and the date '4/3/13 2:25pm'. On the right, it says 'TERADATA ASTER' and 'Logged in as: db\_superuser logout'. Below the bar are tabs for 'Dashboard', 'Processes', 'Nodes', 'Admin' (selected), 'Cluster Management', 'Events', 'Executables', 'Backup', 'Configuration', and 'Logs'. A green success message box in the center says 'Applied SQL-H configuration entry changes successfully'. Under the 'Admin' tab, there's a section titled 'SQL-H Configurations' with a sub-section 'SQL-H configuration entries to specify Hadoop version for Hadoop clusters'. A button 'New SQL-H Configuration Entry' is visible. A table lists configurations: hdp1.aster.com (Server), HDP1.1 (Version), 9083 (Port), private (Network), test data (Comment). There are 'Edit' and 'Delete' icons next to each row.

- 7 To edit an SQL-H configuration, click the pencil icon; to delete it, click the X icon.
- 8 Repeat these steps to create a configuration for each Hadoop cluster you want to access via SQL-H.

## The load\_from\_hcatalog SQL-MR Function

The load\_from\_hcatalog SQL-MR function brings data from Hadoop into Aster Database. The function is invoked on Aster Database.

### load\_from\_hcatalog Usage

The load\_from\_hcatalog SQL-MR function is a regular, single input SQL-MR function. It must be invoked on an empty, fact table in Aster Database. This is usually done by using a dummy table, which will be referred to as “mr\_driver”, created as follows:

#### Creating the mr\_driver table

```
CREATE TABLE mr_driver(
    c1 INT)
DISTRIBUTE BY HASH (c1)
;
```

#### Syntax of load\_from\_hcatalog

Here is the full syntax for load\_from\_hcatalog, where mr\_driver is an empty fact table in Aster Database:

```
SELECT * FROM load_from_hcatalog(
    ON mr_driver
    SERVER ('server_hostname')
    [PORT('server_port')]
    USERNAME('username')
    DBNAME('database_name')
    TABLENAME('hcat_tablename')
    [PARTITIONFILTER('expression_string')]
    [COLUMNS('col1'[, 'col2', ...])]
    [TEMPLETON_PORT('REST_port')]
    [SKIP_ERROR_RECORDS('yes' | 'no')]
) ;
```

For an explanation of what the clauses mean, and what parameters should be passed, see “[SQL-H Argument Clauses](#)” on page 331.

The output schema of the load\_from\_hcatalog SQL-MR function includes the specified columns, and the partition key(s), if supplied. When using \*, both the columns and partition key(s) are returned.



**Tip!** You can issue commands in ACT to display metadata about external HCatalog systems and their databases, schemas and tables. See [Display the HCatalog Metadata in ACT \(page 326\)](#) for more information.

## Display the HCatalog Metadata in ACT

To display metadata about HCatalog databases and tables, you can use ACT. For more information on ACT, see [Aster Database Cluster Terminal \(ACT\) \(page 58\)](#). In ACT, the following commands provide direct access to the HCatalog metadata. Each is described in the sections that follow.

- `\extl host=hostname [option_name=option_value ...]`  
lists all databases on an external system
- `\extd host=hostname database=dbname [option_name=option_value ...]`  
lists all tables in a database or describes a table on an external system

The option names for each of these commands are case insensitive, but the option values are case sensitive. The `option_name=option_value` pairs are delimited by space(s) or tab(s). Spaces and tabs can be embedded in the option name or the option value, if enclosed by a pair of double quotes. Unsupported option names and their values, if specified, will be ignored by ACT.

### List databases in HCatalog

Use the command `\extl` to list databases in HCatalog for a given host.

#### Syntax of `\extl`

```
\extl host=hostname [systemtype=hcatalog] [port=port_number]  
[user=username] [password=pwd]
```

#### Options for `\extl`

*host* - Required. Host name of the node where HCatalog REST server resides.

*systemtype* - Optional. The only valid value is “hcatalog” (case insensitive). If not specified, defaults to “hcatalog”.

*port* - Optional. HCatalog REST server port, also known as Templeton port. If not specified, defaults to 50111.

*user* - Optional. The database user on the external system to use. If not specified, defaults to the value of the current Aster Database user who is logged in to ACT.

*password* - Optional. If not specified, defaults to an empty string.

### Example of \extl usage

Return a list of databases in the external database:

```
beehive> \extl host=hdp1.aster.com port=50111 user=beehive
systemtype=Hcatalog
```

```
List of databases
  Name
-----
default
www_data
finance_dept
```

## List tables or describe a table in HCatalog

Use the command `\extd` *without* the `table` option to list tables in the specified database in HCatalog. Use the command `\extd` *with* the `table` option to describe the specified table in the specified database in HCatalog.

### Syntax of \extd

```
\extd host=hostname [database=dbname] [systemtype=systemtype_value]
[port=port_number] [user=username] [password=pwd] [table=tablename]
```

### Options for \extd

`\extd` uses the same options as `\extl` (see [Options for \extl \(page 326\)](#)), plus the following:

*database* - Optional. Name of the database to access on the external system containing the table. Defaults to "default".

*table* - Optional. Name of the table to describe. If not specified, returns a list of the tables in the specified database.

*format* - Optional. This option is only applicable when describing a table. When the value is set to "extended", additional table metadata, (e.g. partition list, file info) will be displayed.

### Example of \extd usage

#### Example 1:

Return a list of tables in the external database:

```
beehive=> \extd host=hdp1.aster.com port=50111 database=www_data

List of tables
  Name
-----
Raw_Jan_2012
Raw_Feb_2012
Raw_Mar_2012
(3 rows)
```

#### Example 2:

Describe a table in the external database:

```
beehive=> \extd host=hdp1.aster.com port=50111 database=www_data table=
Raw_Mar_2012
```

```
Table "www_data"."Raw_Mar_2012"
Column      | Type   | Partitioned Column
-----+-----+-----
page_id    | int    | f
category_id | int    | t
click_date  | string | t
click_time  | string | f
(4 rows)
```

**Example 3:**

Describe a table in the external database using the extended format:

```
beehive=> \extd host=hdp1.aster.com port=50111 database=www_data
table=Raw_Mar_2012 format=extended
```

```
Table "www_data"."Raw_Mar_2012"
Column      | Type   | Partitioned Column
-----+-----+-----
page_id    | int    | f
category_id | int    | t
click_date  | string | t
click_time  | string | f
(4 rows)
```

Owner:  
hdfs

Location:  
hdfs://plvm09-c3-namenode.asterdata.com:8020/e2ehdptest/Raw\_Mar\_2012

Number of files:  
0

Min file size:  
0

Max file size:  
0

Input format:  
org.apache.hadoop.mapred.TextInputFormat

Output format:  
org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat

Last update time:  
2012-05-17 10:31:22-07

```
Last access time:
      List of partitions
Partition Number | Column Name | Column Value
-----+-----+-----+
          0 | category_id   | 012
          0 | click_date    | 20120301
          1 | category_id   | 119
          1 | click_date    | 20120301
          2 | category_id   | 254
          2 | click_date    | 20120301
          3 | category_id   | 663
          3 | click_date    | 20120301
(8 rows)
```

## Using SQL-H

### Using `load_from_hcatalog` to Create a View

You can use a SQL-MR function to create a view representing an HCatalog table within Aster Database. To create and use the HCatalog table view, define a view using the `load_from_hcatalog` SQL-MR function, and then reference it as a remote HCatalog table. The standard CREATE VIEW syntax can be used. In all examples below, `mr_driver` is an empty fact table in Aster Database.

As long as the view contains the partition column(s), subsequent queries will do partition pruning if the WHERE clause contains a partition key/value combination(s). This is because the planner rewrites those queries to use the PARTITIONFILTER argument when calling the `load_from_hcatalog` function to retrieve data.

#### Example 1: Creating a view over `load_from_hcatalog`

In this example, we will create a view `hcat_view` to represent a table `hcat_table` in HCatalog. Note that we have included the partition key reference in the SELECT clause. We will later use it in the WHERE clause when querying the view.

```
CREATE VIEW hcat_view AS
  SELECT hcat_table.partition_col, hcat_table.col1
    FROM load_from_hcatalog (ON mr_driver
      SERVER ('hdp1.aster.com')
      USERNAME ('user1')
      DBNAME ('test')
      TABLENAME ('table1') AS hcat_table)
    ) ;
```

The definition of the view created will be stored in Aster Database. When a user later queries the view, during query preprocessing the view is replaced with its query definition.

#### Example 2: Querying an HCatalog View

In [Example 1: Creating a view over `load\_from\_hcatalog`](#), we created the view in Aster Database. After that, we can issue queries against the view itself:

```
SELECT * FROM hcat_view WHERE partition_col = '2011-09-01';
```

## Partition Pruning with load\_from\_hcatalog

When querying HCatalog using the load\_from\_hcatalog function or a view created with the function, you can specify a partition filter in the WHERE clause of the query by including a partition key/value combination(s). The optimizer will push down the partition filter into HCatalog to do partition pruning. Although the partition columns are not part of the actual data in an HCatalog table, load\_from\_hcatalog includes the partition columns in its output so they can be referenced in the WHERE clause.

The following partition filter formats are supported when querying a view:

- equality predicate as in "partition\_col = string\_value"

Example:

```
WHERE order_month = '2011-09';
```

- IN predicate, where the expressions in the included list are string literals

Example:

```
WHERE order_month IN('2011-09', '2011-10', '2011-11');
```

- AND/OR within an equality or IN predicate

Example:

```
WHERE order_month IN('2011-09', '2011-10', '2011-11') AND region =
'US_west' OR 'US_east';
```

Note that the supported type of partition filter is a list of string values.

### Example 3: Partition pruning by a list of values

The following example shows a query against the view created in [Example 1: Creating a view over load\\_from\\_hcatalog](#), that specifies the partitions to include using a list of values.

```
SELECT * FROM hcat_view WHERE partition_col IN ('2011-09-01', '2011-09-02') AND col1='abc';
```

When querying the view as shown above, the SQL is rewritten internally and issued as:

```
SELECT *
  FROM load_from_hcatalog (ON mr_driver
    SERVER ('hdp1.aster.com')
    USERNAME ('user1')
    DBNAME ('test')
    TABLENAME ('table1'))
  PARTITIONFILTER ('partition_col = "2011-09-01" OR partition_col =
"2011-09-02"')
  COLUMNS ('partition_col', 'col1'))
  WHERE col1 = 'abc';
```



**Tip!** It is recommended to specify both the partition filter and the columns in the WHERE clause rather than using the PARTITIONFILTER and COLUMNS argument clauses in load\_from\_hcatalog. Aster Database will transform these values into the predicates that HCatalog supports.

For example, Aster converts the

```
WHERE partition_col IN ('partition1','partition2')
```

predicate to the OR expression of equality predicate as the argument to the function

```
PARTITIONFILTER ('partition_col = "partition1" OR partition_col = "partition2"')
```

## SQL-H Argument Clauses

### load\_from\_hcatalog Clauses

#### HCatalog Table Identifiers

The table in HCatalog will be uniquely identified using these argument clauses:

- SERVER

The HCatalog metadata server hostname to which Aster Database should connect. Note that if the metadata server and namenode have different hostnames, you must specify the metadata server hostname, not the namenode hostname. The hostname you supply must have a corresponding SQL-H configuration. However, you may specify a fully qualified hostname, even if the hostname in the SQL-H configuration is not fully qualified. This clause is required.

- PORT

This argument is deprecated. Do not specify it; the Port setting you specified in “[Create a SQL-H configuration](#)” on page 324 will be used automatically. This argument was used in prior versions of SQL-H to specify the HCatalog metadata server port, and is retained for backwards compatibility. This clause is optional, but not recommended.

- DBNAME

The database in HCatalog where this table resides. This clause is required.

- TABLENAME

The tablename in HCatalog. This clause is required.

#### Additional Arguments

- ON *mr\_driver*

The load\_from\_hcatalog function requires a pre-existing *mr\_driver* fact table to be present in Aster Database. This table is referenced in the ON clause of the load\_from\_hcatalog function. The data structure of the *mr\_driver* table is not important, but it should be an empty table. For an example of creating the *mr\_driver* table see [Creating the mr\\_driver table \(page 325\)](#). Data is never loaded into the *mr\_driver* table; it is simply used to fulfill the syntax requirements of the SQL-MR function. This clause is required.

- `USERNAME ('username')`

The user who owns this table in Hcatalog. This clause is required.

- `PARTITIONFILTER`

The partition filter limits the partitions from the HCatalog table that the function will act upon. The column names should be specified using lowercase. The partition filter must contain equality comparisons of the form `colname="colvalue"` and the equalities may be combined with AND/OR operators. The partition filter has the format:

```
PARTITIONFILTER('partition_key = "value1"' [AND|OR 'partition_key = "value2"' ...]).
```

Although you cannot use the IN operator in the partition filter, you may use it in the WHERE clause of a view that is created based on the `load_from_hcatalog` function.

If the partition filter is omitted, the function will fetch from all partitions. This clause is optional.

- `COLUMNS`

This argument is no longer necessary, and you should not use it. The Aster planner generates the list of required columns for the query automatically. This argument was used in prior versions of SQL-H to specify the columns to retrieve, and is retained for backwards compatibility. Specifies one or more columns to act upon. The column names should be specified using lowercase. Any columns with names of mixed case will be brought over with names as all lowercase. This clause is optional, but not recommended.

- `TEMPLETON_PORT`

The HCatalog REST server, also known as the Templeton port. If not specified, default value 50111 will be used. This clause is optional.

- `SKIP_ERROR_RECORDS ('yes' | 'no')`

Use this argument to specify whether or not to skip error records from SQL-H. If set to 'yes', SQL-H will capture any IOException/RunTimeException errors and continue executing. You can find out the total number of skipped records by looking in the SQL-MR logs. If set to 'no', errors will cause the function execution to stop. The default is 'no'. This clause is optional.

## Datatype and Storage Type Support

This section describes supported datatypes and storage types and provides notes on datatype conversion. You may also need to consult the Hadoop documentation for your version of Hadoop.

Any given new Hadoop release may include new datatypes which are supported by Hive but not supported by HCatalog. In HDP 1.3.2, this list includes decimal, timestamp and union. SQL-H does not support these data types because they are not supported in HCatalog.

### Supported Hadoop storage types

The following Hadoop storage types are supported:

- `RCFile`

- text files (CSV, TSV)
- JSON
- Optimized Row Columnar (ORC) files beginning in HDP1.3.2. This includes support for Snappy compression on ORC files.

## HCatalog to Aster Database datatype conversions

This section covers datatype conversion when copying from Hadoop to Aster Database. Note that most of these datatypes are supported by both systems. For datatypes that are not common to both systems, an implicit datatype conversion will be performed by Aster Database when data is read by or transferred to Aster Database.

For the HCatalog datatype binary, the data is converted to the bytea datatype by the load\_from\_hcatalog function. If the source binary data is base64 encoded it will be decoded by the function automatically.

Table 8 - 1: Supported Datatypes

| HCatalog Datatype | Aster Datatype                |
|-------------------|-------------------------------|
| tinyint           | smallint                      |
| smallint          | smallint                      |
| int               | int                           |
| bigint            | bigint                        |
| float             | real                          |
| double            | double                        |
| boolean           | String (t   f)                |
| string            | varchar                       |
| array             | varchar (JSON representation) |
| map               | varchar (JSON representation) |
| struct            | varchar (JSON representation) |
| binary            | bytea                         |

## Example of datatype conversion

The following example shows the process of creating a table in HCatalog, adding data to Hadoop, and querying the data using the load\_from\_hcatalog function. The example shows how JSON data is displayed when queried from Aster Database.

### ***Create the table in HCatalog***

Issue the following to create the source JSON table jsontable in HCatalog:

```
CREATE EXTERNAL TABLE jsontable(
    s string,
    i int,
    d double,
    m map<string, string>,
    bb array<struct<a: int, b: string>>)
row format serde 'org.apache.hcatalog.data.JsonSerDe'
STORED AS TEXTFILE
location '/user/hcat/jsontable';
```

### **JSON input**

Next, input data to the table. This example uses a JSON file as input. Here is an example of the contents of the JSON file:

```
{"s":"ulysses miller", "i":110265804, "d":-41.8176261064764,
"m": {"paris" : "quinn young", "marseilles" : "oscar
thompson", "harrisburg" : "yuri ovid", "grenoble" : "zach
quirinius", "roswell" : "david davidson"}, "bb": [{"a": -1248232584, "b": "calvin miller"}, {"a": -217392663, "b": "quinn
johnson"}, {"a": 83518649, "b": "jessica johnson"}]}
{"s": "david nixon", "i": -1828423853, "d": 1.84110104854699, "m": {"zippy"
: "ulysses carson", "umatilla" : "fred white", "wheaton" : "yuri
underhill"}, "bb": [{"a": -1065615233, "b": "victor garcia"}, {"a": -1270226546, "b": "gabriella underhill"}, {"a": -201437209, "b": "tom
white"}, {"a": 1663961257, "b": "jessica davidson"}]}
{"s": "ethan garcia", "i": -694926894, "d": 298.889104994985, "m": null,
"bb": [{"a": -1100934195, "b": "zach laertes"}, {"a": -1945910877, "b": "victor
ellison"}]}
{"s": "nick quirinius", "i": -542935355, "d": -346.371977618564,
"m": {"zippy" : "luke johnson", "queensville" : "irene king", "xacky" :
"quinn johnson", "wheaton" : "victor miller", "nice" : "yuri garcia"}, "bb": [{"a": -1771422625, "b": "yuri ichabod"}, {"a": -1196274459, "b": "alice
hernandez"}, {"a": -521469660, "b": "luke ellison"}, {"a": -656550123, "b": "xavier young"}, {"a": -461918635, "b": "david falkner"}]}
```

Next, you would issue a command to load the data into HDFS from the JSON file shown above:

```
# hadoop dfs -put <localJSONfilename> /user/hcat/jsontable
```

### **Output in Aster Database**

This example shows what is output when issuing a query against HCatalog from within Aster Database. The data is converted to the varchar datatype, formatted to appear as a JSON representation.

To retrieve the data from Aster Database, do the following:

1 Log in to ACT:

```
# act -w <password>
```

2 Turn “expanded” on:

```
beehive=> \x
Expanded is On
```

3 Issue the call to load\_from\_hcatalog to retrieve the data:

```
beehive=> SELECT * FROM load_from_hcatalog (
beehive=> ON empty_hcat_table
beehive=> SERVER('hdp1.aster.com')
beehive=> DBNAME('default')
beehive=> TABLENAME('jsontable')
beehive=> USERNAME('hcat'));

- [ RECORD 1 ] -----
-----
s | ulysses miller
i | 110265804
d | -41.8176261064764
m | {"grenoble":"zach quirinius","harrisburg":"yuri
ovid","marseilles":"oscar thompson","paris":"quinn
young","roswell":"david davidson"}
bb | [{"a":-1248232584,"b":"calvin miller"}, {"a":-217392663,"b":"quinn
johnson"}, {"a":83518649,"b":"jessica johnson"}]
- [ RECORD 2 ] -----
-----
s | david nixon
i | -1828423853
d | 1.84110104854699
m | {"umatilla":"fred white","wheaton":"yuri
underhill","zippy":"ulysses carson"}
bb | [{"a":-1065615233,"b":"victor garcia"}, {"a":-
1270226546,"b":"gabriella underhill"}, {"a":-201437209,"b":"tom
white"}, {"a":1663961257,"b":"jessica davidson"}]
- [ RECORD 3 ] -----
-----
s | ethan garcia
i | -694926894
d | 298.889104994985
m |
bb | [{"a":-1100934195,"b":"zach laertes"}, {"a":-1945910877,"b":"victor
ellison"}]
- [ RECORD 4 ] -----
-----
s | nick quirinius
i | -542935355
d | -346.371977618564
m | {"nice":"yuri garcia","queenville":"irene king","wheaton":"victor
miller","xacky":"quinn johnson","zippy":"luke johnson"}
bb | [{"a":-1771422625,"b":"yuri ichabod"}, {"a":-1196274459,"b":"alice
hernandez"}, {"a":-521469660,"b":"luke ellison"}, {"a":-
656550123,"b":"xavier young"}, {"a":-461918635,"b":"david falkner"}]
```

## Tips for Working with SQL-H

- 1 Always ensure that there is enough disk space on the Aster Database cluster to which the data will be copied, before beginning a data load operation.
- 2 During execution of these functions, an error message may be returned from either Aster Database or HCatalog. First, determine whether an error message is likely to be based on a problem on the Aster Database side or on the Hadoop/HCatalog side. You can view Aster Database logs by using the AMC.
- 3 You may find the Hadoop documentation helpful. For more information on Hadoop and HCatalog, see:  
<http://incubator.apache.org/hcatalog/>  
<http://hadoop.apache.org/>

### Capturing output to a table

For larger result sets, it's a good idea to capture the output from `load_from_hcatalog` to a table in Aster Database, to avoid the need to repeat the load if a query must be run again. There are two ways to do this:

- 1 Create a table in Aster Database to receive the data:

You can first create a table in Aster Database to receive the data from Hadoop. The columns in the Aster Database table must exactly match the names and structure of the columns in the HCatalog table(s) from which the data will be copied. That is, it must have the same column names, with the same or a compatible datatype in each column. But of course, the Aster Database table does not need to include any columns that will be omitted from the query that selects the data to be loaded. If the datatypes for a column do not match, an implicit datatype conversion will be performed. For more information on datatypes, see “[Datatype and Storage Type Support](#)” on page 332. After creating the table, you may then select into it using the `load_from_hcatalog` function.

- 2 Use `CREATE TABLE...AS...SELECT` (CTAS):

Using CTAS will ensure that the data schema of the Aster Database table is compatible with that of the HCatalog table(s) from which the data will be copied, for all the columns that are included in the query. Because SQL-H creates the schema and does datatype conversion automatically, this is an excellent way to easily copy data, without having to analyze the schema and do datatype conversions manually.

Here is how CTAS works with the `load_from_hcatalog` function: During planning, the `load_from_hcatalog` function describes the HCatalog query to determine the schema of the result set. This schema will then be converted into a compatible Aster Database schema, and the SQL-MR function will report that as the output schema of the `load_from_hcatalog` SQL-MR function. Datatype conversion is performed implicitly.

To see what implicit conversions will be applied, see “[Datatype and Storage Type Support](#)” on page 332.

This sample query will create a new table in Aster Database called aster\_target\_table which contains the same rows as the result of the HCatalog query 'SELECT \* FROM hcat\_src\_table'.

```
CREATE TABLE aster_target_table DISTRIBUTE BY HASH(hcat_partition_col)
AS
SELECT *
FROM load_from_hcatalog(
    ON mr_driver -- empty Aster mr_driver table
    SERVER ('hdp1.aster.com')
    USERNAME ('user1')
    DBNAME ('test')
    TABLENAME ('hcat_src_table')
) ;
```

## Troubleshooting

### SQL-H Queries Fail After Upgrade

If you use SQL-H, and you have upgraded from a pre-5.10 version of Aster Database, you will need to create a new SQL-H configuration for each Hadoop cluster you will be accessing. If you do not do this, any SQL-H queries will fail.

### load\_from\_hcatalog Returns an Error

When a copy of data from HCatalog to Aster Database fails, the entire function will fail. You should check the connection arguments. Then DROP the receiving table, if necessary, and attempt to run the function again.

### No SQL-H Configuration Defined

If you attempt to access a Hadoop cluster that does not have a SQL-H configuration defined, you will see an error message like:

```
ERROR: Failed to load hadoop classes
```

This type of error message indicates that you must first [Create a SQL-H configuration](#).

### Concurrent SQL-H Queries Return a “Busy, please retry.” Error

The default Templeton server limit is sixteen concurrent processes. If you attempt more than the allowed number of concurrent processes when using SQL-H, you will see an error message like:

```
ERROR: Failed to access hcatalog table <tablename>. Busy, please retry.
```

To raise the concurrent processes limit, the Templeton setting is `templeton.exec.max-procs`. The Hortonworks documentation contains information on how to raise the default limit:

<http://docs.hortonworks.com/Templeton/CURRENT/wholesite.pdf>

## Error Logs

You can access logs in the AMC in the following ways:

- Logs for the entire database: (See “Aster Database Logging” in the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*)
- Logs for an individual node: (See “Reading Aster Database Logs” in the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*)
- SQL-MR logs: “[Manage SQL-MapReduce Execution](#)” on page 114

## Errors

The following errors may occur on screen when executing SQL-H in ACT. Each error is followed by the reason for its occurrence:

### Failed to Load Hadoop Classes

```
ERROR: Failed to load hadoop classes
ERROR: Failed to load hadoop classes in drainOutputRows
```

Reason: SQL-H required Hadoop classes were not loaded successfully. This can be caused by an invalid or missing SQL-H configuration for the Hadoop cluster. Make sure you add a SQL-H configuration for your Hadoop cluster by following the procedure in “[Create a SQL-H configuration](#)” on page 324.

### Failed to Create HCatRecord

```
ERROR: Failed to create HCatRecord
```

Reason: Most likely, there are some bad records in the Hive table.

You can choose to skip any bad records and complete the SQL-H query by setting the SKIP\_ERROR\_RECORDS argument clause to 'yes', by default it is set to 'no'.

### Permission Denied

```
ERROR: Failed to read data from hcatalog. task : 0. Details :
org.apache.hadoop.security.AccessControlException: Permission denied:
user=mapred
```

Reason: The user does not have the correct permissions. Check if the username specified in the SQL-H query does have the required permissions. If not, either GRANT the user the correct permissions or use a different username.

### Partition Filter Issue

```
ERROR: Failed to initiate data reading from hcatalog table. Details :
MetaException(message:Error parsing partition filter :
NoViableAltException
```

Reason: An invalid syntax was used for the partition filter, or one of the values is missing its enclosing quotes. Check that the partition filter is using the correct syntax, or allow SQL-H to generate the partition filter automatically using the provided SQL query.

### Error in RPC Response Header

```
ERROR: Failed to read data from hcatalog. task : 1. Details : Error in
RPC response header: AppError
```

Check the complete SQL-MR logs in the AMC or through the ncli sqlmr commands.

### HCatalog Connection Error

```
ERROR: failed to access hcatalog table <tableName>; error connecting to
the hcatalog templeteon server
```

Reason: An invalid server argument was passed. Make sure the servername is correct for the Hive server. This could also be caused by an incorrect port number either being passed via the PORT argument, or in the AMC SQL-H Configuration.

### Function Invoked on Wrong Type of Table

```
ERROR: SQL-MR function LOAD_FROM_HCATALOG failed: Failed to read data
from hcatalog. task : 0. Details : load_from_hcatalog must be invoked on
an empty partitioned table
```

Reason: The function was invoked on a non-fact table in Aster Database.

## Limitations

### Support for Unsecured Hadoop Clusters Only

Note that secured Hadoop clusters are not supported. SQL-H works with unsecured Hadoop clusters only.

### Specifying Partitions by Range

When issuing a query that specifies partitions to return, you must use a list of values (string format). You cannot use a range of values (e.g. WHERE hcat\_table.partition >= '2011-09-01' and hcat\_table.partition\_col <= '2011-09-30').

### Special Character Handling Limitations in HCatalog

The following sections include information special characters handling limitations in HCatalog:

#### Single quote and double quote in partition value

It is not recommended that the partition value in HCatalog contain both single quote (') and double quote ("") characters, because HCatalog does not support the filtering of partitions that include both of these.

#### Backslash

HCatalog does not support backslash (\) as an escape character. It stores a backslash as a regular character.

### **Special characters in partition key**

HCatalog ignores any special characters in the partition key. For example, when using a partition key `\$^`, it is interpreted by HCatalog as simply `$`.



## CHAPTER 9 **Backup and Restoration**

Aster Database Backup enables you to back up and restore individual tables or your entire Aster Database. You initiate backup and restore actions through the Aster Database Backup CLI (command line interface). You can monitor ongoing backup and restoration jobs using the CLI or using the AMC's **Admin > Backup** tab.

Backups can be done at the level of an individual table or for the whole Aster Database. Table-level backups are called "logical backups." Cluster-wide backups are called "physical backups." Backups are online operations, meaning your cluster remains up and can service queries while the backup runs. Both types of backups may be scheduled to occur automatically at a specified interval. Physical backups can be run in an incremental fashion to save space and bandwidth.

See the following sections for details:

- [Introduction to Aster Database Backup](#)
- [Managing the Backup Cluster](#)
- [Physical Backup and Restore](#)
- [Logical Backup and Restore](#)
- [Troubleshooting and Limitations](#)
- [Disaster Recovery Backups](#)

## Introduction to Aster Database Backup

### Overview

Aster Database Backup relies on a Backup Cluster made up of a set of *Backup Nodes*. The Backup Cluster is not an Aster Database; it typically has only a few nodes, and its architecture is not derived from the Aster Database clustering architecture. Each Backup Node is a machine with Aster Database Backup software that enables it to store backed up Aster Database data. You designate one of the Backup Nodes as the *Backup Manager* from which you can manage your backup and restoration actions. (To do this you use the command line tool, `ncluster_backup`, which runs on the Backup Manager.) In addition to its manager activities, the Backup Manager stores data, like any other Backup Node.

When you run a backup, the Aster Database components do the following:

- 1 The Backup Manager first contacts the queen node in Aster Database.
- 2 The queen, in turn, coordinates the backup request with the worker nodes.
- 3 Worker nodes connect directly to the Backup Nodes and stream data in a massively parallel manner to the Backup Nodes.

The number of Backup Nodes does not necessarily correspond to the number of worker nodes. The queen assigns worker nodes to Backup Nodes each time a backup begins, based on the number of worker nodes and the number of Backup Nodes available. During a backup, each Aster Database worker transfers its data directly to its assigned Backup Node. After a backup has been started, the Backup Manager and the Aster Database queen are not part of the backup data path, and so do not become bottlenecks.

Restoration of backup data can also be initiated using the Aster Database Backup Command Line Interface (CLI). During restoration, data is streamed in a massively parallel manner directly from the Backup Nodes to the worker nodes, providing faster restoration.

Table-level restoration is an online operation, meaning it does not interrupt the operation of Aster Database. Physical restoration, on the other hand, requires a restart of Aster Database.

Aster Database Backup also provides disaster recovery capabilities – full backups can be restored to a secondary site that is running the Aster Database database software.

## Backup Philosophy

Aster Database Backup uses a parallelized architecture to offer fast backup and restoration for large data warehouses. Aster Database Backup provides:

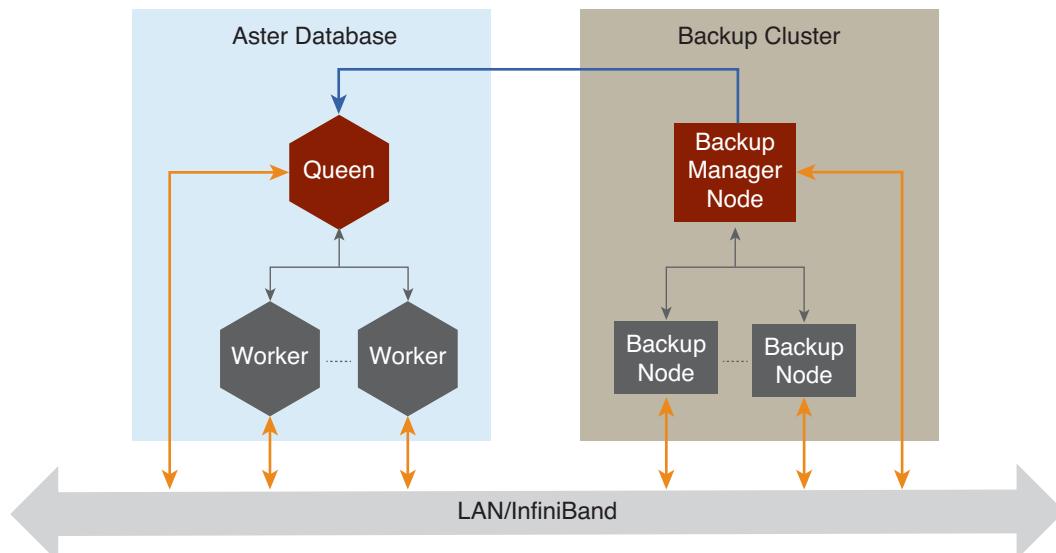
- **Availability:** Data is backed up while the system is operating, to avoid query downtime. Also, table restorations don't require a system restart (physical restorations, however, *do* require a system restart).
- **Flexibility:** A backup can include the full cluster or a single table. Table-level backups are useful when expiring child tables over time (for example, with monthly tables). When older data becomes less frequently accessed or queried, it becomes more cost-effective to migrate it to a permanent archive via the table-level backup option.
- **Scalability:** Multiple concurrent streams (many worker nodes to many Backup Nodes) ensure highly-parallelized backup and restoration for scalable, high-performance operation. Backup Nodes themselves are highly scalable – consistent with the “Always Parallel” mantra of Aster Database. Backup Nodes can linearly scale out to meet changing computational and storage demands.
- **Efficiency:** Physical backups can be done on an incremental basis. Restoration automatically recreates the database from a desired point in time, using incremental and full backups as needed.
- **Fault Isolation:** Backup is an application that is independent of Aster Database. Unlike worker or loader nodes that are dependent on the queen within Aster Database, the Backup Nodes and the Backup Manager are independent applications that are physically and logically separate from Aster Database. Backup copies are intended to maintain an uncorrupted version in the event that the primary Aster Database is compromised. By

isolating backup as an independent application housed on a separate failure domain, Aster Database prevents failure propagation.

- **Manageability:** To minimize the possibility of user error and to save valuable time for database administrators (DBAs), Teradata Aster's design philosophy is to offer ease of use while ensuring full flexibility. Teradata Aster offers the option of either a graphical interface via the Aster Database Management Console (AMC) for simplicity or a command-line environment for more flexibility (e.g., compatibility with scripts). Rich functionality makes life easier for DBAs. For example you can pause and resume backups, schedule backups via a native schedule or external tool (CRON job or third party software), and monitor the progress and history of your backups.
- **Location Independence:** Aster Database enables both local backups (via a LAN) or remote backups (via a WAN) to provide flexibility and improved disaster protection. In the event of a site failure, remote backups can be restored locally at the disaster recovery (DR) site by provisioning a new Aster Database.
- **Cost Efficiency:** Network costs are minimized by compressing data on the worker nodes before it is backed up, therefore minimizing the amount of data transferred and avoiding network congestion. This also applies to data recovery, which restores on the worker nodes from compressed data to minimize network traffic and optimize restoration time objective (RTO) during restores. Disk storage costs are minimized on the Backup Cluster by keeping backup data fully compressed. Of course, Backup Nodes themselves can be inexpensive commodity servers with densely-packed large disk drives to maximize \$/GB savings.

## Backup Architecture

Figure 28: Aster Database Backup Architecture



Aster Database Backup is based on a purpose-built architecture that leverages a set of dedicated Backup Nodes and a Backup Manager to form a “Backup Cluster.” The Backup Cluster architecture can leverage massive parallelism of the Backup Nodes to provide high-performance backup and restoration. For disaster recovery, the Backup Cluster can reside in a location that is geographically separated from the Aster Database. Multiple Aster Databases

can send data to a single Backup Cluster, providing an opportunity to consolidate backups and reduce costs.

Data from Aster Database can be backed up to two types of storage targets:

- **Backup Cluster:** The Backup Nodes themselves can store backups using their direct-attached storage, with hardware-level disk mirroring (e.g. RAID 1) for enhanced data protection. Storage-heavy servers can provide high-density storage at a low cost per terabyte – an important consideration for data volumes associated with data warehousing applications. The Backup Cluster design follows incremental scalability principles similar to those of the Aster Database architecture – more servers can be incrementally provisioned to add backup capacity. Thus, backup storage costs can be managed in a granular manner as data volumes grow. Backup files can subsequently be moved to tapes or VTLs (virtual tape libraries) from Backup Nodes if required by IT or corporate governance policies.
- **Network Storage:** Aster Database Backup also provides the flexibility to store backups on network storage (SAN/NAS) if required by an organization's IT policies. If this option is chosen, Backup Nodes can run backup and restore processes in a massively-parallel manner while using a networked storage array to store backup data.

## Managing the Backup Cluster

Managing the Backup Cluster includes starting, stopping, checking status, storage space and availability and troubleshooting unavailable nodes. Note that upon rebooting a the Backup Manager or a Backup Node, Aster Database Backup software is automatically started.

Starting, stopping, and restarting Aster Database Backup requires logging in as root. If `nBackupNode` and `nBackupMgr` are not in the root path, add them now, or change the working directory to the directory where they reside, which is `/etc/init.d` by default.

### Start the Backup Cluster

- 1 On all Backup Nodes (including the Backup Manager), type:  
`nBackupNode start`
- 2 On the Backup Manager, type:  
`nBackupMgr start`

### Shut down the Backup Cluster

- 1 On the Backup Manager, type:  
`nBackupMgr stop`
- 2 On all Backup Nodes (including the Backup Manager), type:  
`nBackupNode stop`
- 3 Power down all the nodes (including the manager) if needed.

## Restart the Backup Cluster

### Restart the Backup Cluster after machines are powered off:

Power up the Backup Node machines before the Backup Manager machine. Aster Database Backup will start up automatically on all machines upon booting.

- 1 Power up all Backup Nodes.
- 2 Power up the Backup Manager.

### Restart the Backup Cluster without rebooting

- 1 On all nodes (including the Backup Manager) type:

```
nBackupNode restart
```

- 2 On the Backup Manager, type:

```
nBackupMgr restart
```

## Check the Status of the Backup Cluster

- 1 On all Backup Nodes (including the Backup Manager), type:

```
nBackupNode status
```

- 2 On the Backup Manager, type:

```
nBackupMgr status
```

- 3 A status of either Running or Stopped will display.

## Check the Availability and Capacity of the Backup Cluster

Run the Backup Manager CLI (see [Launching the Backup CLI \(page 347\)](#)) for information on running the CLI) and type:

```
show storage
```

This shows all backup nodes along with their used/total storage capacity and current status.

## Troubleshoot an Unavailable Backup Node

Sometimes a Backup Node will become unavailable. When you type `show storage`, its entry will look like the entry for 10.51.4.103 in this example:

```
nCluster Backup> show storage
```

| Checked at: | 2010-Dec-16 15:00:00.364528 |       |      |       |      |
|-------------|-----------------------------|-------|------|-------|------|
| Backup Node | Status                      | Total | Used | Avail | Use% |
| 10.51.4.104 | Available                   | 58.7G | 1.9G | 53.8G | 4    |
| 10.51.4.103 | Unavailable                 |       |      |       |      |
| 10.51.4.102 | Available                   | 58.7G | 1.9G | 53.8G | 4    |
| 10.51.4.101 | Available                   | 58.7G | 1.9G | 53.8G | 4    |
| 10.51.4.100 | Available                   | 58.7G | 2.4G | 53.3G | 5    |

If this happens, first ping the unavailable Backup Node machine to make sure that it's running and reachable:

- If it does not respond to your ping attempt, then restart the unreachable Backup Node machine and, if needed, restore its network connection.
- If it does respond to your ping attempt, then restart the Backup Manager to refresh its view of all nodes. To do this, type:  
`/etc/init.d/nBackupMgr restart`

## Using Aster Database Backup

### Launching the Backup CLI

To begin working with Aster Database Backup, launch its command line interface (CLI). Note that Aster Database Backup must be running in order for the CLI to work. For information on how to start, stop and manage Aster Database Backup, see “[Managing the Backup Cluster](#)” on [page 345](#).

To launch the CLI:

- 1 Open a command shell on any machine that has Aster Database Backup installed.
- 2 Log in as beehive.
- 3 If `ncluster_backup` is not in your path, add it now, or change the working directory to the executable directory. By default, the executables are installed in:  
`$ cd /home/beehive/bin/exec`
- 4 Type `ncluster_backup`, passing the `-h` flag followed by the IP address or DNS name of the Backup Manager:  
`$ ncluster_backup -h <Mgr-IP>`  
For example, if you are working on the manager itself, you will type  
`$ ncluster_backup -h localhost`

### Aster Database Backup Commands

The basic Aster Database Backup commands are `register`, `unregister`, `start`, `pause`, `resume`, `cancel`, `show`, `schedule`, and `delete`, and `Ctrl-D` to exit.

Table 9 - 2: Aster Database Backup commands

| Command                                            | Description                                                                   |
|----------------------------------------------------|-------------------------------------------------------------------------------|
| <code>help [command name]</code>                   | Prints help for specified command                                             |
| <code>Ctrl-D</code>                                | Quits Aster Database Backup CLI                                               |
| <code>register node [Node-IP]</code>               | Registers the node with IP [Node-IP] with the Backup Manager as a Backup Node |
| <code>unregister node [Node-IP]</code>             | Unregisters the Backup Node with IP [Node-IP] from the Backup Manager         |
| <code>start backup [Queen-IP] physical full</code> | Starts a full physical backup of the cluster with queen IP address [Queen-IP] |

Table 9 - 2: Aster Database Backup commands (continued)

| Command                                                                             | Description                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>start backup [Queen-IP]<br/>physical incremental</code>                       | Starts an incremental physical backup of the cluster with queen IP address [Queen-IP]                                                                                                                               |
| <code>start backup [Queen-IP]<br/>logical [TABLE] in [DB]</code>                    | Starts a logical backup of table [TABLE] in database [DB] in the cluster with queen IP address [Queen-IP]                                                                                                           |
| <code>start restore [Queen-IP]<br/>physical [BACKUPID]</code>                       | Starts a physical restore of the cluster with queen IP address [Queen-IP] from the backup identified by [BACKUPID].                                                                                                 |
| <code>start restore [Queen-IP]<br/>logical [BACKUPID]<br/>as [TABLE] in [DB]</code> | Starts a restore of the logical backup identified by [BACKUPID] as the table [TABLE] in the database [DB] in the cluster with queen IP address [Queen-IP]                                                           |
| <code>pause backup [BACKUPID]</code>                                                | Pauses running physical backup identified by [BACKUPID]. Note that you cannot pause logical backups, and you cannot pause restores.                                                                                 |
| <code>resume backup<br/>[BACKUPID]</code>                                           | Resumes a failed or paused physical backup identified by [BACKUPID]. Note that this command is <i>not</i> available for restores or for logical backups.                                                            |
| <code>cancel backup<br/>[BACKUPID]</code>                                           | Permanently cancels a running backup identified by [BACKUPID]                                                                                                                                                       |
| <code>cancel restore<br/>[RESTOREID]</code>                                         | Permanently cancels a running restore identified by [RESTOREID]                                                                                                                                                     |
| <code>show status backup<br/>[BACKUPID]</code>                                      | Shows status of a running backup identified by [BACKUPID]. The storage and status information of the backup nodes is retrieved at 5 min interval                                                                    |
| <code>show status restore<br/>[RESTOREID]</code>                                    | Shows status of a running restore identified by [RESTOREID]                                                                                                                                                         |
| <code>show backups</code>                                                           | Shows all the backups with details like backup ID number, status, progress, start time, and end time. The show backups command displays backups grouped by type (full physical, incremental physical, and logical). |
| <code>show restores</code>                                                          | As an alternative, you can see backup job status in the AMC's Admin > Backup tab.                                                                                                                                   |
| <code>show schedule<br/>[SCHEUDLEID]</code>                                         | Shows all the restores with details like status, start time and end time                                                                                                                                            |
| <code>show schedules</code>                                                         | Shows the backup schedule identified by [SCHEUDLEID]                                                                                                                                                                |
| <code>show storage [BACKUPID]</code>                                                | Shows all backup schedules                                                                                                                                                                                          |
| <code>show storage</code>                                                           | Shows the storage statistics of the backup identified by [BACKUPID]                                                                                                                                                 |
|                                                                                     | Shows all Backup Nodes along with their used/total storage capacity and their current status. Note that the storage and status information of the backup nodes is retrieved at 5 minute intervals.                  |

Table 9 - 2: Aster Database Backup commands (continued)

| Command                                                                      | Description                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| schedule backup physical [Queen-IP] full [TIME] repeat [F_PERIOD] [I_PERIOD] | Creates a backup schedule for a recurring full or incremental physical backup for the cluster with the queen IP [Queen-IP] to begin at time [TIME] and repeat the full backup every [F_PERIOD], with incremental backups to be run every [I_PERIOD]. The format of [F_PERIOD] and [I_PERIOD] is <number> [h d w] interpreted as number of hours, days or weeks, respectively. |
|                                                                              | To schedule only physical backups (with no incremental backups) specify I_PERIOD as 0d (e.g. schedule backup physical 10.51.3.100 full 2010-12-14T06:30 repeat 1w 0d)                                                                                                                                                                                                         |
| schedule backup logical [Queen-IP] [TABLE] in [DB] at [TIME] repeat [PERIOD] | Creates a backup schedule for a recurring table backup of table [TABLE] in database [DB] to begin at time [TIME] and to be repeated every [PERIOD]. The format of [PERIOD] is <number> [h d w] interpreted as number of hours, days or weeks, respectively.                                                                                                                   |
| delete backup [BACKUPID]                                                     | Deletes the backup data identified by [BACKUPID]                                                                                                                                                                                                                                                                                                                              |
| delete schedule [SCHEDULEID]                                                 | Deletes the backup schedule identified by [SCHEDULEID]                                                                                                                                                                                                                                                                                                                        |

## Help with CLI Commands

To get a list of commands, type `help` at the CLI command line:

```
nCluster Backup> help
```

To print a description of a command, type `help` followed by the command name. For example, for help with the `schedule` command, type:

```
nCluster Backup> help schedule
```

## Issuing Backup Commands from a File

You can optionally place backup commands in a file and use the `-f` option to read commands from it. To do this, first create a text file for your commands. Each command will occupy a single line. This file uses a format like:

```
start backup 10.100.7.100 logical sales_fact in retail_sales
start backup 10.100.7.100 logical customer_dim in retail_sales
start backup 10.100.7.100 logical product_dim in retail_sales
```

Then issue `ncluster_backup` with the `-f` option. The following example issues commands from a file called `nightly_backup.txt`:

```
ncluster_backup -f /home/beehive/myScripts/nightly_backup.txt
```

## Quitting the CLI

Type `Ctrl-D` to quit the Backup CLI.

# Physical Backup and Restore

This section explains how to back up all of the databases and metadata in your cluster using Aster Database's physical backup facility:

- [Introduction \(page 350\)](#)
- [Physical Backup \(page 351\)](#)
- [Physical Restore \(page 358\)](#)

## Introduction

To protect against catastrophic data loss, Aster Database Backup provides an online physical backup and restoration capability. Physical backups can be either full or incremental and can be run concurrently with users running queries and loads. Incremental backups save only the data that has changed since the last full backup. Since only incremental changes are saved, incremental backups generally require less processing, network bandwidth, and storage compared with full backups.

In a full physical backup of an Aster Database, all files necessary for restoring that Aster Database are compressed at the source. Once compressed, the files are copied over the network to a Backup Cluster. This set of files includes all database data files, write-ahead log (WAL) files, configuration files, as well as metadata needed for performing a restore.

Backup data on the Backup Cluster is stored in a compressed format to reduce storage requirements. A typical data compression ratio is about 2.5x, but the actual compression ratio will depend on the characteristics of the data being compressed.

Physical restore provides the inverse operation. Data on the Backup Cluster from either a full backup or an incremental backup is copied over the network and is uncompressed at the destination Aster Database. Sending compressed data over the network improves performance and helps optimize restoration times.

The key features of physical backup are:

- **Online physical backups:** Physical backups can be performed while Aster Database is operational, incurring no system downtime.
- **Online incremental physical backups:** Incremental physical backups can also be performed while Aster Database is operational, incurring no system downtime.
- **Physical restore:** Restore operations automatically restore the full database, taking into account prior incremental and full backups to rebuild the database as it was at the point in time when the designated backup was made, whether it was a full or an incremental backup.
- **Computational scalability:** CPU-intensive operations are executed at the worker nodes, which are expected to be deployed in higher numbers and have higher processing power than Backup Nodes. During a physical restore the data is re-loaded into the system without having to be processed by queen or loader nodes, therefore taking advantage of the higher number of processors available at the worker nodes.

- **Parallelism:** Data is transferred in parallel between worker and Backup Nodes. All worker nodes housing active replicas are used during a physical backup operation and the data generated is distributed across all Backup Nodes. This allows for high-speed transfers that would normally be more constrained by hardware characteristics such as network bandwidth.
- **Load balancing:** The data generated by a physical backup is load-balanced across all Backup Nodes according to their current storage utilization.
- **Backup scheduler:** Physical backups, both full and incremental, can be scheduled to occur automatically at specified intervals.
- **Backup pause/resume:** Physical backups, both full and incremental, can be paused and resumed, allowing bandwidth and database performance to be allocated to the most important tasks, as needed.
- **Backup cancellation:** Physical backups may be cancelled, if desired.

## Physical Backup

This section provides instructions on running and managing physical backups:

- [Examples in This Section \(page 351\)](#)
- [Full Physical Backup \(page 352\)](#)
- [Incremental Physical Backup \(page 353\)](#)
- [Scheduling Physical Backups \(page 354\)](#)
- [Scheduling Logical Backups \(page 355\)](#)
- [Showing the Backup Schedule \(page 356\)](#)
- [Deleting a Backup Schedule \(page 356\)](#)
- [Pausing and Resuming Backups \(page 356\)](#)
- [Canceling Backups \(page 357\)](#)
- [Deleting a Physical Backup \(page 358\)](#)

### Examples in This Section

This section shows a running example where the Aster Database being backed up has queen IP address 10.50.200.100 and the Backup Cluster has a Backup Manager with IP address 10.50.34.225. On the 10.50.200.100 cluster, assume that the cluster currently has the following data: a “marketing” table (with 1 million rows) and a “region” table (with 250K rows).

We will use ACT to view the properties of the Aster Database to be backed up. For information on using ACT, see [ACT Quick Start \(page 60\)](#):

```
$ act -h 10.50.200.100

beehive=> \d
      List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+
 public | region | table | beehive
 public | marketing | table | beehive
(2 rows)
```

```
beehive=> select count(*) from marketing;
      count(1)
-----
      1000000
(1 row)

beehive=> select count(*) from region;
      count(1)
-----
      250000
(1 row)
```

## Full Physical Backup

### Procedure

To make a full backup of an Aster Database:

- 1 From any workstation that has Aster Database Backup installed, run the `ncluster_backup` CLI, passing the IP address or DNS name of the Backup Manager:  
`$ ncluster_backup -h <Backup Manager IP>`
- 2 Type the `start backup` command with the IP address of the Aster Database queen (the source of the backup) and the argument `physical full`. If your queen node has multiple IP addresses, use the IP address which is connected to the cluster's private network.  
`nCluster Backup> start backup <Queen IP> physical full`
- 3 Use the `show backups` command to check the progress of your backup. The `show backups` command displays backups grouped by type (full physical, incremental physical, and logical). Alternatively, you can check the progress in the AMC's **Admin > Backup** tab.

### Example

- 1 Start Aster Database Backup:

```
$ ncluster_backup -h 10.50.34.225
```

- 2 Run the full physical backup:

```
nCluster Backup> start backup 10.50.200.100 physical full
```

The Backup Manager returns a Backup ID number like:

```
Full Backup ID: 1224465480754100
```

The backup ID is a unique identifier you can use to refer to the backup. For example, when restoring from a full physical backup, use the backup ID to identify the backup to be restored.



**Tip!** The Aster Database Backup identifies both logical and physical backups by assigning them a backup ID, as they are created. The backup ID is a unique identifier generated by Aster Database Backup. It is useful to keep a log of the backup ID for each backup you perform, along with the database, date/time and any information needed for a logical backup (table, parent table(s), schema, timeframe, etc.) That way, should the need arise to restore a database or a table, you can easily determine which backup ID will provide the desired result.

- 3 While a backup is running, the status of the backup and its incremental progress can be viewed in the CLI by using the `show backups` command (or in the AMC's Admin > Backup tab). To use the CLI:

```
nCluster Backup> show backups
```

```
Full Backups
Backup Id      Cluster IP    Status     %Done      Start Time      End Time
1224465480754100 10.50.200.100 Running  14.35%    2008-09-20 01:18
```

When the backup is finished, typing `show backups` again displays the following:

```
nCluster Backup> show backups
```

```
Full Backups
Backup Id      Cluster IP    Status     %Done      Start Time      End Time
1224465480754100 10.50.200.100 Succeeded 100.00%  2008-09-20 01:18  2008-09-20 01:21
```

In the above example, the backup was first in the "running" state and was 14.35% complete. Later, after the user typed `show backups`, the status had changed to succeeded. The backup had completed and showed 100% completion and the resulting End Time.

The `show backups` command can be useful to see what backups exist on the Backup Cluster and identify their state. For backups that are still in progress, incremental progress can also be tracked.

## Incremental Physical Backup

Incremental backups are performed relative to the last full backup. In the example below, the incremental backup ID 1224466083893978 is performed relative to the full backup ID 1224465480754100.

### ***Procedure:***

To make an incremental backup of an Aster Database:

- 1 Open a command shell on the Backup Manager on the Backup Cluster.
- 2 Use the CLI to connect to the Backup Manager:

```
$ ncluster_backup -h 10.50.34.225
```

- 3 Type the `start backup` command with the IP address of the Aster Database queen (the source of the backup) and the argument `physical incremental`. That is:

```
nCluster Backup> start backup <Queen IP> physical incremental
```

### ***Example:***

- 1 Launch the CLI:

```
$ ncluster_backup -h 10.50.34.225
```

- 2 Start the incremental backup:

```
nCluster Backup> start backup 10.50.200.100 physical incremental
```

The Backup Manager prints the backup's ID:

```
Incremental Backup ID: 1224466083893978
```

As with full physical backups, the backup ID for an incremental physical backup is a unique identifier. This ID can be used to refer to the incremental backup for future operations, such

as restoring from the physical incremental backup. Progress can also be tracked using the `show backups` command or in the AMC's Admin > Backup tab.

### **Example**

1 Launch the CLI:

```
$ ncluster_backup -h 10.50.34.225
```

2 Issue the `show backups` command:

```
nCluster Backup> show backups
...
Incremental Backups
Backup Id      Cluster IP      Status      %Done      Start Time      End Time
1224466083893978  10.50.200.100  Running    42.86%    2008-09-20 01:28
...
```

And later, when issuing the `show backups` command:

```
nCluster Backup> show backups
Incremental Backups
Backup Id      Cluster IP      Status      %Done      Start Time      End Time
1224466083893978  10.50.200.100  Succeeded  100.00%  2008-09-20 01:28  2008-09-20 01:30
```

## **Scheduling Physical Backups**

Aster Database Backup provides a backup scheduler that enables users to schedule backups automatically at a specified date/time. Aster Database Backup scheduling is used to trigger physical as well as logical backups.

As a best practice, Teradata Aster recommends first trying out a physical or logical backup via a one-time backup. A successful outcome helps avoid any inadvertent user error (e.g. mistyping an IP address, database name, or table name). Once the one-time backup is successful, the administrator can convert this into a schedule.

### **Usage**

```
schedule backup physical queen_address full time repeat full_period
incremental_period
```

where

- *queen\_address* is the IP address or host name of your Aster Database queen.
- *time* is the time of the first backup. The format of *time* is YYYY-MM-DDTHH:MM, interpreted as UTC time.
- *full\_period* is the interval at which full backups should be started. The format of *full\_period* is <number>[h|d|w] interpreted as number of hours, days or weeks, respectively.
- *incremental\_period* is the interval at which incremental backups should be started. The format of *incremental\_period* is <number>[h|d|w] interpreted as number of hours, days or weeks, respectively.

### **Example**

Schedule a physical backup:

```
$ ncluster_backup -h 10.50.34.225
```

```
nCluster Backup> schedule backup physical 10.50.200.100 full 2008-07-31T06:30 repeat 2w 2d
```

Upon successfully adding a schedule, Aster Database Backup returns the backup's schedule ID:

```
Schedule ID: 1224799752
```

The above example schedules a physical backup of the Aster Database with queen IP address 10.50.200.100. The scheduled backup will start at 6:30 AM on July 31, 2008 UTC (Coordinated Universal Time). The repeat keyword is used to indicate if the backup activity is recurring. In the above example, a full backup will be taken at the start time and would be automatically taken once every two weeks (represented by "2w"). An incremental backup will be taken every other day (represented by "2d").

#### ***Scheduling a physical backup without incrementals:***

To schedule a physical backup to take only full backups and *exclude incremental backups* from the schedule, simply specify repeat time for incremental backups as 0 days. For example:

```
nCluster Backup> schedule backup physical 192.168.1.10 full 2010-08-10T10:15 repeat 1w 0d
```

### **Scheduling Logical Backups**

The command to schedule logical backups is similar to that for physical ones:

#### ***Usage:***

```
schedule backup logical queen_address table_name in db_name at time
repeat period
```

where

- *queen\_address* is the IP address or host name of your Aster Database queen.
- *table\_name* is the (optionally schema qualified) name of the table to be backed up.
- *db\_name* is the name of the database to be backed up.
- *time* is the time of the first backup. The format of *time* is YYYY-MM-DDTHH:MM, interpreted as UTC time.
- *period* is the interval at which backups should be started. The format of *period* is <number>[h|d|w] interpreted as number of hours, days or weeks, respectively.

#### ***Example:***

Schedule the backup:

```
$ ncluster_backup -h 10.50.34.225
```

```
nCluster Backup> schedule backup logical 10.50.200.100 testTable in
TestDB at 2008-07-31T01:00 repeat 1d
```

Upon successfully adding a schedule, Aster Database Backup returns the backup's schedule ID:

```
Schedule ID: 1224799753
```

The above example schedules a logical backup (refer to "[Logical Backup and Restore](#)" on [page 361](#)) of table testTable from database TestDB on queen 10.50.200.100 starting at 1:00 AM on July 31, 2008 (UTC). A logical backup will be taken at that time and will be taken every day (represented by "repeat 1d").

## Showing the Backup Schedule

You can print the backup schedule using the “show schedule” command. Type “show schedule” with no arguments to see all schedules, or type “show schedule” followed by a schedule ID to show only that schedule.

### **Example:**

Type the “show schedule” command for a particular *physical* backup:

```
$ ncluster_backup -h 10.50.34.225
nCluster Backup> show schedule 1224799752

Schedule Information for Physical Backup
Identifier    Cluster IP      Start Time          Repeat (Full)   Repeat (Incr)
1224799752    192.168.1.10  2008-07-30 23:30:00  2w              2d
```

Type the “show schedule” command for a particular *logical* backup:

```
nCluster Backup> show schedule 1224799753

Schedule Information for Logical Backup
Identifier    Cluster IP      Start Time          Repeat      Database Name  Table Name
1224799753    192.168.1.10  2008-Jul-30 18:00:00  1d          TestDB        testTable
```

## Deleting a Backup Schedule

You can delete (or “unschedule”) a backup schedule using the `delete schedule` command. Type `delete schedule` followed by the backup ID.

### **Example:**

```
$ ncluster_backup -h 10.50.34.225
nCluster Backup> delete schedule 1224799753
Now, when we look for the schedule again...
nCluster Backup> show schedule 1224799753
...it is gone:
Error executing request: Invalid request
```

## Pausing and Resuming Backups

While backups can execute concurrently with queries and loads, running the backup still consumes system resources. In some cases, you may want to pause the current backup so the system can allocate all its resources to queries or data loading. The pause and resume commands let you do this. Note that only physical backups (either full or incremental) can be paused. Logical backups and restores *cannot* be paused.



---

**Tip!** You can also pause and resume a backup job using the AMC's Admin > Backup tab

---

To pause a backup from the CLI, type “pause backup” followed by the backup ID. To resume, type “resume backup” followed by the backup ID.

Pausing a backup stops the backup until you resume it. Any data that has already been transferred from the subject Aster Database to the Backup Cluster will not be transferred again. Incremental work that the backup performed prior to being paused is preserved.

***Example of Pausing a Backup:***

```
$ ncluster_backup -h 10.50.34.225
nCluster Backup> start backup 10.50.200.100 physical full
```

The Backup Manager prints the backup's ID:

```
Full Backup ID: 1224475037516924
...
```

Use the pause command to pause...

```
nCluster Backup> pause backup 1224475037516924
...
nCluster Backup> show backups

Full Backups
Backup Id      Cluster IP      Status    %Done   Start Time      End Time
1224475037516924  10.50.200.100  Paused   21.43%  2008-09-20 03:57
```

To resume the backup, use the “resume” command.

***Example of Resuming a Backup:***

```
$ ncluster_backup -h 10.50.34.225

nCluster Backup> resume backup 1224475037516924
nCluster Backup> show backups

Full Backups
Backup Id      Cluster IP      Status    %Done   Start Time      End Time
1224475037516924  10.50.200.100  Running  42.86%  2008-09-20 03:57
```

## Cancelling Backups

To cancel a backup that is in progress, use the cancel command. Type `cancel backup` followed by the backup ID. In this example, the full physical backup 1224474338327362 is initially running and then cancelled.



**Tip!** You can also cancel backup and restoration jobs using the AMC's Admin > Backup tab.

***Example:***

```
$ ncluster_backup -h 10.50.34.225

nCluster Backup> show backups

Full Backups
Backup Id      Cluster IP      Status    %Done   Start Time      End Time
1224474338327362  10.50.200.100  Running  14.83%  2008-09-20 03:45

nCluster Backup> cancel backup 1224474338327362
nCluster Backup> show backups
```

Full Backups

| Backup Id        | Cluster IP    | Status    | %Done  | Start Time       | End Time         |
|------------------|---------------|-----------|--------|------------------|------------------|
| 1224474338327362 | 10.50.200.100 | Cancelled | 15.83% | 2008-09-20 03:45 | 2008-09-20 03:47 |

Unlike a paused backup, a cancelled backup cannot be resumed. Once a backup is cancelled, it is permanently cancelled. You can cancel both “running” and “paused” backups.

## Deleting a Physical Backup

The storage used for a physical backup can be reclaimed by issuing a `delete backup` command. After the delete command finishes, the data associated with the given physical backup will have been removed from all Backup Nodes.

Before you can delete a physical backup, you must delete all its subsequent incremental backups. That is, if there is an incremental backup with a timestamp later than the backup you are trying to delete, your attempt to delete will fail.

### Usage:

```
delete backup <backup id>
```

### Example:

```
nCluster Backup> show backups
```

Full Backups

| Backup Id        | Cluster IP    | Status    | %Done   | Start Time       | End Time         |
|------------------|---------------|-----------|---------|------------------|------------------|
| 1224465480754100 | 10.50.200.100 | Succeeded | 100.00% | 2008-09-20 01:18 | 2008-09-20 01:21 |

Incremental Backups

| Backup Id        | Cluster IP    | Status    | %Done   | Start Time       | End Time         |
|------------------|---------------|-----------|---------|------------------|------------------|
| 1224466083893978 | 10.50.200.100 | Succeeded | 100.00% | 2008-09-20 01:28 | 2008-09-20 01:30 |

...

```
nCluster Backup> delete backup 1224466083893978
```

```
nCluster Backup> delete backup 1224465480754100
```

## Physical Restore

Use the `start restore` command of `ncluster_backup` to restore Aster Database from a physical backup. You can carry out a physical restore from a full or incremental backup. When you type `start restore`, the backup files are copied from the Backup Cluster to Aster Database and uncompressed. All data in Aster Database is replaced with the data from the backup. Any new data, databases, and tables saved *since the last backup* are lost during the restoration. See the following sections:

- [Restore Procedure \(page 358\)](#)
- [Restore Example \(page 360\)](#)

### Restore Procedure



**Notice!** When you perform a physical restore, *all databases* on the cluster are overwritten with the data from the backup! You cannot undo the restore.

To perform a physical restore, follow the steps below:

- 1 Before restoring a physical backup, perform a **Soft Shutdown** of the target Aster Database. The target Aster Database is the cluster to which you intend to apply the restore.

To do this, working as the root user at the queen command line, type the command:

```
# ncli system softshutdown
```

- 2 Then clean all data off of the target Aster Database by running the following command:

```
# ConfigureNCluster.py --clean_data
```

- 3 Perform a **Soft Startup** of Aster Database:

```
# ncli system softstartup
```



**Note:** As a prerequisite for Restore, the target nCluster should be clean before restoring the data from a physical full backup. This prerequisite is also applicable before performing restore from a physical incremental backup.

- 4 Activate the Aster Database cluster.

- a To do this, point your web browser to the Aster Database Management Console (AMC), located at `http://<nCluster_queen_IP>` by default. Log in with the superuser login (db\_superuser by default) and password. The dashboard appears.
- b Using the menus at the top of the screen, navigate to **Admin > Cluster Management**.
- c Click the **Activate Cluster** button.

- 5 Make sure your Aster Database has the same version number as the cluster from which the backup was made. Cross-version restores are not supported.

- 6 Make sure your Aster Database has the *same partition count* as the backup you will restore. If the cluster is active and no administrator has performed a partition split since the backup was taken, then the partition count is the same.

- 7 Run `ncluster_backup` and type `show backups` to list the available backups. Note the ID number of the backup you wish to restore. Choose your backup based on its timestamp, not based on whether it is a full or incremental backup. Whether the backup is full or incremental does not matter; Aster Database restores the cluster to its state as of the timestamp you choose.

- 8 Type `start restore` followed by the backup ID.

- 9 Type `show restores` to check the status of the restore job. Once it shows a status of Succeeded, proceed to the next step.

- 10 Soft restart the cluster by logging in to the queen as root and soft restarting the queen using the command:

```
# ncli system softrestart
```

- 11 Activate the cluster by going to the AMC's **Admin** tab again and clicking on **Activate Cluster**. This will incorporate all servers and also replicate the data to restore the original replication factor.

## Restore Example

This section continues with the running example from the last section. In this example, a restore is performed from a full physical backup onto an Aster Database with queen IP address 10.50.98.100.

Restoring back to the original Aster Database where the backup was taken just requires specifying the original Aster Database queen IP address. The same command line in the Aster Database Backup CLI is used to restore from either an incremental backup or full physical backup. The only difference is different backup IDs will be used.

This example restores from full physical backup ID 1224465480754100. Once the restore has started, the worker nodes on the Aster Database being restored will turn passive (blue in the AMC). They will remain in that state during the entire restoration operation.

When the restore is done, the cluster needs to be restarted. This can be done by using a Soft Restart as described at the end of this example.

```
$ ncluster_backup -h 10.50.34.225

nCluster Backup> show backups

Full Backups
Backup Id      Cluster IP      Status      %Done      Start Time      End Time
1224465480754100  10.50.200.100  Succeeded  100.00%  2008-09-20 01:18  2008-09-20 01:21

Incremental Backups
Backup Id      Cluster IP      Status      %Done      Start Time      End Time
1224466083893978  10.50.200.100  Succeeded  100.00%  2008-09-20 01:28  2008-09-20 01:30

Logical Backups
Backup Id      Cluster IP      Status      Start Time      End Time
nCluster Backup> start restore 10.50.98.100 physical 1224465480754100
Restore ID: 1224467145960081
```

The status of a restore can be viewed using the `show restores` command. When the backup is complete, it will show status `Succeeded`.

```
$ ncluster_backup -h 10.50.34.225

nCluster Backup> show restores

Physical and Logical Restores
Restore Id      Cluster IP      Status      Start Time      End Time
1224467145960081  10.50.98.100  Running    2008-09-20 01:45
nCluster Backup>
...
nCluster Backup> show restores
```

```
Physical and Logical Restores
Restore Id      Cluster IP      Status      Start Time      End Time
1224467145960081  10.50.98.100  Succeeded  2008-09-20 01:45  2008-09-20 01:47
```

Upon completion of a physical restore, the Aster Database being restored needs to be restarted. Restart the cluster by logging in to the queen as root and soft restarting the queen using the command:

```
# ncli system softrestart
```

After a Soft Restart, the servers will go through a sequence of states, eventually coming up in the AMC as **Prepared**. When a node restarts, it may pass through the states of **New** to **Preparing** to **Upgrading** to **Prepared**. This is normal. At this point, there is one copy of the newly restored data. To restore the replication factor on the cluster, go to the AMC's **Admin** tab again and click on **Activate Cluster**. This will incorporate all servers and also replicate the data to restore the original replication factor.

## Logical Backup and Restore

This section shows how to make and manage logical (table-level) backups in Aster Database.

- [Introduction to Logical Backup \(page 361\)](#)
- [Table-Level Backup \(page 363\)](#)
- [Table-Level Restore \(page 365\)](#)

### Introduction to Logical Backup

Logical backup is a feature of Aster Database that allows a user to take a consistent snapshot of a table in the system and transfer its contents and associated metadata to a set of Backup Nodes. Data is extracted in parallel from all the worker nodes and sent directly to the Backup Nodes, allowing for high-speed data transfers. Because all the metadata associated with a table is also extracted, a logical restore operation can later be performed to reconstruct the table exactly as it was at the time the logical backup was taken.

There are several key differences between logical and physical backups. One is related to the data format used. While a physical backup preserves the native format used by the Aster Database, a logical backup extracts the data in a non-proprietary text format. Another important difference between physical and logical backups is related to granularity. Physical backups work at the cluster level, while logical backups are used to preserve individual tables. These two backup options complement each other.

A logical backup is also different from a data export operation. First, logical backups include metadata associated with a table – the SQL statements needed to recreate the table, ownership information, associated indexes, table and column constraints, etc. In essence, logical backups include everything that is needed to re-create the table exactly as it was before. In contrast, bulk export operations just extract the contents of a given table. Second, logical backup operations move data directly from worker nodes to Backup Nodes, keeping both queen nodes and loader nodes out of the critical data path. In contrast, bulk export passes data through queen or loader nodes before the exported data reaches the end destination.

### Key features

These are the key features of logical backup:

- **Parallelism:** Data is transferred in parallel between worker and Backup Nodes. All worker nodes housing active replicas are used during a logical backup operation and the data generated is distributed across all Backup Nodes. This allows for high-speed transfers that

would normally be more constrained by hardware characteristics such as network bandwidth.

- **Load balancing:** The data generated by a logical backup is load-balanced across all Backup Nodes according to their current storage utilization.
- **Data compression:** Data is compressed using standard algorithms before leaving the worker nodes. This reduces the storage requirements at the Backup Nodes and can also improve performance significantly in configurations where the time taken by a logical backup operation is dominated by data transfer (e.g. network bandwidth bottleneck). During a logical restore operation, the compressed data is first transmitted back to the worker nodes, where it is then decompressed.
- **Computational scalability:** All CPU-intensive operations – including data compression and decompression – are executed at the worker nodes, which are expected to be deployed in higher numbers and have higher processing power than Backup Nodes. As already mentioned, during a logical restore the data is re-loaded into the system without having to be processed by queen or loader nodes, therefore taking advantage of the higher number of processors available at the worker nodes
- **Backup pipelining:** Data is extracted, compressed, and transferred from worker nodes to Backup Nodes in a pipelined fashion, using minimal disk space at the worker nodes. This allows logical backup operations to be performed even in scenarios with high disk utilization at worker nodes.
- **Online backup & restoration:** Both logical backup and logical restore operations can be performed while Aster Database is operational, incurring no system downtime. Replication occurs as part of the restore operation.

## Use cases

Logical backups can be used to create more frequent backups of specific tables in an Aster Database system. For instance, consider a deployment with 100 TB of non-replicated data distributed across a large Aster Database system that has most (e.g. 80%) of the disk space taken by raw fact tables and less than 20 percent by DIMENSION and aggregated summary tables. Due to the size of the system, an administrator may decide to perform physical backups once every month while taking logical backups of important DIMENSION and summary tables every week. Note that a physical backup is the method used to protect a complete Aster Database.

In case of more severe budgetary constraints, the administrator may even decide to avoid physical backups altogether and only protect DIMENSION and summary tables through logical backups, reducing storage requirements at the expense of leaving fact tables unprotected.

Logical backups can also be used as a form of data archiving. For instance, assume that a fact table is stored in a time-based hierarchy created using autopartitioning, with a different child partition containing the data produced each month. To minimize storage requirements in the cluster, one may decide to "evict" old data from the cluster – for instance, all the records created over two years ago – to accommodate newer data. One way to do this would be to DETACH the child partition(s) with the old data and take a logical backup of each resulting monthly table before it is evicted. Then you could execute a logical restore operation whenever

the data is needed again (e.g. ad hoc historical query, management request, legal response, regulatory compliance audit, etc). The restored table could be used as a standalone table or reATTACHED to the parent table as a partition, depending on the requirements of the project.

## Limitations

The inherent characteristics of logical backups create some limitations, as mentioned below:

- 1 Incremental logical backups are not supported – a logical backup can only be used to create a complete copy of a table.
- 2 Restore formatting overhead – while data blocks associated with a physical backup can be readily used after being transferred back into the Aster Database being restored, data in a logical backup needs to be “reloaded.” The data has to be converted back from the text format to the native format used by the Aster Database, which causes some performance overhead. This process is not executed by queen or loader nodes. It occurs automatically on the worker nodes after the data has been transferred.
- 3 Replicating overhead on restore – Note that a logical restore operation is executed as a single modifying transaction, so it is expected to take longer than a logical backup because of the cost of replicating the data as part of transaction commit.
- 4 Logical backups for partitioned tables created using autopartitioning– For partitioned tables created using autopartitioning, you back up the parent table only. All child partitions in the hierarchy are backed up automatically with the parent table. This is a benefit because it simplifies backup and recovery. It is not necessary to back up each child partition individually.
- 5 However, it should be noted that in the unusual case that a backup of *only* a child partition is desired, the child partition cannot be backed up individually unless it is first detached from the parent table using DETACH. It may then optionally be re-ATTACHED following the backup, if desired. For more information on partitioned tables created using autopartitioning, see [Automatic Logical Partitioning \(page 35\)](#).
- 6 Logical backups for parent/child tables created using inheritance– For parent/child tables created using inheritance, Aster Database only allows a single table to be backed up or restored, so if a parent table and all of its  $n$  children tables need to be backed up,  $n+1$  logical backup operations need to be performed. However, this process can be automated using the Aster Database Backup scheduling utility. For more information on parent/child tables created using inheritance, see [Columnar Tables \(page 54\)](#).
- 7 You may reach your file system’s limit for the number of allowed subdirectories. For each backup, one directory is created in the /home/beehive/data/ directory and a symbolic link is created in the /home/beehive/data/.metadata directory. Some file systems such as the EXT3 file system limit the number of subdirectories a directory can contain. If you are using the EXT3 filesystem is used for storing backup data, no more than 30,000 backups can be stored concurrently.

## Table-Level Backup

This section demonstrates how to use the Aster Database Backup CLI to perform logical backup and restore operations. In the examples given below, logical backup operations are

performed against table “testtable” in database “mydatabase” on the Aster Database with queen IP address 10.50.30.100. See the following sections for instructions:

- [Running a Logical Backup \(page 364\)](#)
- [Scheduling a Logical Backup \(page 364\)](#)
- [Full Physical Backup \(page 352\)](#)

## Running a Logical Backup

A logical backup can be started immediately through the `start backup` command. In the example below, a logical backup is started for table “testtable” that exists in the Aster Database system with queen IP address 10.50.30.100. The logical backup receives the backup identifier 1224686772252294, and after the operation has been started, its progress can be monitored using either the `show backups` command (or the more specific syntax in the form, `show status backup 1224686772252294`) or in the AMC’s Admin > Backup tab.

The backup of “testtable” completes successfully in less than one minute, as indicated by the output of the `show backups` command. If any errors had been encountered during the logical backup operation, the status of the backup would have changed to Failed.

### ***Example:***

```
$ ncluster_backup -h 10.50.34.225

nCluster Backup> start backup 10.50.30.100 logical testtable in mydatabase
Logical Backup ID: 1224686772252294

nCluster Backup> show status backup 1224686772252294
Status: Succeeded

nCluster Backup> show backups

Full Backups
Backup Id      Cluster IP      Status    %Done   Start Time   End Time
Incremental Backups
Backup Id      Cluster IP      Status    %Done   Start Time   End Time
Logical Backups
Backup Id      Cluster IP      Status    Start Time       End Time
1224686772252294  10.50.30.100  Succeeded  2008-09-22 14:46  2008-09-22 14:46
```

## Scheduling a Logical Backup

All the commands shown above execute logical backup operations immediately. In addition, logical backup operations can also be scheduled in advance using the `schedule` command. Aster Database provides the same scheduling capabilities for physical and logical backups, with the exception of incremental backups, which apply only to physical backups.

### ***Usage:***

```
nCluster Backup> schedule backup logical <IPADDR> <SCHEMA.TABLENAMES> in <DATABASE> at <TIME> repeat <PERIOD>
```

Format of TIME is YYYY-MM-DDTHH:MM, interpreted as UTC time.

Format of PERIOD is <number> [h | d | w] interpreted as number of hours, days or weeks, respectively.

**Examples:**

```
$ ncluster_backup -h 10.50.34.225
```

```
nCluster Backup> schedule backup logical 192.168.1.10 testTable in
TestDB at 2010-07-31T01:00 repeat 1d
```

Schedules logical backup of table testTable from database TestDB starting at 1:00am on July 31, 2010. A logical backup of table testTable will be taken at the specified time, and will be repeated every day (24 hours). Note that logical backups do not support incremental backups.

### Deleting a Logical Backup

As with physical backups, the storage used for a logical backup can be reclaimed by issuing a “delete backup” command. As with a logical restore operation, the command requires the unique identifier assigned to a logical backup. After the “delete” command finishes, the data associated with the given logical backup will have been removed from all Backup Nodes.

Usage:

```
delete backup <backup id>
```

**Example:**

```
$ ncluster_backup -h 10.50.34.225

nCluster Backup> show backups
...
Logical Backups
Backup Id      Cluster IP      Status      Start Time      End Time
1224686772252294  10.50.30.100  Succeeded  2008-09-22 14:46  2008-09-22 14:46

nCluster Backup> delete backup 1224686772252294
```

## Table-Level Restore

Use the start restore command of ncluster\_backup to restore a table from backup.

**Procedure:**

To perform a table-level restore (logical restore), follow the steps below:

- 1 Make sure of the following:
  - a Make sure your Aster Database has the same version number as the cluster from which the backup was made. Cross-version restores are *not* supported.
  - b The table you are restoring must *not* exist in the database. If it exists, drop it now.
  - c If the table you are restoring had a parent table when it was backed up, that parent table *must* exist in the database. If the parent is missing, create it now.
- 2 Run ncluster\_backup and type show backups to list the available backups. Note the ID number of the backup you wish to restore.

- 3 Type `start restore` followed by the backup ID. Make a note of the Restore ID that appears when the restore begins.
- 4 Type `show restores` to check the status of the restore job, identified by its Restore ID. Once it shows a status of `Succeeded`, the table is restored and can be queried.

**Example:**

As in the backup creation examples earlier in this section, the identifier for the backup of table “`testtable`” is `1224686772252294`. Start the logical restore as shown here:

```
$ ncluster_backup -h 10.50.34.225

nCluster Backup> start restore 10.50.30.100 logical 1224686772252294 as testtable in
mydatabase
Restore ID: 1224687094536897

nCluster Backup> show restores

Physical and Logical Restores
Restore Id      Cluster IP      Status      Start Time      End Time
1224687094536897  10.50.30.100  Running    2008-09-22 14:51

nCluster Backup> show restores

Physical and Logical Restores
Restore Id      Cluster IP      Status      Start Time      End Time
1224687094536897  10.50.30.100  Succeeded  2008-09-22 14:51  2008-09-22 14:51
```

Note that a logical restore operation is executed as a single modifying transaction, so it is expected to take longer than a logical backup because of the cost of replicating the data as part of transaction commit. As shown above, the logical restore is displayed as “Running” until it completes, when its status is changed to “Succeeded.”

## Monitoring and Managing Backups through the AMC

You can optionally use the AMC to check the status of backups. The AMC also allows you to pause, resume and cancel backups. For more information on managing and monitoring backups through the AMC, see the *Teradata Aster Big Analytics Appliance 3H Database Administrator Guide*.

## Troubleshooting and Limitations

Aster Backup can perform only one operation at a time. You may perform one backup or one restore at a time. If you wish to interrupt one operation to perform another, you can use the ‘pause’ feature of Aster Backup. This limitation affects all versions of Aster Backup.

## Disaster Recovery Backups

As part of a disaster recover strategy, you can copy data from the Aster Database Backup Nodes to tape or other storage. This protects against the loss of Backup Nodes, and it allows you to remove older backups from your Aster Database Backup Cluster periodically to free space there, while maintaining copies for historical or legal reasons. To do this, you will need to copy the backup files, either manually or through scripts you create, to your own backup storage system.

# Aster Glossary

This glossary lists terms you will encounter in building and using databases and applications in Aster Database.

## ***ACT***

Aster Database Cluster Terminal (ACT) is the terminal-based SQL query client for Aster Database.

## ***AMC***

Aster Management Console is a web-based administrative console that allows you to monitor and control Aster Database.

## ***Aster Database partitioning***

See [distribution \(of rows\)](#).

## ***Aster Database Data Validator***

Discontinued utility in Aster Database that was used to check data.

## ***Aster Database Loader***

Also written as “ncluster\_loader”, this is Aster Database’s command-line bulk loading utility. Customers are encouraged to use this rather than Bulk Feeder. It provides an alternative to the SQL INSERT statement and offers much better performance and error handling. *Hint!* Do not confuse Aster Database Loader with a [loader node](#).

## ***Aster Database replication***

To provide availability, Aster Database is designed to maintain multiple copies (usually two) of your data. Maintaining these copies is called replication, and, when you create your cluster, you specify a desired replication factor that tells the system how many copies to maintain.

Teradata Aster recommends running Aster Database at a replication factor of two, which means the cluster stores two copies of your data at all times.

Replication is achieved by maintaining a copy of each Aster Database vworker. Recall that, in the distributed architecture of Aster Database, your data is distributed across many vworkers that do the work of retrieving data and performing calculations on the data.

For a given partition, we refer to the vworker holding the active copy of your data as the “active vworker” and the one holding the backup copy as the “passive vworker.” If the active vworker fails, the passive vworker takes over immediately.

#### ***automatic logical partitioning***

The method of partitioning or splitting a large table into child partitions to optimize query performance and simplify table administration. Automatic logical partitioning uses the PARTITION BY RANGE or PARTITION BY LIST clause to create a partitioned table, and is the preferred method of logical partitioning.

#### ***\_bee\_stats database***

See [system tables](#).

#### ***backup queen***

A second queen that you can activate if your queen fails. Usually it's kept in powered-down or STOPPED state with no workers connected, and only powered up when needed. Sometimes called a backup queen.

#### ***balance process***

Also known as “balance processing,” this is the act of making sure active vworkers are evenly distributed on your cluster's hardware, so that each worker node has about the same number of active vworkers as all other worker nodes.

#### ***balance data***

Also known as “balance storage,” this is the act of making sure your Aster Database contains the required number of copies of your data (as specified by your replication factor; usually two copies), and making sure that each replica copy is located on a separate physical worker node from the primary copy.

#### ***BIT***

Outdated name for [ACT](#).

#### ***Bulk Feeder***

Unsupported bulk-loading application; replaced with [Aster Database Loader](#).

#### ***child partition***

In an [automatic logical partitioning](#) schema, a child partition is one partition of the data in the partitioned table.

#### ***child table***

In a [parent-child table inheritance](#) schema, a child table is one partition of the data.

#### ***coordinator***

Old term for the Aster Database [queen](#).

#### ***CSV***

Comma-separated value file format where commas are used as field separators.

## **CTAS**

**CREATE TABLE AS SELECT** This is just a variant of a CREATE TABLE statement with a SELECT subquery that populates the new table with rows from an existing table. These are used very frequently in the Aster Database context to manually repartition a table or to do the ‘transform’ part of ELT. See ELT and ETL.

## ***data locality***

The state of having needed data local to (on the same machine as) an operation or other data. Having data locality is a key factor affecting the efficiency of an operation in an MPP system.

## ***data model***

A database’s structure of tables and columns that determine what form and format the data will be stored in.

## ***DDL***

Data definition language to create and alter database tables.

## ***dimension table***

One of the two main table types in a *star schema*-style database. A row in a dimension table usually describes an item in detail. A dimension table stores unchanging or slowly changing descriptions of the participants in the actions tracked by your database. (The details of each action are recorded in the *fact table*.) For example, product names and descriptions usually live in a dimension table. The volume of data in a dimension table typically grows slowly. Often a dimension table enumerates the set of known values for a particular category.

## ***distributed dimension table***

In Aster Database you can optionally distribute a dimension table by declaring a [distribution key](#) on it.

## ***distributed query planning***

The queen manages the distribution of data in the cluster, prepares top-level, partition-aware query plans, issues queries to vworkers, and assembles the query results. The vworkers, in turn, prepare local query plans and execute the queen’s queries in parallel. The queen structures top-level queries so that little or no data is shipped to the queen until the final phase, when the query results are assembled and sent to the client.

## ***distribution (of rows)***

Distribution of rows (sometimes called “physical partitioning”) means splitting a table’s data across many vworkers in Aster Database to allow scaling. This is a key Aster Database feature. A physical partition is a subset or “slice” of rows stored on a vworker.

Don’t confuse Aster Database distribution with the common data modeling practice of [logical partitioning](#) (a.k.a. [parent-child table inheritance](#) or [automatic logical partitioning](#)), which you can also do in Aster Database. *The difference is this:* distribution happens automatically based on the distribution key you declare using `DISTRIBUTE BY` when you create the table. The distribution is automatic in the sense that you don’t have to declare the boundaries of each partition. Instead, you just say which column (this is called the distribution key column)

provides the values that will be used to define the distribution, and Aster Database chooses boundaries to split up the records. Logical partitioning, on the other hand, requires you to explicitly declare the boundaries of each partition.

### ***distribution key***

When you create a fact (and optionally when you create a dimension table), you specify a distribution key that determines how Aster Database will physically distribute that table's data across the cluster. The distribution key specifies which column's value will be evaluated to determine its location in the cluster. If the table has a primary key defined, then the distribution key must be one of the columns from the primary key. The column you choose as your distribution key must be of a datatype allowed for use as a distribution key.

### ***ELT***

Aster's better alternative to the long-standing datawarehousing practice of ETL (extract, transform, and load). In Aster Database it's usually much better to extract, load, and *only then* transform, because you can use the computing power of the cluster to carry out the data transformations. The main tools for performing such transformations are the CTAS command and SQL-MapReduce transformation functions that you write.

### ***ETL***

The long-standing datawarehousing practice for loading data into the warehouse. ETL stands for extract, transform, and load. By 'transform', we mean the reformatting of the data that you must do to ensure consistent and correct data representation in the warehouse. In Aster Database, we prefer to follow the ELT approach to loading, rather than ETL.

### ***fact table***

One of the two main table types in a *star schema*-style database. In a star schema, the fact table is usually the largest table in the database and records the minute-to-minute actions that your database was built to track. (The job of storing the more detailed information about the actions' participants is delegated to a set of *dimension tables*.) In the fact table, each row usually represents an action or movement, such as a sales transaction or a web pageview. Because of this, the volume of data in a fact table tends to grow fast.

A fact table contains two types of columns: columns that contain facts (say, timestamp and price of a sale) and columns that are foreign keys to the dimension tables (links to the rows, for example, that describe the product sold and the customer who bought it). Note that Aster Database does not enforce referential constraints. Foreign keys are used mainly for joining tables. The effective primary key of a fact table is usually a composite of more than one column.

You create a fact table in Aster Database with the CREATE FACT TABLE command, and you must distribute each fact table by declaring one of its columns to be its distribution key, using the keyword DISTRIBUTE BY.

### ***foreign key***

The column that is used to join a fact table with a dimension table. Aster Database does not enforce referential constraints. Foreign keys are used mainly for joining tables.

## ***Hadoop***

Apache Hadoop is an open source platform for storing and managing big data. Teradata Aster provides SQL-H to enable business users to access the Hadoop data from Aster Database directly. Aster Database manages communication with Hadoop nodes through SQL-H to read data for SQL queries and SQL-MR functions.

## ***hash distribution***

See [physical partitioning](#).

## ***HCatalog***

HCatalog is the table and storage management service for data stored in Apache [Hadoop](#).

## ***Hive***

An open-source SQL layer for Hadoop. It is not compliant with SQL-92 and supports none of the SQL guarantees.

## ***ICE***

The InterConnect Executable process in Aster Database. This is the Aster Database service responsible for finding and shuffling partitions of data between vworkers. For example, if the `users` table is distributed across many partitions and you run the query, `SELECT * FROM myusers`, then ICE collects the rows from all the partitions.

## ***imbalanced***

Undesirable cluster state that you should fix by running either a [balance data](#) or [balance process](#). This typically means you have one worker node with more than the desired number of active vworkers running on it, or you have a single worker node hosting both an active vworker and its corresponding replica vworker.

## ***in-database applications***

Aster Database's in-cluster, in-database applications let you inject user functions (applications) into the data flow at the lowest levels. For many applications, Aster Database is superior to other distributed computing frameworks because Aster Database provides better tools for data manipulation (partitioning, sorting, and the like), as well as process management and workload management.

## ***incorporate***

Outdated term for [balance data](#).

## ***JDBC***

Standard Java API that allows clients to access a database. Aster Database offers a JDBC driver.

## ***list partitioning***

See [logical partitioning](#).

## ***loader node***

An optional node in Aster Database that is specialized in loading data. Normally, you can route all loading directly through the queen, but for high-volume loading requirements, you

can deploy loader nodes to increase loading capacity. When using loader nodes, you initiate the loading using the ncluster\_loader utility, which communicates with the queen. The queen then delegates loading to the loader nodes, which load data into the appropriate vworkers in parallel. You can also force the use of a particular loader node.

### ***logical partition***

A [child table](#) or [child partition](#) and its data.

### ***logical partitioning***

Splitting one large table into smaller logical pieces for faster performance and easier management. This is done via [automatic logical partitioning](#) (preferred) or [parent-child table inheritance](#) (supported for backward compatibility) and is a common database practice as well as a popular feature of Aster Database.

Each partition is created as a child partition of the single partitioned table. The top level table is normally empty, there to represent the data set. Some logical partitioning designs contain multiple generations of partitions. For example, you might have a schema in which table `sales_2008` has yearly child partitions `sales_2008_01` through `sales_2008_12`, and each yearly child partition has daily child partitions like `sales_2008_01_01` through `sales_2008_01_31`.

Don't confuse logical partitioning with the more automatic [physical partitioning](#) feature of Aster Database.

For clarity when discussing logical partitioning in this document, we avoid the term "logical partition," and instead use the more explicit terms [child table](#) or [child partition](#).

### ***machine***

See [node](#).

### ***materialized projection***

A relatively narrow table that contains a copy of a group of columns that are commonly accessed together. A materialized projection usually contains a subset of the columns of a wider table and is created to allow queries to run faster.

### ***nc\_ tables***

see [system tables](#).

### ***NIC bonding***

Network link aggregation that allows you to combine multiple network interface cards to support a common connection for better performance.

### ***node***

In the cluster, a node is a server machine that hosts vworkers, a loader, or a queen. Typically a node is a physical machine, but if you've installed your cluster on VMware or in the cloud, then it's a virtual machine. In Aster Database, each node has a designated role as a queen node, worker node, loader node.

### ***node splitting***

See [partition splitting](#).

### ***ODBC***

A standard API that allows clients to access a database. Aster Database offers an ODBC driver.

### ***Optimized Transport***

A massively parallel communication transport mechanism that enables dynamic [repartitioning](#) of data.

### ***parent-child table inheritance***

An older method of splitting a large table into child tables to optimize query performance using the INHERITS keyword. This approach has been replaced by the preferred [automatic logical partitioning](#).

### ***partition***

See [physical partition](#) or [logical partition](#). For clarity, we avoid using the unqualified term “partition” in this document and instead say “[child table](#)” or “[child partition](#)” for a logical partition, or “[physical partition](#)” for a partition that Aster Database maintains automatically based on a distribution key.

### ***partition count***

Each worker node in Aster Database contains a number of vworkers. The total number of vworkers in the cluster is the “partition count” of the cluster.

### ***partition splitting***

The act of increasing the number of vworkers in your Aster Database. Having the appropriate ratio of CPU cores to vworkers ensures efficient use of your workers’ computing power. As your cluster grows and you add more worker machines, it eventually makes sense to increase the total number of vworkers in order to maintain a good ratio. Contact Teradata support to find out the proper CPU core/vworker ratio for your hardware.

Don’t confuse [repartitioning](#) with partition splitting. Repartitioning happens to rows inside a query, and does not involve changing the physical location of rows on disk. In repartitioning, only the location of an in-memory copy of the row is changed. Partition splitting, on the other hand, ‘permanently’ moves some rows to a different vworker for storage.

### ***partitioning***

See [logical partitioning](#) or [distribution \(of rows\)](#). The term “partitioning” is ambiguous.

### ***passive coordinator***

Outdated term for [backup queen](#).

### ***physical partition***

See [vworker](#).

### ***physical partitioning***

Outdated term. See [distribution \(of rows\)](#).

### ***primary interface***

The Ethernet NIC that the Aster Database administrator designated as the main networking interface for cluster communications. This is specified by interface name and is often *eth0*.

### ***primary queen***

When discussing the queen and the [backup queen](#), we refer to the currently operating queen as the “primary queen”.

### ***queen***

The queen node is the Aster Database coordinator, distributed query planner, distributed query coordinator, and keeper of the data dictionary and system tables. The queen is responsible for cluster, transaction, and storage management. The queen handles software delivery to all nodes. See also [distributed query planning](#).

### ***range partitioning***

See [logical partitioning](#).

### ***repartitioning***

The act of reshuffling the rows of a distributed table to nodes on the cluster where they are needed for a join or aggregation. Repartitioning is frequently a prerequisite step for query execution in which the data required for a join is laid out as though it were distributed by the attribute/expression in the join or the aggregation. For example, when you run `SELECT column-a FROM foo GROUP BY column-a`, if *column-a* is not the distribution key of *foo*, then Aster Database must repartition *foo* so that, for the duration of this operation, it's distributed on *column-a*.

Don't confuse repartitioning with [partition splitting](#). Repartitioning happens to rows inside a query, and does not involve changing the physical location of rows on disk. In repartitioning, only the location of an in-memory copy of the row is changed. Partition splitting, on the other hand, 'permanently' moves some rows to a different vworker for storage.

### ***replicate***

The act of updating the replica of a given piece of data when that piece of data changes. With each change in a vworker's data, Aster Database ensures that the vworker's replica gets a record of the change. See [Aster Database replication](#).

*Tip!* The term “replica” also arises in the case of a [replicated dimension table](#). Don't confuse the two.

### ***replicated dimension table***

A dimension table whose entire contents are copied to all vworkers for faster lookup. This is the default behavior of a dimension table in Aster Database, or you can include the clause `DISTRIBUTE BY REPLICATION` in your `CREATE TABLE` statement to create a replicated dimension table.

*Good to know:* Don't confuse replicated dimension tables with [Aster Database replication](#)! They are *not* closely related. What's being replicated in Aster Database replication are vworkers

(sometimes called “partitions”) whereas what’s being replicated in a replicated dimension table is the whole contents of the table.

### ***replication***

See [Aster Database replication](#).

### ***replication factor (goal)***

Also written as “RF(g)”, this is the desired number of copies of data to be kept in Aster Database. This is almost always 2. This is specified at installation time and can be changed. This setting is stored on the queen as /home/beehive/config/goalReplicationFactor.

### ***replication factor (current)***

Also written as “RF(c)”, this is your cluster’s current replication factor. RF(c) is the replication degree of *the partition with the lowest replication degree in the cluster*. In other words, if one partition in the cluster has lost its replica, meaning its current replication degree has fallen to 1, then the current replication factor of your cluster is 1. When RF(c) falls below RF(g), the AMC alerts you that you need to take action to restore your cluster’s replication factor.

### ***RF***

See [replication factor \(current\)](#) and [replication factor \(goal\)](#).

### ***schema***

Logical subdivision of a database, typically schemas are used to cordon off sections of the database so that different groups of users have authority over the use of those sections.

*Tip!* In this document, we do not use the term “schema” to mean data model. We use “data model” instead.

### ***shared-nothing***

A distributed computing architecture in which nodes are independent and do not share disk or memory.

### ***SMC***

Outdated term for the [AMC](#).

### ***SQL-MapReduce***

Aster’s programming framework and API for writing data analysis and manipulation functions that you can run in a distributed manner.

### ***SQL-MapReduce function***

A function, usually invoked in a SELECT statement, that operates in Aster Database’s SQL-MapReduce framework. You can write SQL-MapReduce functions yourself, or use Teradata Aster’s functions.

### ***standby queen***

See [backup queen](#).

### ***star schema***

The database design schema that DBAs most commonly use in Aster Database is the *star schema*, consisting of (usually) one *fact table* surrounded by a set of *dimension tables*. Fact tables store the running log of events or transactions. Dimension tables describe items in detail. When you diagram the schema, it looks like a star, with the central fact table surrounded by dimension tables.

### ***stats db***

See [system tables](#).

### ***system tables***

Tables that hold Aster Database system information. These tables' names start with “nc\_”. These tables are often referred to as the “stats db” or as the “\_bee\_stats db”.

### ***tuple***

A “tuple” is an *ordered* set of values that we think of as a single record. Rows are the elements that comprise a database table. At any given time, each row will be represented by a specific tuple of values. A row can be updated over time to contain a different tuple. In the Aster Database documentation, we use “row,” except on those rare occasions when we’re trying to show the distinction between a tuple and a row.

### ***UDF***

user-defined function

### ***vworker***

A virtual worker responsible for storing and operating on data in Aster Database. Conceptually, a vworker is roughly equivalent to a physical data partition in Aster Database, and as a result you will often hear people refer to a vworker as a “partition” or “physical partition.”

The queen delegates work to vworkers, and query results are aggregated and returned via the queen. In a typical installation, you'll have as many active vworkers per worker node as you have CPU cores per worker node. See also [partition count](#).

### ***view***

A stored query accessible as a virtual table. A view is composed of the result set of a query. A view is not part of the physical schema, but is instead a dynamic, virtual table computed or collated from data in the database.

### ***virtual worker***

See [vworker](#).

### ***WAL file***

A Postgres write-ahead log file.

***worker***

In this document, we avoid this term. Instead, we say **vworker** to mean the basic Aster Database unit that does work, or we say **worker node** to mean the physical or virtual server that acts a worker machine in the cluster.

***worker node***

An Aster Database node (machine) that contains **vworkers**.



# Error Codes

All messages emitted by the Aster Database are assigned five-character error codes that follow the SQL standard's conventions for “SQLSTATE” codes. Applications that need to know which error condition has occurred should usually test the error code, rather than looking at the textual error message. Note that some, but not all, of the error codes produced by Aster Database are defined by the SQL standard; some additional error codes for conditions not defined by the standard have been invented or borrowed from other databases.

According to the standard, the first two characters of an error code denote a class of errors, while the last three characters indicate a specific condition within that class. Thus, an application that does not recognize the specific error code can still be able to infer what to do from the error class.

In the tables that follow, we list the Aster Database error codes and the meaning of each.

Table 11 - 1: Error Code Class 00 - Successful Completion Codes

| Error Code | Error Description     |
|------------|-----------------------|
| 0          | SUCCESSFUL COMPLETION |

Table 11 - 2: Error Code Class 0A - Feature Not Supported

| Error Code | Error Description     |
|------------|-----------------------|
| 0A000      | FEATURE NOT SUPPORTED |

Table 11 - 3: Error Code Class 21 - Cardinality Violation

| Error Code | Error Description     |
|------------|-----------------------|
| 21000      | CARDINALITY VIOLATION |

Table 11 - 4: Error Code Class 22 - Data Exception

| Error Code | Error Description           |
|------------|-----------------------------|
| 22000      | DATA EXCEPTION              |
| 22021      | CHARACTER NOT IN REPERTOIRE |

Table 11 - 4: Error Code Class 22 - Data Exception (continued)

| Error Code | Error Description                          |
|------------|--------------------------------------------|
| 22008      | DATETIME FIELD OVERFLOW                    |
| 22012      | DIVISION BY ZERO                           |
| 22005      | ERROR IN ASSIGNMENT                        |
| 2200B      | ESCAPE CHARACTER CONFLICT                  |
| 22022      | INDICATOR OVERFLOW                         |
| 22015      | INTERVAL FIELD OVERFLOW                    |
| 2201E      | INVALID ARGUMENT FOR LOGARITHM             |
| 2201F      | INVALID ARGUMENT FOR POWER FUNCTION        |
| 2201G      | INVALID ARGUMENT FOR WIDTH BUCKET FUNCTION |
| 22018      | INVALID CHARACTER VALUE FOR CAST           |
| 22007      | INVALID DATETIME FORMAT                    |
| 22019      | INVALID ESCAPE CHARACTER                   |
| 2200D      | INVALID ESCAPE OCTET                       |
| 22025      | INVALID ESCAPE SEQUENCE                    |
| 22P06      | NONSTANDARD USE OF ESCAPE CHARACTER        |
| 22010      | INVALID INDICATOR PARAMETER VALUE          |
| 22020      | INVALID LIMIT VALUE                        |
| 22023      | INVALID PARAMETER VALUE                    |
| 2201B      | INVALID REGULAR EXPRESSION                 |
| 22009      | INVALID TIME ZONE DISPLACEMENT VALUE       |
| 2200C      | INVALID USE OF ESCAPE CHARACTER            |
| 2200G      | MOST SPECIFIC TYPE MISMATCH                |
| 22004      | NULL VALUE NOT ALLOWED                     |
| 22002      | NULL VALUE NO INDICATOR PARAMETER          |
| 22003      | NUMERIC VALUE OUT OF RANGE                 |
| 22026      | STRING DATA LENGTH MISMATCH                |
| 22001      | STRING DATA RIGHT TRUNCATION               |
| 22011      | SUBSTRING ERROR                            |
| 22027      | TRIM ERROR                                 |
| 22024      | UNTERMINATED C STRING                      |

Table 11 - 4: Error Code Class 22 - Data Exception (continued)

| Error Code | Error Description             |
|------------|-------------------------------|
| 2200F      | ZERO LENGTH CHARACTER STRING  |
| 22P01      | FLOATING POINT EXCEPTION      |
| 22P02      | INVALID TEXT REPRESENTATION   |
| 22P03      | INVALID BINARY REPRESENTATION |
| 22P04      | BAD COPY FILE FORMAT          |
| 22P05      | UNTRANSLATABLE CHARACTER      |

Table 11 - 5: Error Code Class 23 - Integrity Constraint Violation

| Error Code | Error Description              |
|------------|--------------------------------|
| 23000      | INTEGRITY CONSTRAINT VIOLATION |
| 23502      | NOT NULL VIOLATION             |
| 23514      | CHECK VIOLATION                |
| 23518      | PARTITION KEY ERROR            |

Table 11 - 6: Error Code Class 25 - Invalid Transaction State

| Error Code | Error Description         |
|------------|---------------------------|
| 25000      | INVALID TRANSACTION STATE |
| 25P02      | IN FAILED SQL TRANSACTION |

Table 11 - 7: Error Code Class 26 - Invalid SQL Statement Name

| Error Code | Error Description          |
|------------|----------------------------|
| 26000      | INVALID SQL STATEMENT NAME |

Table 11 - 8: Error Code Class 28 - Invalid Authorization Specification

| Error Code | Error Description                   |
|------------|-------------------------------------|
| 28000      | INVALID AUTHORIZATION SPECIFICATION |

Table 11 - 9: Error Code Class 2B - Dependent Privilege Descriptors Still Exist

| Error Code | Error Description             |
|------------|-------------------------------|
| 2BP01      | DEPENDENT OBJECTS STILL EXIST |

Table 11 - 10: Error Code Class 2D - Invalid Transaction Termination

| Error Code | Error Description               |
|------------|---------------------------------|
| 2D000      | INVALID TRANSACTION TERMINATION |

Table 11 - 11: Error Code Class 34 - Invalid Cursor Name

| Error Code | Error Description   |
|------------|---------------------|
| 34000      | INVALID CURSOR NAME |

Table 11 - 12: Error Code Class 40 - Transaction Rollback

| Error Code | Error Description                          |
|------------|--------------------------------------------|
| 40000      | TRANSACTION ROLLBACK                       |
| 40002      | TRANSACTION INTEGRITY CONSTRAINT VIOLATION |
| 40001      | SERIALIZATION FAILURE                      |
| 40003      | STATEMENT COMPLETION UNKNOWN               |
| 40P01      | DEADLOCK DETECTED                          |

Table 11 - 13: Error Code Class 42 - Syntax Error or Access Rule Violation

| Error Code | Error Description                     |
|------------|---------------------------------------|
| 42000      | SYNTAX ERROR OR ACCESS RULE VIOLATION |
| 42601      | SYNTAX ERROR                          |
| 42501      | INSUFFICIENT PRIVILEGE                |
| 42846      | CANNOT COERCE                         |
| 42803      | GROUPING ERROR                        |
| 42602      | INVALID NAME                          |
| 42622      | NAME TOO LONG                         |
| 42939      | RESERVED NAME                         |
| 42804      | DATATYPE MISMATCH                     |
| 42P18      | INDETERMINATE DATATYPE                |
| 42809      | WRONG OBJECT TYPE                     |
| 42703      | UNDEFINED COLUMN                      |
| 42883      | UNDEFINED FUNCTION                    |
| 42P01      | UNDEFINED TABLE                       |

Table 11 - 13: Error Code Class 42 - Syntax Error or Access Rule Violation (continued)

| Error Code | Error Description           |
|------------|-----------------------------|
| 42P02      | UNDEFINED PARAMETER         |
| 42704      | UNDEFINED OBJECT            |
| 42701      | DUPLICATE COLUMN            |
| 42P03      | DUPLICATE CURSOR            |
| 42P04      | DUPLICATE DATABASE          |
| 42723      | DUPLICATE FUNCTION          |
| 42P07      | DUPLICATE TABLE             |
| 42712      | DUPLICATE ALIAS             |
| 42710      | DUPLICATE OBJECT            |
| 42702      | AMBIGUOUS COLUMN            |
| 42725      | AMBIGUOUS FUNCTION          |
| 42P08      | AMBIGUOUS PARAMETER         |
| 42P09      | AMBIGUOUS ALIAS             |
| 42P10      | INVALID COLUMN REFERENCE    |
| 42611      | INVALID COLUMN DEFINITION   |
| 42P11      | INVALID CURSOR DEFINITION   |
| 42P12      | INVALID DATABASE DEFINITION |
| 42P13      | INVALID FUNCTION DEFINITION |
| 42P16      | INVALID TABLE DEFINITION    |
| 42P17      | INVALID OBJECT DEFINITION   |

Table 11 - 14: Error Code Class 53 - Insufficient Resources

| Error Code | Error Description      |
|------------|------------------------|
| 53000      | INSUFFICIENT RESOURCES |
| 53100      | DISK FULL              |
| 53200      | OUT OF MEMORY          |
| 53300      | TOO MANY CONNECTIONS   |

Table 11 - 15: Error Code Class 54 - Program Limit Exceeded

| Error Code | Error Description      |
|------------|------------------------|
| 54000      | PROGRAM LIMIT EXCEEDED |
| 54001      | STATEMENT TOO COMPLEX  |
| 54011      | TOO MANY COLUMNS       |
| 54023      | TOO MANY ARGUMENTS     |

Table 11 - 16: Error Code Class 55 - Object Not In Prerequisite State

| Error Code | Error Description                |
|------------|----------------------------------|
| 55000      | OBJECT NOT IN PREREQUISITE STATE |
| 55006      | OBJECT IN USE                    |
| 55P03      | LOCK NOT AVAILABLE               |

Table 11 - 17: Error Code Class XX - Internal Error

| Error Code | Error Description |
|------------|-------------------|
| XX000      | INTERNAL ERROR    |

# Appendix 1 Parent/Child Tables

Beginning in Aster Database 4.6, parent/child tables created through inheritance was replaced by [Automatic Logical Partitioning](#), which is now the preferred method of logical partitioning. Logical partitioning through inheritance (parent/child tables) has been retained for backwards compatibility. This appendix describes how parent/child tables differ from automatic logical partitioning, and how they work. They have their own syntax and rules, distinct from automatic logical partitioning. This appendix has two sections:

- [Parent/Child Tables and Automatic Logical Partitioning Compared \(page 386\)](#) compares the benefits, rules and syntax for these two methods of logical partitioning.
- [Logical Partitioning Through Inheritance \(Parent/Child\) \(page 389\)](#) presents the syntax, rules, and some examples for logical partitioning through inheritance.

## Parent/Child Tables and Automatic Logical Partitioning Compared

The following sections are included to help you decide whether to keep the parent/child tables structure or to move to the newer method of logical partitioning. The benefits of the newer method are discussed, as well as some of the structural differences.

### Benefits of Automatic Logical Partitioning over Parent/Child Tables

Some benefits of the new automatic logical partitioning mechanism over logically partitioning tables through inheritance are:

- 1 The syntax for creating a child partition is more compact than writing a CREATE TABLE statement with an INHERITS clause. In particular, the partitioning column and parent table name do not need to be repeated for each child.
- 2 When using automatic logical partitioning, Aster Database will ensure that there are no overlaps in the values matched by different child partitions. In other words, a given record can match at most one partition.
- 3 The automatic logical partitioning format helps avoid mistakes with NULL values. With parent/child tables through inheritance, a common mistake would be to create child tables

with constraints like `CHECK(a >= 0 AND a < 10)`, `CHECK(a >= 10 AND a < 20)` where column "a" is sometimes NULL.

In the example, when column "a" is NULL, it can match both constraints, which means that it's difficult to predict which child table it belongs to. This is because in the SQL standard an expression such as `NULL < 20` will evaluate to NULL, and the SQL standard says that a CHECK constraint is not violated when the predicate evaluates to NULL.

The correct way to do this with parent/child tables would be something like:

```
CREATE TABLE t_parent (x int, a int, b text) DISTRIBUTUE BY HASH(x);
CREATE TABLE t_child1 (CHECK(a >= 0 AND a < 10 AND a IS NOT NULL))
INHERITS (t_parent);
CREATE TABLE t_child2 (CHECK(a >= 10 AND a < 20 and a IS NOT NULL))
INHERITS (t_parent);
```

With the new format (automatic logical partitioning), NULL values will not match a partition unless explicitly stated when defining it.

- 4 Many properties are automatically kept consistent across all children. Such properties include columns, constraints, ownership/privileges, indexes, etc. The index management is perhaps the most useful: CREATE INDEX on the parent table will automatically index all the child tables.
- 5 SQL commands operate on the hierarchy as on one table. With the older method of partitioning tables through inheritance, SELECT, UPDATE and DELETE worked in this way. Now you can also use INSERT, CREATE INDEX, MERGE, COPY, TRUNCATE, VACUUM, ANALYZE, CLUSTER, REINDEX, ALTER, GRANT, and REVOKE on the parent table to operate on the whole hierarchy.
- 6 Issuing \d in ACT now shows the parent table only, uncluttered by the listing of individual child partitions. This makes it easier to see the table names in the database at a glance. Note that the child tables may be viewed by accessing the system table `nc_user_child_partitions` (for example, issue `SELECT * from nc_user_child_partitions`).
- 7 An INSERT statement will always partition the data correctly, routing of rows to the correct child partition. This is not the case with inheritance partitioning, in which INSERT statements would target the parent table by default. (However, note that the AUTOPARTITION keyword is supported in the INSERT statement beginning in Aster Database 4.6, which allows INSERT data to flow to child tables created using inheritance).
- 8 The old style of logical partitioning through inheritance required a lot of manual work in order to do a logical backup of the entire hierarchy. First, each table would need to be backed up individually. Then when restoring, the parent table would need to be restored before the child tables, or else restoring the child tables would fail. Automatic logical partitioning makes backup/restore of a hierarchy the same as it is for a single table.

In the following example we use a PARTITION BY RANGE clause to create a fact table `trans` with four daily child partitions using automatic logical partitioning (strictly speaking, the first partition is a catch-all for older records):

```
CREATE FACT TABLE trans( id int, country varchar, ts timestamp )
DISTRIBUTE BY HASH(id)
PARTITION BY RANGE(ts) (
```

```

PARTITION oldrecords( END '2011-01-01' ), -- everything pre-2011
PARTITION jan01_2011( END '2011-01-02' ),
PARTITION jan02_2011( END '2011-01-03' ),
PARTITION jan03_2011( END '2010-01-04' )
) ;

```

The hierarchy may contain multiple generations, and operations on the parent table cascade automatically to all child partitions. In most cases, you manage only the top-level parent table. This includes SQL commands, ACT commands, backup/restore and loading.

## Automatic Logical Partitioning vs. parent-child tables with inheritance

If you are moving to automatic logical partitioning from using parent/child tables with inheritance, you should read this section. It explains how the newer, preferred method, automatic logical partitioning, is different from parent/child tables created through inheritance:

- 1 There can be no PRIMARY KEY or UNIQUE constraints on logically partitioned tables.
- 2 You must include a `DISTRIBUTE BY` clause when creating a logically partitioned table. To distribute data using a column name, choose `DISTRIBUTE BY HASH (column_name)`. If you want the child partitions to be replicated on each node, specify `DISTRIBUTE BY REPLICATION`. If you do not specify a `DISTRIBUTE BY` clause, the `CREATE TABLE` will fail with the error message:

```
ERROR: use of PARTITION BY clause requires the use of the DISTRIBUTE BY clause at or near ";"
```

Using inheritance partitioning, it is possible to define arbitrarily complex CHECK constraints (but complex constraints are less likely to help query optimization).

- 3 If used, the `ONLY` clause is ignored, since we are viewing the hierarchy as a single table.
- 4 In most cases, you manage only the top-level parent table. For SQL commands that need to act on a child partition in isolation, you must first detach the partition from the parent, making it a standalone table. The table may be reattached as a partition if desired when the operation has completed. For example, if you want to only ANALYZE a child partition, you need to DETACH it to create a standalone table, ANALYZE the table, and reATTACH it as a partition of the parent table. The DETACH and ATTACH clauses are part of the `ALTER TABLE` command.
- 5 INDEX cascades to child partitions. That is, `CREATE INDEX` on the parent table will automatically index all the child partitions. This simplifies index creation and makes queries faster. However, it can slow down data load times for the partitions. To load data into an indexed partition more quickly, you can detach the partition from the partitioned table, drop the index, load the data, recreate the index, and then attach the partition back to its parent.

If you know that data will land in a specific partition, you can load the data into a separate table and then use the `ALTER TABLE...ATTACH PARTITION` command to add the table as a partition after loading the data. You can create the table without any indexes or constraints; when you attach it as a partition, the indexes and constraints will be inherited automatically.

# Logical Partitioning Through Inheritance (Parent/Child)

In releases of Aster Database prior to version 4.6, logical partitioning was supported using inheritance. Aster Database 5.10.00.01 continues to support logical partitioning with inheritance for backwards compatibility with tables created in this way. Using logical partitioning with inheritance, the single, logical table is divided into child tables using parent-child table inheritance declared with the `INHERITS` keyword. Logical partitioning with inheritance is also known as “parent-child table inheritance” or “child table partitioning”.



**Tip!** Don't confuse logical partitioning with distribution! See [Distribution and Logical Partitioning Compared \(page 29\)](#).

## Table Inheritance Overview

Table inheritance differs from automatic logical partitioning in that the parent and child entities are still distinct tables, linked through inheritance. In automatic logical partitioning, there is only one table, with its data divided into partitions, such that for loading data, backup, and SQL statements, only the parent table is referenced and operations automatically act on child partitions as necessary. Because managing data is simplified with automatic logical partitioning, it is recommended over using logical partitioning through inheritance. For more information, see [“Automatic Logical Partitioning” on page 35](#).

If you want to maintain backwards compatibility, for example with existing scripts, it is still possible create parent/child tables through inheritance (the old way). However, there is no way to migrate an existing hierarchy using inheritance to the new automatic partitioning format. If you use the old method, you don't get the benefit of automatic partitioning on `INSERT`. If you have an old style hierarchy, you can still perform an `INSERT`, but you must specify the `AUTOPARTITION` keyword in the `INSERT` clause. For the new logically partitioned tables, you do not need to specify the `AUTOPARTITION` keyword. Data is automatically inserted into the partition where it belongs.

## Benefits of Logical Partitioning through Inheritance

Logical partitioning through inheritance provides some of the same benefits as automatic logical partitioning:

- 1 Query performance is improved dramatically for certain kinds of queries.
- 2 Update performance is improved too, since each piece of the table has indexes smaller than an index on the entire data set would be.
- 3 Bulk deletes may be accomplished by simply removing a child table, if that requirement is planned into the partitioning design. `DROP TABLE` is far faster than a bulk `DELETE`.

## How to Logically Partition a Table with Inheritance

To logically partition a table, you use inheritance (`INHERITS`). Each logical partition is created as a child table of a single parent table. The parent table itself is normally empty; it exists just to represent the entire data set. You may create a multi-generational tree, for example with a parent table that is the parent of child tables for each year, which in turn are the parents of child tables for each month, which in turn are the parents of child tables for each day. Data is usually stored only in the ultimate child generation.

To determine which records belong to which partition, you define list partitioning check constraints or range partitioning check constraints (`CHECK`). A *list partitioning* constraint provides a list of key values that belong to a partition. A *range partitioning* constraint specifies a numeric, alphabetical, or date range of key values that belong to a partition.

A simple, logically partitioned table with inheritance can be set up using the following steps.

- 1 Create the base table (the ultimate parent), from which all of the logical partitions will inherit. This table will contain no data. There is no special requirement for indexes or constraints on this table.
- 2 Create the child tables, each of which `INHERITS` its form from its parent table. A child table may or may not add columns to the set of columns inherited from the parent. Each child table acts as a logical partition.
- 3 Add a `CHECK` constraints to each child table to define the allowed key values for that logical partition. See “[Check Constraints](#)” on page 161. Typical examples would be:

```
CHECK ( order_timestamp::date > '2008-09-30' and order_timestamp::date < '2008-11-01' )
CHECK ( ymdh between '2005-07-01' and '2005-07-31' );
CHECK ( city IN ( 'New York', 'Palo Alto', 'San Francisco' ) );
CHECK ( entity_id = 2 );
```



**Tip!** Take care that the constraints you define *do not create overlapping logical partitions*. A simple mistake would be to set up range constraints like this:

```
CHECK ( ymdh BETWEEN '2005-07-01' AND '2005-08-01' );
CHECK ( ymdh BETWEEN '2005-08-01' AND '2005-09-01' );
In this example, it is not clear in which partition the ymdh value '2005-08-01' resides.
```

---

Later, you can modify inheritance and constraints using the `ALTER TABLE` command.

Below is an example of a child table that inherits from its parent, the `orders` table:

```
CREATE FACT TABLE orders_2008_10 (
    order_id      int ,
    customer_id   int NOT NULL,
    amount        int ,
    order_timestamp timestamp,
    CHECK ( order_timestamp::date > '2008-09-30' and order_timestamp::date
< '2008-11-01' )
) INHERITS (orders);
```

The result of the following query will include a count of rows in all children of the `orders` table (the `orders_2008_10` table as well as any others you might have defined):

```
SELECT COUNT(*) FROM orders;
```

Here's an example that might benefit from logical partitioning. In the query below, Aster Database includes the `orders_2008_10` child table (and any earlier-dated child tables it finds) because its CHECK constraint matches the filter condition in the WHERE clause.

```
SELECT COUNT(*) FROM ORDERS WHERE order_timestamp::date < '2008-11-01';
```

## Inheritance: What do the child tables inherit?

A child table inherits its parent's

- columns and their datatypes
- `DISTRIBUTE BY` constraints
- check constraints
- table type: the child of a fact table is a fact table; the child of a dimension table is a dimension table.

A child table *does not inherit* its parent's

- `PRIMARY KEY` constraints
- index definitions

## Cascading: What commands cascade to child tables?

A SELECT, UPDATE, or DELETE action on the parent will cascade (apply to) its children, subject to the check constraints of each child. However, you may not use UPDATE to move data from one child table to another. Attempting this will return an error, and the UPDATE will not be made.

Many SQL commands support the `ONLY` keyword, which specifies that the command will apply only to the table that is named in the statement, and not to any of its children.

`INSERT` and `COPY` do not cascade. You must `INSERT` or `COPY` directly to the correct destination table (which is usually a child table), or you can use the autopartitioning feature (see the *Aster Client Guide*) of Aster Database Loader to have Aster Database automatically distribute each row to the right child table. Note that when using the newer logical partitioning syntax [“Automatic Logical Partitioning” on page 35](#), `INSERT` and `COPY` *do* cascade to child partitions. In fact, managing logically partitioned tables is simpler using automatic logical partitioning, and for that reason the newer method is recommended.

## A Simple Child Table Partitioning Example

As a simple example of logical partitioning through inheritance, let's consider two tables: `cities` and `capitals`. Now, because capitals are also cities, you want some way to show the `capitals` implicitly when you list all `cities`. You might invent a schema like this:

```
CREATE TABLE capitals (
    name      text,
    population real,
    altitude  int,    -- (in ft)
    state     char(2)
);
```

```
CREATE TABLE non_capitals (
    name      text,
    population real,
    altitude   int      -- (in ft)
) ;
```

This method works fine as far as querying goes, but can get difficult when you need to update several rows, for one thing.

A better solution is:

```
CREATE TABLE cities (
    name      text,
    population real,
    altitude   int      -- (in ft)
) ;

CREATE TABLE capitals (
    state      char(2)
) INHERITS (cities);
```

In this case, a row of capitals inherits all columns (name, population, and altitude) from its parent, cities. The type of the column name is text, a native Aster Database type for variable length character strings. State capitals have an extra column, `state`, that shows their state. In Aster Database, a table can inherit from zero or more other tables.

For example, the following query finds the names of all cities, including state capitals that are located at an altitude over 500 ft.:

```
SELECT name, altitude
  FROM cities
 WHERE altitude > 500;
which returns:
```

| name      | altitude |
|-----------|----------|
| Las Vegas | 2174     |
| Mariposa  | 1953     |
| Madison   | 845      |
| (3 rows)  |          |

On the other hand, the following query finds all the cities that are not state capitals and are situated at an altitude of 500 ft. or higher:

```
SELECT name, altitude
  FROM ONLY cities
 WHERE altitude > 500;



name	altitude
Las Vegas	2174
Mariposa	1953
(2 rows)	


```

Here the `ONLY` before `cities` indicates that the query should be run only on the `cities` table, and not on tables below `cities` in the inheritance hierarchy.

As we noted in an earlier section, an `INSERT` or `COPY` on the parent does not cascade or send its data to the child tables. This means that, in the following example, the following `INSERT` statement will fail:

```
INSERT INTO cities (name, population, altitude, state)
    VALUES ('New York', NULL, NULL, 'NY');
```

We might hope that the data would somehow be routed to the capitals table, but this does not happen: `INSERT` always inserts into exactly the table specified.

## A Detailed Child Table Partitioning Example

Suppose we are constructing a database for a large website. The company measures traffic volumes every day as well as impression volumes per page in each topic. Conceptually, we want a fact table defined as shown below. Recall that Aster Database's `CREATE TABLE` creates a *fact* table by default, so we can type `CREATE FACT TABLE` or we can type it as shown here:

```
CREATE TABLE impressions (
    page_id      int not null,
    logdate      date not null,
    topic_id     int
);
```

We know that most queries will access just the last week's, month's or quarter's data, since the main use of this table will be to prepare online reports for clients.

In this situation we can use partitioning to help us meet all of our different requirements for the impressions table. Following the steps outlined above, partitioning can be set up like this:

1 The base table is the impressions table, declared exactly as above.

2 Next, we create one partition for each active month:

```
CREATE TABLE impressions_yy04mm02 ( ) INHERITS (impressions);
CREATE TABLE impressions_yy04mm03 ( ) INHERITS (impressions);
...
CREATE TABLE impressions_yy05mm11 ( ) INHERITS (impressions);
CREATE TABLE impressions_yy05mm12 ( ) INHERITS (impressions);
CREATE TABLE impressions_yy06mm01 ( ) INHERITS (impressions);
```

3 We add non-overlapping table constraints, so that our table creation script becomes:

```
CREATE TABLE impressions_yy05mm01 (
    CHECK ( logdate >= DATE '2005-01-01' AND logdate < DATE '2005-02-01' )
) INHERITS (impressions);

CREATE TABLE impressions_yy05mm02 (
    CHECK ( logdate >= DATE '2005-02-01' AND logdate < DATE '2005-03-01' )
) INHERITS (impressions);

...
CREATE TABLE impressions_yy05mm12 (
    CHECK ( logdate >= DATE '2005-12-01' AND logdate < DATE '2006-01-01' )
) INHERITS (impressions);

CREATE TABLE impressions_yy06mm01 (
    CHECK ( logdate >= DATE '2006-01-01' AND logdate < DATE '2006-02-01' )
) INHERITS (impressions);
```

4 Finally, we add indexes on the `topic_id` column:

```
CREATE INDEX idx_topic_id_impressions_yy05mm01 ON impressions_yy05mm01
(topic_id);
...
CREATE INDEX idx_topic_id_impressions_yy05mm12 ON impressions_yy05mm12
(topic_id);
CREATE INDEX idx_topic_id_impressions_yy06mm01 ON impressions_yy06mm01
(topic_id);
```

We choose not to add further indexes at this time.

### How does logical partitioning improve performance?

How did we gain in query performance by implementing the partitions shown above? Queries benefit from constraints defined on the partition. Consider the following query:

```
SELECT count(*) FROM impressions WHERE logdate >= DATE '2005-12-01' and
logdate < DATE '2006-01-01';
```

The optimizer now knows that this query will find qualifying rows only in the `impressions_yy05_mm12` table, and hence completely skip scanning of the other tables derived from `impressions`. If the `impressions` table is huge, we just finished the query by looking at only a twelfth of the data!

Such optimization only works when the query's WHERE clause contains constants. A parameterized query will not be optimized, since the planner cannot know what partitions the parameter value might select at runtime. For the same reason, functions such as `CURRENT_DATE` must be avoided.



## Appendix 2 Licenses for Tools Used in Aster Database

This appendix contains the license text for open source products that may be included in Aster Database version 5.10.00.01. The license text shown here is also available on your Aster Database queen in the /home/beehive/licenses directory.

### List of Third-Party Tools in Aster Database

The following software may be included in this product:

Acegi Security  
Amazon EC2  
Antlr  
Apache Commons  
Apache Geronimo  
Apache Jakarta ORO  
Apache JMeter  
Apache HTTP Server  
Apache Maven  
Apache Maven Doxia  
Apache Portability Runtime  
Apache Struts  
Apache Tiles  
Apache Tomcat Server  
Apache Velocity  
ASM  
Base64.c  
Base64.java  
Bean Shell  
Blosc Shuffling  
Boost  
BZip2  
CG Lib  
Codehaus Plexus  
Colt  
ConfigIni.pm  
Confuse  
Cpp Unit  
Curl  
DateFormatter.js

## List of Third-Party Tools in Aster Database

Dialog.py  
Direct Web Remoting  
DOM4J  
Eclipse  
EditLine  
Ehcache  
Enterprise JavaBeans  
Excanvas.js  
Expat  
EZMorph  
Free Marker  
FreeType  
Fuse  
Ganglia  
GNU GCC  
GNU GDB  
GNU C Library  
GNU Scientific Library  
Google Gson  
Google Perf Tools  
Gson Pretty Printer  
GTK/GNOME GLib  
Hadoop  
Hibernate  
Hibernate Middle Gen  
HTML Parser  
iODBC  
IOzone  
IPerf  
JAMA - Java Matrix Package  
JavaBeans Activation Framework  
Java Arch for XML Binding  
Java Mail  
Java Persistence API  
Java Secure Channel (JSCH)  
Java Servlet API  
Java Transaction API  
Java XML Web Services  
JavaScript Templating  
JBoss Java Assist  
Jetty  
JQuery  
JQuery BGI Frame  
JQuery Beauty Tips  
JQuery Dimensions  
JQuery Flot  
JQuery Highlight 3  
JQuery Hover Intent  
JQuery Multi-Select  
JQuery QTip  
JQuery Scroll To  
JQuery Sparkline  
JQuery Tool Tip  
JQuery Tree Table  
JQuery UI  
JQuery Validate  
Json C  
Json Cpp  
Json Lib

JSP Standard Tag Library  
JSP Unified Expression Lang  
JTidy  
JUnit  
Lib Art  
Lib Edit  
Lib PNG  
Lib uuid  
Lib XML2  
LM Bench  
Log4J  
Merriampark Levenshtein Dist  
Microsoft Reporting Services  
MinGW  
MochiKit  
Mod Auth PQSQL  
Mono  
Mono Security Protocol  
MurmurHash2  
NCurses  
NetCat  
Netlogd  
Net SNMP  
Netty  
NPG SQL  
NPG SQL TestSuite  
OCaml  
OleDB  
OpenCSV  
Open LDAP  
Open SSL  
Open SSH  
OpenSymphony OGNL  
OpenSymphony Quartz  
OpenSymphony XWork  
OProfile  
Oracle JDBC  
Perl  
PG Foundry IP4R  
php  
PlotKit  
Popup.js  
Popt  
Porter Stemmer  
PostgreSQL  
PostgreSQL datetime.h  
PostgreSQL JDBC  
PSQL ODBC  
Ptr.h  
Public Domain Curses  
Python  
Python Nose  
Python Logilab  
Readline  
RRDTool  
Ruby  
Silk Icons  
Simple XML  
slf4j

The Source for these packages are available upon request:

```
Spring Framework
String Template
Sun JDK
Swig
SysStat
Trove
TrueZip
Uncomplicated Firewall
Unix ODBC
UTF8 CPP
Web Services Desc Lang for Java
Webworks/Quadratay
XPP3
XStream
Zlib
```

## **The Source for these packages are available upon request:**

```
Email support@asterdata.com
Subject "opensource"
Include the package name(s) you are requesting.
```

```
Apache HTTP Server
Bean Shell
ConfigIni.pm
Cpp Unit
Dialog.py
Eclipse
Enterprise JavaBeans
Fuse
Ganglia
GNU GCC
GNU C Library
Gson Pretty Printer
Java Persistence API
JBoss Java Assist
Jetty
Json Cpp
JSP Standard Tag Library
JSP Unified Expression Lang
Lib Art
LM Bench
OleDB
Perl
PSQL ODBC
Python
Python Nose
Python Logilab
Readline
RRDTool
Ruby
SysStat
Trove
```

Uncomplicated Firewall  
Unix ODBC

## Text of Third-Party Software Licenses

Below, we include the text of the licenses for third-party and open-source software that might be included in Aster Database.

Acegi Security  
Apache License v2

```
=====
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Acegi Security System for      ==
== Spring distribution.  ==
=====
```

The end-user documentation included with a redistribution, if any, must include the following acknowledgement:

"This product includes software developed by the Acegi Security System for Spring Project (<http://acegisecurity.org>)."

Alternately, this acknowledgement may appear in the software itself, if and wherever such third-party acknowledgements normally appear.

The names "Acegi", "Acegi Security System" and "Acegi Security System for Spring" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [ben.alex@acegi.com.au](mailto:ben.alex@acegi.com.au).

Amazon EC2

Amazon Software License

This Amazon Software License ("License") governs your use, reproduction, and distribution of the accompanying software as specified below.

### 1. Definitions

"Licensor" means any person or entity that distributes its Work.

"Software" means the original work of authorship made available under this License.

"Work" means the Software and any additions to or derivative works of the Software that are made available under this License.

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the meaning as provided under U.S. copyright law; provided, however, that for the purposes of this License, derivative works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work.

## Text of Third-Party Software Licenses

Works, including the Software, are "made available" under this License by including in or with the Work either (a) a copyright notice referencing the applicability of this License to the Work, or (b) a copy of this License.

### 2. License Grants

2.1 Copyright Grant. Subject to the terms and conditions of this License, each Licenser grants to you a perpetual, worldwide, non-exclusive, royalty-free, copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense and distribute its Work and any resulting derivative works in any form.

2.2 Patent Grant. Subject to the terms and conditions of this License, each Licensor grants to you a perpetual, worldwide, non-exclusive, royalty-free patent license to make, have made, use, sell, offer for sale, import, and otherwise transfer its Work, in whole or in part. The foregoing license applies only to the patent claims licensable by Licensor that would be infringed by Licensor's Work (or portion thereof) individually and excluding any combinations with any other materials or technology.

### 3. Limitations

3.1 Redistribution. You may reproduce or distribute the Work only if (a) you do so under this License, (b) you include a complete copy of this License with your distribution, and (c) you retain without modification any copyright, patent, trademark, or attribution notices that are present in the Work.

3.2 Derivative Works. You may specify that additional or different terms apply to the use, reproduction, and distribution of your derivative works of the Work ("Your Terms") only if (a) Your Terms provide that the use limitation in Section 3.3 applies to your derivative works, and (b) you identify the specific derivative works that are subject to Your Terms. Notwithstanding Your Terms, this License (including the redistribution requirements in Section 3.1) will continue to apply to the Work itself.

3.3 Use Limitation. The Work and any derivative works thereof only may be used or intended for use with the web services, computing platforms or applications provided by Amazon.com, Inc. or its affiliates, including Amazon Web Services LLC.

3.4 Patent Claims. If you bring or threaten to bring a patent claim against any Licensor (including any claim, cross-claim or counterclaim in a lawsuit) to enforce any patents that you allege are infringed by any Work, then your rights under this License from such Licensor (including the grants in Sections 2.1 and 2.2) will terminate immediately.

3.5 Trademarks. This License does not grant any rights to use any Licensor's or its affiliates' names, logos, or trademarks, except as necessary to reproduce the notices described in this License.

3.6 Termination. If you violate any term of this License, then your rights under this License (including the grants in Sections 2.1 and 2.2) will terminate immediately.

4. Disclaimer of Warranty. THE WORK IS PROVIDED "AS IS" WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT. YOU BEAR THE RISK OF UNDERTAKING ANY ACTIVITIES UNDER THIS LICENSE. SOME STATES' CONSUMER LAWS DO NOT ALLOW EXCLUSION OF AN IMPLIED

WARRANTY, SO THIS DISCLAIMER MAY NOT APPLY TO YOU.

**5. Limitation of Liability.** EXCEPT AS PROHIBITED BY APPLICABLE LAW, IN NO EVENT AND UNDER NO LEGAL THEORY, WHETHER IN TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE SHALL ANY LICENSOR BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATED TO THIS LICENSE, THE USE OR INABILITY TO USE THE WORK (INCLUDING BUT NOT LIMITED TO LOSS OF GOODWILL, BUSINESS INTERRUPTION, LOST PROFITS OR DATA, COMPUTER FAILURE OR MALFUNCTION, OR ANY OTHER COMMERCIAL DAMAGES OR LOSSES), EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the notice folder accompanying the file. Such other license terms will then apply in lieu of the terms of the Amazon Software License above.

%% The following software may be included in this product:

gnu getopt jar.

Use of any of this software is governed by the terms of the license below:

GNU LIBRARY GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.  
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is  
numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis

## Text of Third-Party Software Licenses

or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a

"work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1

above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work

## Text of Third-Party Software Licenses

during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining

where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

## Text of Third-Party Software Licenses

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

%% The following software may be included in this product:

xalan-j2.jar

Use of any of this software is governed by the terms of the license below:

```
=====
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xalan serializer      ==
== distribution.   ==
=====
```

This product includes software developed by IBM Corporation (<http://www.ibm.com>) and The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software was originally based on the following:

- software copyright (c) 1999-2002, Lotus Development Corporation.,  
<http://www.lotus.com>.

## Text of Third-Party Software Licenses

- software copyright (c) 2001-2002, Sun Microsystems.,  
<http://www.sun.com>.
- software copyright (c) 2003, IBM Corporation., <http://www.ibm.com>.

%% The following software may be included in this product:

commons-discovery.jar  
commons-httpclient.jar  
commons-logging.jar  
xalan-j2-serializer.jar  
xml-security.jar  
log4j.jar  
wss4j.jar  
wstx-asl.jar

Use of any of this software is governed by the terms of the license below:

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

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

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

#### END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

%% The following software may be included in this product:  
xfire-all.jar  
xfire-jsr181-api.jar

Copyright (c) 2005 Envoi Solutions LLC

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

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

## Text of Third-Party Software Licenses

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

```
%% The following software may be included in this product:  
jdom.jar
```

```
/* ======  
 * The Apache Software License, Version 1.1  
 *  
 * Copyright (c) 2000 The Apache Software Foundation. All rights  
 * reserved.  
 *  
 * Redistribution and use in source and binary forms, with or without  
 * modification, are permitted provided that the following conditions  
 * are met:  
 *  
 * 1. Redistributions of source code must retain the above copyright  
 * notice, this list of conditions and the following disclaimer.  
 *  
 * 2. Redistributions in binary form must reproduce the above copyright  
 * notice, this list of conditions and the following disclaimer in  
 * the documentation and/or other materials provided with the  
 * distribution.  
 *  
 * 3. The end-user documentation included with the redistribution,  
 * if any, must include the following acknowledgment:  
 *     "This product includes software developed by the  
 *     Apache Software Foundation (http://www.apache.org/)."  
 *     Alternately, this acknowledgment may appear in the software itself,  
 *     if and wherever such third-party acknowledgments normally appear.  
 *  
 * 4. The names "Apache" and "Apache Software Foundation" must  
 * not be used to endorse or promote products derived from this  
 * software without prior written permission. For written  
 * permission, please contact apache@apache.org.  
 *  
 * 5. Products derived from this software may not be called "Apache",  
 * nor may "Apache" appear in their name, without prior written  
 * permission of the Apache Software Foundation.  
 *  
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED  
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR  
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT  
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF  
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND  
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT  
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
 * SUCH DAMAGE.  
 * ======
```

```

*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
* Portions of this software are based upon public domain software
* originally written at the National Center for Supercomputing Applications,
* University of Illinois, Urbana-Champaign.
*/

```

## Antlr

## A N T L R

## =====

## \*SOFTWARE RIGHTS\*

ANTLR 1989-2006 Developed by Terence Parr @ University of San Francisco

We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commercial software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

The primary ANTLR guy:

Terence Parr  
 parrt@cs.usfca.edu  
 parrt@antlr.org

## WELCOME TO ANTLR!

If you have problems or think you have found a bug in ANTLR, see the section BUGS in the ANTLR manual.

Please consult the INSTALL.txt file for information on tested configurations. If you have a comment about an already tested configuration, or have tried ANTKR on a new configuration, please let us know as described in INSTALL.txt. Free software only works if we all help out.

## Text of Third-Party Software Licenses

Finally, we cannot guarantee that this release will not completely wipe out all of your work from your system. We do some simple testing before each release, but you are completely on your own. We recommend testing this release on a source repository that is not critical to your work.

THIS SOFTWARE IS SUPPLIED COMPLETELY "AS IS".  
NO WARRANTY....

Thanks for your support!

-The ANTLR Team-

---

### WHAT IS ANTLR?

ANTLR, (AN)other (T)ool for (L)anguage (R)ecognition - formerly known as PCCTS - is a language tool that provides a framework for constructing recognizers, compilers, and translators from grammatical descriptions containing actions in the following languages:

Java,  
C++,  
C# or  
Python

(You can use PCCTS 1.xx to generate C-based parsers).

Computer language translation has become a common task. While compilers and tools for traditional computer languages (such as C or Java) are still being built, their number is dwarfed by the thousands of mini-languages for which recognizers and translators are being developed. Programmers construct translators for database formats, graphical data files (e.g., PostScript, AutoCAD), text processing files (e.g., HTML, SGML). ANTLR is designed to handle all of your translation tasks.

Prof. Terence Parr has been working on ANTLR since 1989 and, together with his colleagues, has made a number of fundamental contributions to parsing theory and language tool construction, leading to the resurgence of LL(k)-based recognition tools.

Have a look at the history section at the end of this document on how ANTLR has evolved over time. For most up-to-date information read <http://www.antlr.org/history.html>.

---

### UPGRADING?

See

<http://www.antlr.org/blog/CHANGES-2.7.7.txt>

for a description of features new in this version. There are no incompatibilities known to a previous 2.7.x installation. If you found a problem please let us know.

---

## INSTALLATION?

Please read the INSTALL.txt file for installation instructions. The brief summary is:

```
$ ./configure  
$ make  
$ make test      # optional  
$ su root        # optional  
$ make install
```

---

## ANTLR IS INSTALLED - WHAT'S NEXT?

Please read "doc/getting-started.html" on what you are supposed to do. Here's a very brief summary for the impatient:

ANTLR is a command line tool. To run ANTLR you need to have JAVA installed. The basic steps are:

- a. write a grammar file - mygrammar.g
- b. run ANTLR like

```
$ CLASSPATH=antlr.jar  
$ java antlr.Tool mygrammar.g
```
- c. write a driver program using source code generated by ANTLR, ie. Main.java, main.cpp, Main.cs or main.py
- d. link generated code, your driver code, ANTLR's core library and any additional library you are using together to get an executable
- f. run the executable on arbitrary input to be parsed

For a set of standard examples have a look into directory "examples" and appropriate subdirectories.

---

## WANT TO KNOW MORE?

The documentation is in the "doc" subdirectory and "index.html" is the main entry point.

Further information available at

<http://www.antlr.org>

---

## WHO CONTRIBUTED TO THIS MESS?

## Text of Third-Party Software Licenses

Project Lead and Supreme Dictator Terence Parr, University of San Francisco

Help with initial coding John Lilly, Empathy Software

C++ code generator by Peter Wells and Ric Klaren

C# code generation by Micheal Jordan, Kunle Odutola and Anthony Oguntimohin.

Python's universe has been extended by Wolfgang Heflinger and Marq Kole

Substantial intellectual effort donated by Loring Craymer, Monty Zukowski, Jim Coker, Scott Stanchfield, John Mitchell, Chapman Flack (UNICODE, streams)

Source changes for Eclipse and NetBeans by Marco van Meegen and Brian Smith

Infrastructure support from Perforce - The world's best source code control system

---

## WANNA KNOW ABOUT ANTLR's HISTORY?

The PCCTS project began as a parser-generator project for a graduate course at Purdue University in the Fall of 1988 taught by Hank Dietz "translator-writing systems". Under the guidance of Professor Dietz, the parser generator, ANTLR (originally called YUCC), continued after the termination of the course and eventually became the subject of Terence Parr's Master's thesis. Originally, lexical analysis was performed via a simple scanner generator which was soon replaced by Will Cohen's DLG in the Fall of 1989 (DFA-based lexical-analyzer generator, also an offshoot of the graduate translation course).

The alpha version of ANTLR was totally rewritten resulting in 1.00B. Version 1.00B was released via an internet newsgroup (comp.compilers) posting in February of 1990 and quickly gathered a large following. 1.00B generated only LL(1) parsers, but allowed the merged description of lexical and syntactic analysis. It had rudimentary attribute handling similar to that of YACC and did not incorporate rule parameters or return values; downward inheritance was very awkward. 1.00B-generated parsers terminated upon the first syntax error. Lexical classes (modes) were not allowed and DLG did not have an interactive mode.

Upon starting his Ph.D. at Purdue in the Fall of 1990, Terence Parr began the second total rewrite of ANTLR. The method by which grammars may be practically analyzed to generate LL(k) lookahead information was discovered in August of 1990 just before Terence's return to Purdue. Version 1.00 incorporated this algorithm and included the AST mechanism, lexical classes, error classes, and automatic error recovery; code quality and portability were higher. In February of 1992 1.00 was released via an article in SIGPLAN Notices. Peter Dahl, then Ph.D. candidate, and Professor Matt O'Keefe (both at the University of Minnesota) tested this version extensively. Dana

Hoggatt (Micro Data Base Systems, Inc.) tested 1.00 heavily.

Version 1.06 was released in December 1992 and represented a large feature enhancement over 1.00. For example, rudimentary semantic predicates were introduced, error messages were significantly improved for k>1 lookahead and ANTLR parsers could indicate that lookahead fetches were to occur only when necessary for the parse (normally, the lookahead "pipe" was constantly full). Russell Quong joined the project in the Spring of 1992 to aid in the semantic predicate design. Beginning and advanced tutorials were created and released as well. A makefile generator was included that sets up dependencies and such correctly for ANTLR and DLG. Very few 1.00 incompatibilities were introduced (1.00 was quite different from 1.00B in some areas).

Version 1.10 was released on August 31, 1993 after Terence's release from Purdue and incorporated bug fixes, a few feature enhancements and a major new capability -- an arbitrary lookahead operator (syntactic predicate), "(a)?b". This feature was codesigned with Professor Russell Quong also at Purdue. To support infinite lookahead, a preprocessor flag, ZZINF\_LOOK, was created that forced the ANTLR() macro to tokenize all input prior to parsing. Hence, at any moment, an action or predicate could see the entire input sentence. The predicate mechanism of 1.06 was extended to allow multiple predicates to be hoisted; the syntactic context of a predicate could also be moved along with the predicate.

In February of 1994, SORCERER was released. This tool allowed the user to parse child-sibling trees by specifying a grammar rather than building a recursive-descent tree walker by hand. Aaron Sawdey at The University of Minnesota became a second author of SORCERER after the initial release. On April 1, 1994, PCCTS 1.20 was released. This was the first version to actively support C++ output. It also included important fixes regarding semantic predicates and (...)+ subrules. This version also introduced token classes, the "not" operator, and token ranges.

On June 19, 1994, SORCERER 1.00B9 was released. Gary Funck of Intrepid Technology joined the SORCERER team and provided very valuable suggestions regarding the "transform" mode of SORCERER.

On August 8, 1994, PCCTS 1.21 was released. It mainly cleaned up the C++ output and included a number of bug fixes.

From the 1.21 release forward, the maintenance and support of all PCCTS tools was picked up by Parr Research Corporation.

A sophisticated error handling mechanism called "parser exception handling" was released for version 1.30. 1.31 fixed a few bugs.

Release 1.33 is the version corresponding to the initial book release.

ANTLR 2.0.0 came out around May 1997 and was partially funded so Terence hired John Lilley, a maniac coder and serious ANTLR hacker, to build much of the initial version. Terence did the grammar analyzer, naturally.

John Mitchell, Jim Coker, Scott Stanchfield, and Monty Zukowski donate lots of brain power to ANTLR 2.xx in general.

ANTLR 2.1.0, July 1997, mainly improved parsing performance, decreased parser memory requirements, and added a lot of cool lexer features including a case-insensitivity option.

ANTLR 2.2.0, December 1997, saw the introduction of the new <http://www.antlr.org> website. This release also added grammar inheritance, enhanced AST support, and enhanced lexical translation support (each lexical rule now was considered to return a Token object even when referenced by another lexical rule).

ANTLR 2.3.0, June 1998, was the first version to have Peter Wells C++ code generator.

ANTLR 2.4.0, September 1998, introduced the ParseView parser debugger by Scott Stanchfield. This version also had a semi-functional -html option to generate HTML from your grammar for reading purposes. Scott and Terence updated the file I/O to be JDK 1.1.

ANTLR 2.5.0, November 1998, introduced the filter option for the lexer that lets ANTLR behave like SED or AWK.

ANTLR 2.6.0, March 1999, introduced token streams. Chapman Flack, Purdue Graduate student, pounded me at the right moment about streams, nudging me in the right direction.

MageLang Institute currently provides support and continues development of ANTLR.

MageLang becomes jGuru.com as we quit doing Java training and start building the jGuru Java developer's website.

2.7.0 released January 19, 2000 had the following enhancements:

- \* Nongreedy subrules
- \* Heterogeneous trees
- \* Element options. To support heterogeneous trees, elements such as token references may now include options.
- \* Exception hierarchy redesign
- \* XML serialization
- \* Improved C++ code generator
- \* New Sather code generator

2.7.1 released October 1, 2000 had the following enhancements

- \* ANTLR now allows UNICODE characters because Terence made case-statement expressions more efficient ;) See the unicode example in the distribution and the brief blurb in the documentation.
- \* Massively improved C++ code generator (Thanks to Ric Klaren).
- \* Added automatic column setting support.
- \* Ter added throws to tree and regular parsers .

2.7.2 release January 19, 2003 was mainly a bug fix release,

- \* but also included a C# code generator by Micheal Jordan, Kunle Odutola and Anthony Oguntimehin. :)
- \* I (who, Ter?) added an antlr.build.Tool 'cause I hate ANT. This release does UNICODE properly now. Added limited lexical lookahead hoisting. Sather code generator disappears. Source changes for

Eclipse and NetBeans by Marco van Meegen and Brian Smith.

2.7.3 released March 22, 2004 was mainly a bug fix release,

- \* but included the parse-tree/derivation code to aid in debugging
- \* plus the cool TokenStreamRewriteEngine that makes rewriting or tweaking input files particularly easy.

2.7.4 released May 9, 2004 was mainly a bug fix release.

2.7.5 release Xmas 2004 had the following enhancements:

- \* A Python code generator has been implemented and contributed by Wolfgang Haefelinger and Marq Kole.
- \* A new make/autoconf framework as been contributed by Wolfgang Haefelinger
- \* A MSI based installer has been contributed by Wolfgang Haefelinger.

2.7.6 release xmas 2005 was mainly a bug fix release.

- \* Scott Stanchfield added file/line information for Java target and cleaned up a bunch of classloader stuff.
- \* Added stuff to support Prashant Deva's cool ANTLRStudio.

2.7.7 release September 7, 2006 was a bug fix release.

=====

README.txt - last update September 6th, 2006

Apache Commons  
Apache License v2

Apache Geronimo  
Apache License v2

Apache Jakarta ORO  
Apache License v2

Apache JMeter  
Apache License v2

Apache HTTP Server  
Apache License v2

Apache Maven  
Apache License v2

Apache Maven Doxia  
Apache License v2

Apache Portability Runtime  
Apache License v2

Apache Struts  
Apache License v2

## Text of Third-Party Software Licenses

Apache Struts  
Copyright 2000-2007 The Apache Software Foundation

This product includes software developed by  
The Apache Software Foundation (<http://www.apache.org/>).  
Dojo (<http://dojotoolkit.org/>).  
domTT (<http://www.mojavelinux.com/projects/domtooltip/>).

The binary distributions includes the following third party software:  
ANTLR (<http://www.antlr.org/>).  
Classworlds (<http://classworlds.codehaus.org/>).  
EZMorph (<http://ezmorph.sourceforge.net/>)  
FreeMarker (<http://freemarker.org/>).  
JSON-lib (<http://json-lib.sourceforge.net/>).  
OGNL (<http://www.opensymphony.com/ognl/>).  
Plexus (<http://plexus.codehaus.org/>).  
SiteMesh (<http://www.opensymphony.com/sitemesh/>).  
XWork (<http://www.opensymphony.com/xwork/>).  
XPP3 (<http://www.extreme.indiana.edu/xgws/xsoap/xpp/>).  
XStream (<http://xstream.codehaus.org/>).

Apache Tiles  
Apache License v2

Apache Tomcat Server  
Apache License v2

Apache Velocity  
Apache License v2

ASM

```
<html>
<!--
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000,2002,2003 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
```

\* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
 \* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
 \* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF  
 \* THE POSSIBILITY OF SUCH DAMAGE.

-->

<body>

Provides a small and fast bytecode manipulation framework.

<p>

The <a href="http://www.objectweb.org/asm">ASM</a> framework is organized around the {@link org.objectweb.asm.ClassVisitor ClassVisitor} and {@link org.objectweb.asm.CodeVisitor CodeVisitor} interfaces, which allows one to visit the fields and methods of a class, including the bytecode instructions of each method.

<p>

In addition to these two main interfaces, ASM provides a {@link org.objectweb.asm.ClassReader ClassReader} class, that can parse an existing class and make a given visitor visit it. ASM also provides a {@link org.objectweb.asm.ClassWriter ClassWriter} class, which is a visitor that generates Java class files.

<p>

In order to generate a class from scratch, only the {@link org.objectweb.asm.ClassWriter ClassWriter} class is necessary. Indeed, in order to generate a class, one must just call its `visit<i>XXX</i>` methods with the appropriate arguments to generate the desired fields and methods. See the "helloworld" example in the ASM distribution for more details about class generation.

<p>

In order to modify existing classes, one must use a {@link org.objectweb.asm.ClassReader ClassReader} class to analyse the original class, a class modifier, and a {@link org.objectweb.asm.ClassWriter ClassWriter} to construct the modified class. The class modifier is just a {@link org.objectweb.asm.ClassVisitor ClassVisitor} that delegates most of the work to another {@link org.objectweb.asm.ClassVisitor ClassVisitor}, but that sometimes changes some parameter values, or call additional methods, in order to implement the desired modification process. In order to make it easier to implement such class modifiers, ASM provides the {@link org.objectweb.asm.ClassAdapter ClassAdapter} and {@link org.objectweb.asm.CodeAdapter CodeAdapter} classes, which implement the {@link org.objectweb.asm.ClassVisitor ClassVisitor} and {@link org.objectweb.asm.CodeVisitor CodeVisitor} interfaces by delegating all work to other visitors. See the "adapt" example in the ASM distribution for more details about class modification.

<p>

The size of the core ASM library, <tt>asm.jar</tt>, is only 25KB, which is much more smaller than the size of the <a href="http://jakarta.apache.org/bcel">BCEL</a> library (350KB without the class verifier), and than the size of the <a href="http://serp.sourceforge.net">SERP</a> library (150KB). ASM is also much more faster than these tools. Indeed the overhead of a load time class transformation process is of the order of 60% with ASM, 700% or more with BCEL, and 1100% or more with SERP (see the <tt>test/perf</tt> directory in the ASM distribution)! kASM is a subset of ASM, for class generation only (it includes the {@link org.objectweb.asm.ClassWriter ClassWriter}, but not the {@link org.objectweb.asm.ClassReader ClassReader}). The size of this library,

## Text of Third-Party Software Licenses

<tt>kasm.jar</tt>, is only 16KB.

```
@since ASM 1.3
</body>
</html>
```

### Base64.c

libb64-1.1.src.tar.gz COPYRIGHT from <http://sourceforge.net/projects/libb64>

Copyright-Only Dedication (based on United States law)  
or Public Domain Certification

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

### Base64.java Apache License v2

```
/*
 * Copyright 2001-2005 The Apache Software Foundation.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

```
 */
/***
 * Provides Base64 encoding and decoding as defined by RFC 2045.
 *
 * <p>This class implements section <cite>6.8. Base64 Content-Transfer-Encoding</cite>
 * from RFC 2045 <cite>Multipurpose Internet Mail Extensions (MIME) Part One:
 * Format of Internet Message Bodies</cite> by Freed and Borenstein.</p>
 *
 * @see <a href="http://www.ietf.org/rfc/rfc2045.txt">RFC 2045</a>
 * @author Apache Software Foundation
 * @since 1.0-dev
 */

```

Bean Shell  
Lesser GNU Public License v2

Blosc Shuffling

Blosc - A blocking, shuffling and lossless compression library

Copyright (C) 2009-2010 Francesc Alted (faltet@pytables.org)

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

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

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

Boost

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

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

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative

## Text of Third-Party Software Licenses

works are solely in the form of machine-executable object code generated by a source language processor.

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

### BZip2

---

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2007 Julian R Seward. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

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

Julian Seward, [jseward@bzip.org](mailto:jseward@bzip.org)  
bzip2/libbzip2 version 1.0.5 of 10 December 2007

---

CG Lib  
Apache License v2

Codehaus Plexus  
Apache License v2

Colt

Packages cern.colt\* , cern.jet\*, cern.clhep

Copyright (c) 1999 CERN - European Organization for Nuclear Research.  
Permission to use, copy, modify, distribute and sell this software  
and its documentation for any purpose is hereby granted without fee,  
provided that the above copyright notice appear in all copies and  
that both that copyright notice and this permission notice appear in  
supporting documentation. CERN makes no representations about the  
suitability of this software for any purpose. It is provided "as is"  
without expressed or implied warranty.

Packages hep.aida.\*

Written by Pavel Binko, Dino Ferrero Merlino, Wolfgang Hoschek, Tony  
Johnson, Andreas Pfeiffer, and others. Check the FreeHEP home page  
for more info. Permission to use and/or redistribute this work is  
granted under the terms of the LGPL License, with the exception that  
any usage related to military applications is expressly forbidden.  
The software and documentation made available under the terms of this  
license are provided with no warranty.

ConfigIni.pm  
Artistic License

=head1 CHANGES

Version 0.01  
- Initial release.

=head1 AUTHOR

C. J. Kirsle <kirsle -at- rainbowboi.com>

=head1 COPYRIGHT AND LICENSE

Copyright (C) 2006 by A. U. Thor

This library is free software; you can redistribute it and/or modify  
it under the same terms as Perl itself, either Perl version 5.8.7 or,  
at your option, any later version of Perl 5 you may have available.

Confuse

```
* Copyright (c) 2002,2003,2007 Martin Hedenfalk <martin@bzero.se>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
```

## Text of Third-Party Software Licenses

```
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

Cpp Unit  
Lesser GNU Public License v2.1

Curl

### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2008, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

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

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

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

DateFormatter.js  
MIT License

```
/*
 * Date Format 1.2.3
 * (c) 2007-2009 Steven Levithan <stevenlevithan.com>
 * MIT license
 *
 * Includes enhancements by Scott Trenda <scott.trenda.net>
 * and Kris Kowal <cixar.com/~kris.kowal/>
 *
 * Accepts a date, a mask, or a date and a mask.
 * Returns a formatted version of the given date.
 * The date defaults to the current date/time.
 * The mask defaults to dateFormat.masks.default.
 *
 * From http://blog.stevenlevithan.com/archives/date-time-format
 */
```

Dialog.py  
Lesser GNU Public License v2.1

```
# dialog.py --- A python interface to the Linux "dialog" utility
```

```

# Copyright (C) 2000 Robb Shecter, Sultanbek Tezadov
# Copyright (C) 2002, 2003, 2004 Florent Rougon
#
# This library is free software; you can redistribute it and/or
# modify it under the terms of the GNU Lesser General Public
# License as published by the Free Software Foundation; either
# version 2.1 of the License, or (at your option) any later version.
#
# This library is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# Lesser General Public License for more details.
#
# You should have received a copy of the GNU Lesser General Public
# License along with this library; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

```

Direct Web Remoting  
Apache License v2

DOM4J

Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved.

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

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact [dom4j-info@metastuff.com](mailto:dom4j-info@metastuff.com).
4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.
5. Due credit should be given to the DOM4J Project - <http://www.dom4j.org>

THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

## Text of Third-Party Software Licenses

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### Eclipse

Eclipse Public License v1.0

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

#### 1. DEFINITIONS

Contribution means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
  - i) changes to the Program, and
  - ii) additions to the Program;where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

Contributor means any person or entity that distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement, including all Contributors.

#### 2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

### 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
  - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
  - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
  - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program. Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

### 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses

## Text of Third-Party Software Licenses

relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

### 5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement , including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

### 6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement

and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

#### EditLine

```
/*
 * Copyright (c) 1997 The NetBSD Foundation, Inc.
 * All rights reserved.
 *
 * This code is derived from software contributed to The NetBSD Foundation
 * by Jaromir Dolecek.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by the NetBSD
 *      Foundation, Inc. and its contributors.
 * 4. Neither the name of The NetBSD Foundation nor the names of its
 *    contributors may be used to endorse or promote products derived
 *    from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
 * ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
 * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
```

## Text of Third-Party Software Licenses

```
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
* POSSIBILITY OF SUCH DAMAGE.  
*/
```

Ehcache  
Apache License v2

Enterprise JavaBeans

EJB 3.0 License  
Specification: JSR-000220 Enterprise JavaBeans v.3.0 ("Specification")  
Version: 3.0  
Status: Final Release  
Release: 8 May 2006  
Copyright 2006 SUN MICROSYSTEMS, INC.  
4150 Network Circle, Santa Clara, California 95054, U.S.A  
All rights reserved.  
LIMITED LICENSE GRANTS

1. License for Evaluation Purposes. Sun hereby grants you a fully-paid, non-exclusive, non-transferable, worldwide, limited license (without the right to sublicense), under Sun's applicable intellectual property rights to view, download, use and reproduce the Specification only for the purpose of internal evaluation. This includes (i) developing applications intended to run on an implementation of the Specification, provided that such applications do not themselves implement any portion(s) of the Specification, and (ii) discussing the Specification with any third party; and (iii) excerpting brief portions of the Specification in oral or written communications which discuss the Specification provided that such excerpts do not in the aggregate constitute a significant portion of the Specification.

2. License for the Distribution of Compliant Implementations. Sun also grants you a perpetual, nonexclusive, non-transferable, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) under any applicable copyrights or, subject to the provisions of subsection 4 below, patent rights it may have covering the Specification to create and/or distribute an Independent Implementation of the Specification that: (a) fully implements the Specification including all its required interfaces and functionality; (b) does not modify, subset, superset or otherwise extend the Licensor Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the Licensor Name Space other than those required/authorized by the Specification or Specifications being implemented; and (c) passes the Technology Compatibility Kit (including satisfying the requirements of the applicable TCK Users Guide) for such Specification ("Compliant Implementation"). In addition, the foregoing license is expressly conditioned on your not acting outside its scope. No license is granted hereunder for any other purpose (including, for example, modifying the Specification, other than to the extent of your fair use rights, or distributing the Specification to third parties). Also, no right, title, or interest in or to any trademarks, service marks, or trade names of Sun or Sun's licensors is granted hereunder. Java, and Java-related logos, marks and names are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

3. Pass-through Conditions. You need not include limitations (a)-(c) from the previous paragraph or any other particular "pass through" requirements in any license You grant concerning the use of your Independent Implementation or products derived from it. However, except with respect to Independent Implementations (and products derived from them) that

satisfy limitations (a)-(c) from the previous paragraph, You may neither: (a) grant or otherwise pass through to your licensees any licenses under Sun's applicable intellectual property rights; nor (b) authorize your licensees to make any claims concerning their implementation's compliance with the Specification in question.

#### 4. Reciprocity Concerning Patent Licenses.

- a. With respect to any patent claims covered by the license granted under subparagraph 2 above that would be infringed by all technically feasible implementations of the Specification, such license is conditioned upon your offering on fair, reasonable and non-discriminatory terms, to any party seeking it from You, a perpetual, non-exclusive, non-transferable, worldwide license under Your patent rights which are or would be infringed by all technically feasible implementations of the Specification to develop, distribute and use a Compliant Implementation.
- b. With respect to any patent claims owned by Sun and covered by the license granted under subparagraph 2, whether or not their infringement can be avoided in a technically feasible manner when implementing the Specification, such license shall terminate with respect to such claims if You initiate a claim against Sun that it has, in the course of performing its responsibilities as the Specification Lead, induced any other entity to infringe Your patent rights.
- c. Also with respect to any patent claims owned by Sun and covered by the license granted under subparagraph 2 above, where the infringement of such claims can be avoided in a technically feasible manner when implementing the Specification such license, with respect to such claims, shall terminate if You initiate a claim against Sun that its making, having made, using, offering to sell, selling or importing a Compliant Implementation infringes Your patent rights.

5. Definitions. For the purposes of this Agreement: "Independent Implementation" shall mean an implementation of the Specification that neither derives from any of Sun's source code or binary code materials nor, except with an appropriate and separate license from Sun, includes any of Sun's source code or binary code materials; "Licensor Name Space" shall mean the public class or interface declarations whose names begin with "java", "javax", "com.sun" or their equivalents in any subsequent naming convention adopted by Sun through the Java Community Process, or any recognized successors or replacements thereof; and "Technology Compatibility Kit" or "TCK" shall mean the test suite and accompanying TCK User's Guide provided by Sun which corresponds to the Specification and that was available either (i) from Sun 120 days before the first release of Your Independent Implementation that allows its use for commercial purposes, or (ii) more recently than 120 days from such release but against which You elect to test Your implementation of the Specification.

This Agreement will terminate immediately without notice from Sun if you breach the Agreement or act outside the scope of the licenses granted above.

#### DISCLAIMER OF WARRANTIES

THE SPECIFICATION IS PROVIDED "AS IS". SUN MAKES NO REPRESENTATIONS OR WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT (INCLUDING AS A CONSEQUENCE OF ANY PRACTICE OR IMPLEMENTATION OF THE SPECIFICATION), OR THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE. This document does not represent any commitment to release or implement any portion of the Specification in any product. In addition, the Specification could include technical inaccuracies or typographical errors.

#### LIMITATION OF LIABILITY

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUE, PROFITS OR DATA, OR FOR SPECIAL,

## Text of Third-Party Software Licenses

INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED IN ANY WAY TO YOUR HAVING, IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION, EVEN IF SUN AND/OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You will indemnify, hold harmless, and defend Sun and its licensors from any claims arising or resulting from: (i) your use of the Specification; (ii) the use or distribution of your Java application, applet and/or implementation; and/or (iii) any claims that later versions or releases of any Specification furnished to you are incompatible with the Specification provided to you under this license.

### RESTRICTED RIGHTS LEGEND

U.S. Government: If this Specification is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in the Software and accompanying documentation shall be only as set forth in this license; this is in accordance with 48 C.F.R. 227.7201 through 227.7202-4 (for Department of Defense (DoD) acquisitions) and with 48 C.F.R. 2.101 and 12.212 (for non-DoD acquisitions).

### REPORT

If you provide Sun with any comments or suggestions concerning the Specification ("Feedback"), you hereby: (i) agree that such Feedback is provided on a non-proprietary and non-confidential basis, and (ii) grant Sun a perpetual, non-exclusive, worldwide, fully paid-up, irrevocable license, with the right to sublicense through multiple levels of sublicensees, to incorporate, disclose, and use without limitation the Feedback for any purpose.

### GENERAL TERMS

Any action related to this Agreement will be governed by California law and controlling U.S. federal law. The U.N. Convention for the International Sale of Goods and the choice of law rules of any jurisdiction will not apply.

The Specification is subject to U.S. export control laws and may be subject to export or import regulations in other countries. Licensee agrees to comply strictly with all such laws and regulations and acknowledges that it has the responsibility to obtain such licenses to export, re-export or import as may be required after delivery to Licensee.

This Agreement is the parties' entire agreement relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, conditions, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification to this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

Rev. April, 2006

Sun/Final/Full

Excanvas.js  
Apache License v2

```
// Copyright 2006 Google Inc.  
//
```

```
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//   http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
```

#### Expat

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd  
and Clark Cooper  
Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers.

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

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

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

#### EZMorph Apache License v2

#### Free Marker

FreeMarker 1.x was released under the LGPL license. Later, by community consensus, we have switched over to a BSD-style license. As of FreeMarker 2.2pre1, the original author, Benjamin Geer, has relinquished the copyright in behalf of Visigoth Software Society. The current copyright holder is the Visigoth Software Society.

-----  
Copyright (c) 2003 The Visigoth Software Society. All rights reserved.

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

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

## Text of Third-Party Software Licenses

2. The end-user documentation included with the redistribution, if any, must include the following acknowledgement:

"This product includes software developed by the Visigoth Software Society (<http://www.visigoths.org/>)."

Alternately, this acknowledgement may appear in the software itself, if and wherever such third-party acknowledgements normally appear.
3. Neither the name "FreeMarker", "Visigoth", nor any of the names of the project contributors may be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [visigoths@visigoths.org](mailto:visigoths@visigoths.org).
4. Products derived from this software may not be called "FreeMarker" or "Visigoth" nor may "FreeMarker" or "Visigoth" appear in their names without prior written permission of the Visigoth Software Society.

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

---

This software consists of voluntary contributions made by many individuals on behalf of the Visigoth Software Society. For more information on the Visigoth Software Society, please see <http://www.visigoths.org/>

## FreeType

### The FreeType Project LICENSE

---

2006-Jan-27

Copyright 1996-2002, 2006 by  
David Turner, Robert Wilhelm, and Werner Lemberg

## Introduction

---

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG

(Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- o We don't promise that this software works. However, we will be interested in any kind of bug reports. ('as is' distribution)
- o You can use this software for whatever you want, in parts or full form, without having to pay us. ('royalty-free' usage)
- o You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. ('credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```
"""
Portions of this software are copyright ? <year> The FreeType
Project (www.freetype.org). All rights reserved.
"""
```

Please replace <year> with the value from the FreeType version you actually use.

## Legal Terms

---

### 0. Definitions

---

Throughout this license, the terms 'package', 'FreeType Project', and 'FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the 'FreeType Project', be they named as alpha, beta or final release.

'You' refers to the licensee, or person using the project, where 'using' is a generic term including compiling the project's source code as well as linking it to form a 'program' or 'executable'. This program is referred to as 'a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

## Text of Third-Party Software Licenses

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

### 1. No Warranty

-----

THE FREETYPE PROJECT IS PROVIDED 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

### 2. Redistribution

-----

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- o Redistribution of source code must retain this license file ('FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.
- o Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

### 3. Advertising

-----

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: 'FreeType Project', 'FreeType Engine', 'FreeType library', or 'FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it.

Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

#### 4. Contacts

-----

There are two mailing lists related to FreeType:

- o [freetype@nongnu.org](mailto:freetype@nongnu.org)

Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- o [freetype-devel@nongnu.org](mailto:freetype-devel@nongnu.org)

Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

<http://www.freetype.org>

--- end of FTL.TXT ---

Fuse  
 GNU Public License v2  
 Lesser GNU Public License v2

Ganglia  
 GNU Public License v2

Copyright (c) 2001, 2002, 2003, 2004, 2005 by  
 The Regents of the University of California. All rights reserved.

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

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

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

```
// +-----+  

// | TemplatePower:  

// | offers you the ability to separate your PHP code and your HTML  

// +-----+
```

## Text of Third-Party Software Licenses

```
// Copyright (C) 2001,2002 R.P.J. Velzeboer, The Netherlands
//
// This program is free software; you can redistribute it and/or
// modify it under the terms of the GNU General Public License
// as published by the Free Software Foundation; either version 2
// of the License, or (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with this program; if not, write to the Free Software
// Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
// 02111-1307, USA.
//
// Author: R.P.J. Velzeboer, rovel@codocad.nl The Netherlands
//
// +-----+
// | http://templatepower.codocad.com
// +-----+
```

GNU GCC  
GNU Public License v2  
Lesser GNU Public License v2.1

GNU GDB  
GNU Public License v3

GNU C Library  
GNU Public License v2  
Lesser GNU Public License v2.1

GNU Scientific Library  
GNU Public License v3

Google Gson  
Apache License v2

Copyright (c) 2008-2009 Google Inc.

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

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

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

Google Perf Tools

Copyright (c) 2005, Google Inc.  
All rights reserved.

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

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

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

Gson Pretty Printer  
Lesser GNU Public License v3

```
/***
 * A decent pretty-printer for Gson. Features include:
 *
 * <ul>
 * <li>Will try to use a single line where possible</li>
 * <li>Configurable indent string</li>
 * </ul>
 *
 * This file is (c) Copyright 2009 Uprizer Labs LLC It is released to the public under the
 Lesser GNU Public License
 * v3.0 See http://www.gnu.org/licenses/lgpl.html for details.
 *
 * @author Ian Clarke <ian@uprizer.com>, minor additions by Eric Poggen (NuOrbit Media,
 nuorbit.com)
 *
 * new GsonPrettyPrinter(gb.create()).ppJson(some_object);
 */
```

GTK/GNOME GLib  
GNU Public License v2

Hadoop  
Apache License v2

This product includes software developed by The Apache Software

## Text of Third-Party Software Licenses

Foundation (<http://www.apache.org/>) .

Hibernate  
Lesser GNU Public License v2.1

Hibernate Middle Gen

Copyright (c) 2001, Aslak Hellesoy, BEKK Consulting  
All rights reserved.

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

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

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

HTML Parser  
Common Public License 1.0

iODBC

iODBC Driver Manager  
Copyright (C) 1995 by Ke Jin <[kejin@empress.com](mailto:kejin@empress.com)>  
Copyright (C) 1996-2009 by OpenLink Software <[iodbc@openlinksw.com](mailto:iodbc@openlinksw.com)>  
All Rights Reserved.

This software is released under either the GNU Library General Public License (see LICENSE.LGPL) or the BSD License (see LICENSE.BSD).

Note that the only valid version of the LGPL license as far as this project is concerned is the original GNU Library General Public License Version 2, dated June 1991.

While not mandated by the BSD license, any patches you make to the iODBC may be contributed back into the iODBC project at your discretion. Contributions will benefit the Open Source and Data Access community as a whole. Submissions may be made at <http://www.iodbc.org>.

Copyright (C) 1995-2009, OpenLink Software Inc and Ke Jin.  
All rights reserved.

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

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

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

#### IOzone

"a filesystem benchmark tool. The benchmark generates and measures a variety of file operations. Iozone has been ported to many machines and runs under many operating systems." See <http://www.iozone.org/>. This material is distributed under a license that states:

Copyright 1991, 1992, 1994, 1998, 1999, 2002 William D. Norcott License to freely use and distribute this software is hereby granted by the author, subject to the condition that this copyright notice remains intact. The author retains the exclusive right to publish derivative works based on this work, including, but not limited to, revised versions of this work.

#### IPerf

Copyright (c) 1999-2007, The Board of Trustees of the University of Illinois All Rights Reserved.

Iperf performance test  
 Mark Gates  
 Ajay Tirumala  
 Jim Ferguson  
 Jon Dugan  
 Feng Qin  
 Kevin Gibbs  
 John Estabrook  
 National Laboratory for Applied Network Research  
 National Center for Supercomputing Applications

## Text of Third-Party Software Licenses

University of Illinois at Urbana-Champaign  
<http://www.ncsa.uiuc.edu>

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

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

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

Neither the names of the University of Illinois, NCSA, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## JAMA - Java Matrix Package

From <http://math.nist.gov/javanumerics/jama/>

JAMA's initial design, as well as this reference implementation, was developed by

Joe Hicklin  
Cleve Moler  
Peter Webb

... from The MathWorks  
Ronald F. Boisvert  
Bruce Miller  
Roldan Pozo  
Karin Remington

... from NIST

Copyright Notice This software is a cooperative product of The MathWorks and the National Institute of Standards and Technology (NIST) which has been released to the public domain. Neither The MathWorks nor NIST assumes any responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

## JavaBeans Activation Framework Common Dev and Distrib 1.0

## Java Arch for XML Binding

Common Dev and Distrib 1.0

Java Mail

Common Dev and Distrib 1.0

Java Persistence API

Common Dev and Distrib 1.0

Java Secure Channel (JSCH)

JSch 0.0.\* was released under the GNU LGPL license. Later, we have switched over to a BSD-style license.

---

Copyright (c) 2002-2010 Atsuhiko Yamanaka, JCRAFT, Inc.

All rights reserved.

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

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

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

Java Servlet API

Java Servlet License

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THIS SPECIFICATION TO YOU ONLY FOR EVALUATION PURPOSES AND ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS AGREEMENT. PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY. BY DOWNLOADING THIS SPECIFICATION, YOU ACCEPT THE TERMS AND CONDITIONS OF THE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY IT, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THIS PAGE.

Specification: JSR-000154 Java<sup>(tm)</sup> Servlet 2.5  
("Specification")

## Text of Third-Party Software Licenses

Version: 2.5

Status: Maintenance Release 2

Release: 28 August 2007

Copyright 2007 SUN MICROSYSTEMS, INC.  
All rights reserved.

### LIMITED EVALUATION LICENSE

Sun hereby grants you a fully-paid, non-exclusive, non-transferable, worldwide, limited license (without the right to sublicense), under Sun's applicable intellectual property rights to view, download, use and reproduce the Specification only for the purpose of internal evaluation. This includes (i) developing applications intended to run on an implementation of the Specification, provided that such applications do not themselves implement any portion(s) of the Specification, and (ii) excerpting brief portions of the Specification in oral or written communications which discuss the Specification provided that such excerpts do not in the aggregate constitute a significant portion of the Technology. No license of any kind is granted hereunder for any other purpose including, for example, creating and distributing implementations of the Specification, modifying the Specification (other than to the extent of your fair use rights), or distributing the Specification to third parties. Also, no right, title, or interest in or to any trademarks, service marks, or trade names of Sun or Sun's licensors, Sun or the Sun's licensors is granted hereunder. If you wish to create and distribute an implementation of the Specification, a license for that purpose is available at <http://www.jcp.org>. The foregoing license is expressly conditioned on your acting within its scope, and will terminate immediately without notice from Sun if you breach the Agreement or act outside the scope of the licenses granted above. Java, and Java-related logos, marks and names are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

### DISCLAIMER OF WARRANTIES

THE SPECIFICATION IS PROVIDED "AS IS". SUN MAKES NO REPRESENTATIONS OR WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT (INCLUDING AS A CONSEQUENCE OF ANY PRACTICE OR IMPLEMENTATION OF THE SPECIFICATION), OR THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE. This document does not represent any commitment to release or implement any portion of the Specification in any product. In addition, the Specification could include technical inaccuracies or typographical errors.

#### LIMITATION OF LIABILITY

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUE, PROFITS OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, RELATED IN ANY WAY TO YOUR HAVING OR USING THE SPECIFICATION, EVEN IF SUN AND/OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### RESTRICTED RIGHTS LEGEND

U.S. Government: If this Specification is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in the Software and accompanying documentation shall be only as set forth in this license; this is in accordance with 48 C.F.R. 227.7201 through 227.7202-4 (for Department of Defense (DoD) acquisitions) and with 48 C.F.R. 2.101 and 12.212 (for non-DoD acquisitions).

#### REPORT

If you provide Sun with any comments or suggestions concerning the Specification ("Feedback"), you hereby: (i) agree that such Feedback is provided on a non-proprietary and non-confidential basis, and (ii) grant Sun a perpetual, non-exclusive, worldwide, fully paid-up, irrevocable license, with the right to sublicense through multiple levels of sublicensees, to incorporate, disclose, and use without limitation the Feedback for any purpose.

#### GOVERNING LAW

Any action relating to or arising out of this Agreement will be governed by California law and controlling U.S. federal law. The U.N. Convention for the International Sale of Goods and the choice of law rules of any jurisdiction will not apply.

Rev. January 2006

#### Java Transaction API

##### JTA License

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THIS SPECIFICATION TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS AGREEMENT. PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY. BY DOWNLOADING THIS SPECIFICATION, YOU ACCEPT THE TERMS AND CONDITIONS OF THE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY

## Text of Third-Party Software Licenses

IT, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THIS PAGE.

Specification: JSR-000907 Java(tm) Transaction API (JTA) Specification ("Specification")

Version: 1.1

Status: Maintenance Release

Release: 14 February 2007

Copyright 2007 SUN MICROSYSTEMS, INC.  
4150 Network Circle, Santa Clara, California 95054,  
U.S.A  
All rights reserved.

### LIMITED LICENSE GRANTS

1. License for Evaluation Purposes. Sun hereby grants you a fully-paid, non-exclusive, non-transferable, worldwide, limited license (without the right to sublicense), under Sun's applicable intellectual property rights to view, download, use and reproduce the Specification only for the purpose of internal evaluation. This includes (i) developing applications intended to run on an implementation of the Specification, provided that such applications do not themselves implement any portion(s) of the Specification, and (ii) discussing the Specification with any third party; and (iii) excerpting brief portions of the Specification in oral or written communications which discuss the Specification provided that such excerpts do not in the aggregate constitute a significant portion of the Specification.

2. License for the Distribution of Compliant Implementations. Sun also grants you a perpetual, non-exclusive, non-transferable, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) under any applicable copyrights or, subject to the provisions of subsection 4 below, patent rights it may have covering the Specification to create and/or distribute an Independent Implementation of the Specification that: (a) fully implements the Specification including all its required interfaces and functionality; (b) does not modify, subset, superset or otherwise extend the Licensor Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the Licensor Name Space other than those required/authorized by the Specification or Specifications being implemented; and (c) passes the Technology Compatibility Kit (including satisfying the requirements of the applicable TCK Users Guide) for such Specification ("Compliant Implementation"). In addition, the foregoing license is expressly conditioned on your not acting outside its scope. No license is granted hereunder for any other purpose

(including, for example, modifying the Specification, other than to the extent of your fair use rights, or distributing the Specification to third parties). Also, no right, title, or interest in or to any trademarks, service marks, or trade names of Sun or Sun's licensors is granted hereunder. Java, and Java-related logos, marks and names are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

3. Pass-through Conditions. You need not include limitations (a)-(c) from the previous paragraph or any other particular "pass through" requirements in any license You grant concerning the use of your Independent Implementation or products derived from it. However, except with respect to Independent Implementations (and products derived from them) that satisfy limitations (a)-(c) from the previous paragraph, You may neither: (a) grant or otherwise pass through to your licensees any licenses under Sun's applicable intellectual property rights; nor (b) authorize your licensees to make any claims concerning their implementation's compliance with the Specification in question.

#### 4. Reciprocity Concerning Patent Licenses.

a. With respect to any patent claims covered by the license granted under subparagraph 2 above that would be infringed by all technically feasible implementations of the Specification, such license is conditioned upon your offering on fair, reasonable and non-discriminatory terms, to any party seeking it from You, a perpetual, non-exclusive, non-transferable, worldwide license under Your patent rights which are or would be infringed by all technically feasible implementations of the Specification to develop, distribute and use a Compliant Implementation.

b With respect to any patent claims owned by Sun and covered by the license granted under subparagraph 2, whether or not their infringement can be avoided in a technically feasible manner when implementing the Specification, such license shall terminate with respect to such claims if You initiate a claim against Sun that it has, in the course of performing its responsibilities as the Specification Lead, induced any other entity to infringe Your patent rights.

c Also with respect to any patent claims owned by Sun and covered by the license granted under subparagraph 2 above, where the infringement of such claims can be avoided in a technically feasible manner when implementing the Specification such license, with respect to such claims, shall terminate if You initiate a claim against Sun that its making, having made, using, offering to sell, selling or importing a Compliant Implementation infringes Your patent rights.

5. Definitions. For the purposes of this Agreement:  
Independent Implementation shall mean an implementation of the Specification that neither derives from any of Sun's source code or binary code materials nor, except with an appropriate and separate license from Sun, includes any of Sun's source code or binary code materials; "Licensor Name Space" shall mean the public class or interface declarations whose names begin with "java", "javax", "com.sun" or their equivalents in any subsequent naming convention adopted by Sun through the Java Community Process, or any recognized successors or replacements thereof; and Technology Compatibility Kit or "TCK" shall mean the test suite and accompanying TCK User's Guide provided by Sun which corresponds to the Specification and that was available either (i) from Sun's 120 days before the first release of Your Independent Implementation that allows its use for commercial purposes, or (ii) more recently than 120 days from such release but against which You elect to test Your implementation of the Specification.

This Agreement will terminate immediately without notice from Sun if you breach the Agreement or act outside the scope of the licenses granted above.

#### DISCLAIMER OF WARRANTIES

THE SPECIFICATION IS PROVIDED "AS IS". SUN MAKES NO REPRESENTATIONS OR WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT (INCLUDING AS A CONSEQUENCE OF ANY PRACTICE OR IMPLEMENTATION OF THE SPECIFICATION), OR THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE. This document does not represent any commitment to release or implement any portion of the Specification in any product. In addition, the Specification could include technical inaccuracies or typographical errors.

#### LIMITATION OF LIABILITY

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUE, PROFITS OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED IN ANY WAY TO YOUR HAVING, IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION, EVEN IF SUN AND/OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You will indemnify, hold harmless, and defend Sun and its licensors from any claims arising or resulting from: (i) your use of the Specification; (ii) the use or distribution of your Java application, applet and/or implementation; and/or (iii) any claims that

later versions or releases of any Specification furnished to you are incompatible with the Specification provided to you under this license.

#### RESTRICTED RIGHTS LEGEND

U.S. Government: If this Specification is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in the Software and accompanying documentation shall be only as set forth in this license; this is in accordance with 48 C.F.R. 227.7201 through 227.7202-4 (for Department of Defense (DoD) acquisitions) and with 48 C.F.R. 2.101 and 12.212 (for non-DoD acquisitions).

#### REPORT

If you provide Sun with any comments or suggestions concerning the Specification ("Feedback"), you hereby: (i) agree that such Feedback is provided on a non-proprietary and non-confidential basis, and (ii) grant Sun a perpetual, non-exclusive, worldwide, fully paid-up, irrevocable license, with the right to sublicense through multiple levels of sublicensees, to incorporate, disclose, and use without limitation the Feedback for any purpose.

#### GENERAL TERMS

Any action related to this Agreement will be governed by California law and controlling U.S. federal law. The U.N. Convention for the International Sale of Goods and the choice of law rules of any jurisdiction will not apply.

The Specification is subject to U.S. export control laws and may be subject to export or import regulations in other countries. Licensee agrees to comply strictly with all such laws and regulations and acknowledges that it has the responsibility to obtain such licenses to export, re-export or import as may be required after delivery to Licensee.

This Agreement is the parties' entire agreement relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, conditions, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification to this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

Rev. April, 2006

Java XML Web Services  
Common Dev and Distrib 1.0

JavaScript Templating  
MIT License

Javascript Micro-Templating

Copyright John Resig - <http://ejohn.org/> - MIT Licensed

JBoss Java Assist

Mozilla Public License v1.1

MOZILLA PUBLIC LICENSE

Version 1.1

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "'Contributor'" means each entity that creates or contributes to the creation of Modifications.

1.2. "'Contributor Version'" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "'Covered Code'" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "'Electronic Distribution Mechanism'" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "'Executable'" means Covered Code in any form other than Source Code.

1.6. "'Initial Developer'" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "'Larger Work'" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "'License'" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "'Modifications'" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. Source Code License.

### 2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and  
 (b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

### 2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with

## Text of Third-Party Software Licenses

its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

### 3. Distribution Obligations.

#### 3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

#### 3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

#### 3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

### 3.4. Intellectual Property Matters

#### (a) Third Party Claims.

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If

Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the `LEGAL` file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the `LEGAL` file.

(c) Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

**3.5. Required Notices.**

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

**3.6. Distribution of Executable Versions.**

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

**3.7. Larger Works.**

You may create a Larger Work by combining Covered Code with other code not

## Text of Third-Party Software Licenses

governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

### 4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the `LEGAL` file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

### 5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

### 6. Versions of the License.

#### 6.1. New Versions.

Netscape Communications Corporation (''Netscape'') may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

#### 6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

#### 6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases ''Mozilla'', ''MOZILLAPL'', ''MOZPL'', ''Netscape'', "MPL", ''NPL'' or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

### 7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

### 8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim

(excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

#### 9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

#### 10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

#### 11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This

## Text of Third-Party Software Licenses

License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

### 12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

### 13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as ?Multiple-Licensed??. ?Multiple-Licensed? means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the MPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

## EXHIBIT A -Mozilla Public License.

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Javassist.

The Initial Developer of the Original Code is Shigeru Chiba. Portions created by the Initial Developer are

Copyright (C) 1999-2006 Shigeru Chiba. All Rights Reserved.

Contributor(s) : \_\_\_\_\_.

Alternatively, the contents of this file may be used under the terms of the GNU Lesser General Public License Version 2.1 or later (the "LGPL"), in which case the provisions of the LGPL are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the LGPL, and not to allow others to use your version of this file under the terms of the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the LGPL. If you do not delete the provisions above, a recipient may use your version of this file under the terms of either the MPL or the LGPL.

Jetty  
Apache License v2

```
=====
Jetty Web Container
Copyright 1995-2006 Mort Bay Consulting Pty Ltd
=====
```

This product includes some software developed at The Apache Software Foundation (<http://www.apache.org/>).

The javax.servlet package used by Jetty is copyright Sun Microsystems, Inc and Apache Software Foundation. It is distributed under the Common Development and Distribution License. You can obtain a copy of the license at <https://glassfish.dev.java.net/public/CDDLv1.0.html>.

The UnixCrypt.java code ~Implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville.

The default JSP implementation is provided by the Glassfish JSP engine from project Glassfish <http://glassfish.dev.java.net>. Copyright 2005 Sun Microsystems, Inc. and portions Copyright Apache Software Foundation.

Some portions of the code are Copyright:

2006 Tim Vernum  
1999 Jason Gilbert.

The jboss integration module contains some LGPL code.

## JQuery

Copyright (c) 2011 John Resig, <http://jquery.com/>

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

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

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

## JQuery BGI Frame MIT License

```
/* Copyright (c) 2006 Brandon Aaron (http://brandonaaron.net)
 * Dual licensed under the MIT (http://www.opensource.org/licenses/mit-license.php)
 * and GPL (http://www.opensource.org/licenses/gpl-license.php) licenses.
```

## Text of Third-Party Software Licenses

```
/*
 * $LastChangedDate: 2007-07-21 18:45:56 -0500 (Sat, 21 Jul 2007) $
 * $Rev: 2447 $
 *
 * Version 2.1.1
 */
```

### JQuery Beauty Tips MIT License

```
/*
 * @name BeautyTips
 * @desc a tooltips/balloon-help plugin for jQuery
 *
 * @author Jeff Robbins - Lullabot - http://www.lullabot.com
 * @version 0.9.5 release candidate 1 (5/20/2009)
 */
```

### JQuery Dimensions MIT License

```
/* Copyright (c) 2007 Paul Bakaus (paul.bakaus@googlemail.com) and Brandon Aaron
(brandon.aaron@gmail.com || http://brandonaaron.net)
 * Dual licensed under the MIT (http://www.opensource.org/licenses/mit-license.php)
 * and GPL (http://www.opensource.org/licenses/gpl-license.php) licenses.
 *
 * $LastChangedDate: 2007-12-20 08:43:48 -0600 (Thu, 20 Dec 2007) $
 * $Rev: 4257 $
 *
 * Version: 1.2
 *
 * Requires: jQuery 1.2+
```

### JQuery Flot MIT License

```
/* Javascript plotting library for jQuery, v. 0.6.
 *
 * Released under the MIT license by IOLA, December 2007.
 *
 */
// first an inline dependency, jquery.colorhelpers.js, we inline it here
// for convenience

/* Plugin for jQuery for working with colors.
 *
 * Version 1.0.
 *
 * Inspiration from jQuery color animation plugin by John Resig.
 *
 * Released under the MIT license by Ole Laursen, October 2009.
 */
```

### JQuery Highlight 3

## MIT License

```
/*
highlight v3

Highlights arbitrary terms.

<http://johannburkard.de/blog/programming/javascript/highlight-javascript-text-
highlighting-jquery-plugin.html>

MIT license.

Johann Burkard
<http://johannburkard.de>
<mailto:jb@eaio.com>

*/
```

## JQuery Hover Intent

## MIT License

```
* hoverIntent r5 // 2007.03.27 // jQuery 1.1.2
* <http://cherne.net/brian/resources/jquery.hoverIntent.html>
*
* hoverIntent is currently available for use in all personal or commercial
* projects under both MIT and GPL licenses. This means that you can choose
* the license that best suits your project, and use it accordingly.
```

## JQuery Multi-Select

## MIT License

```
/*
// jQuery multiSelect
//
// Version 1.2.2 beta
//
// Cory S.N. LaViska
// A Beautiful Site (http://abeautifulsite.net/)
// 09 September 2009
//
// Visit http://abeautifulsite.net/notebook/62 for more information
//
// (Amended by Andy Richmond, Letters & Science Deans' Office, University of California,
Davis)
//
...
//
// Licensing & Terms of Use
//
// This plugin is dual-licensed under the GNU General Public License and the MIT License and
// is copyright 2008 A Beautiful Site, LLC.
//
*/
```

## JQuery QTip

## Text of Third-Party Software Licenses

### MIT License

```
/*
 * jquery.qtip. The jQuery tooltip plugin
 *
 * Copyright (c) 2009 Craig Thompson
 * http://craigsworks.com
 *
 * Licensed under MIT
 * http://www.opensource.org/licenses/mit-license.php
 *
 * Launch : February 2009
 * Version : 1.0.0-rc3
 * Released: Tuesday 12th May, 2009 - 00:00
 * Debug: jquery.qtip.debug.js
 */
```

### JQuery Scroll To MIT License

```
/**
 * jQuery.ScrollTo - Easy element scrolling using jQuery.
 * Copyright (c) 2007-2009 Ariel Flesler - aflesler(at)gmail(dot)com | http://
flesler.blogspot.com
 * Dual licensed under MIT and GPL.
 * Date: 5/25/2009
 * @author Ariel Flesler
 * @version 1.4.2
 *
 * http://flesler.blogspot.com/2007/10/jqueryscrollto.html
 */
```

### JQuery Sparkline

```
* jquery.sparkline.js
*
* v1.4.3
* (c) Splunk, Inc
* Contact: Gareth Watts (gareth@splunk.com)
* http://omnipotent.net/jquery.sparkline/
*
* Generates inline sparkline charts from data supplied either to the method
* or inline in HTML
*
* Compatible with Internet Explorer 6.0+ and modern browsers equipped with the canvas tag
* (Firefox 2.0+, Safari, Opera, etc)
*
* License: New BSD License
*
* Copyright (c) 2009, Splunk Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
*
*      * Redistributions of source code must retain the above copyright notice,
*      this list of conditions and the following disclaimer.
```

```

*      * Redistributions in binary form must reproduce the above copyright notice,
*      this list of conditions and the following disclaimer in the documentation
*      and/or other materials provided with the distribution.
*      * Neither the name of Splunk Inc nor the names of its contributors may
*      be used to endorse or promote products derived from this software without
*      specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
* SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
* OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

#### JQuery Tool Tip

MIT License

```

/*
 * jQuery Tooltip plugin 1.1
 *
 * http://bassistance.de/jquery-plugins/jquery-plugin-tooltip/
 *
 * Copyright (c) 2006 J?rn Zaefferer, Stefan Petre
 *
 * Dual licensed under the MIT and GPL licenses:
 *   http://www.opensource.org/licenses/mit-license.php
 *   http://www.gnu.org/licenses/gpl.html
 *
 * Revision: $Id: jquery.tooltip.js 2237 2007-07-04 19:11:15Z joern.zaefferer $
 */

```

#### JQuery Tree Table

MIT License

```

/*
 * jQuery treeTable Plugin 2.3.0
 * http://ludo.cubicphuse.nl/jquery-plugins/treeTable/
 *
 * Copyright 2010, Ludo van den Boom
 * Dual licensed under the MIT or GPL Version 2 licenses.
 */

```

#### JQuery UI

MIT License

```

/*
 * jQuery UI 1.7.3
 *
 * Copyright (c) 2009 AUTHORS.txt (http://jqueryui.com/about)
 * Dual licensed under the MIT (MIT-LICENSE.txt)
 * and GPL (GPL-LICENSE.txt) licenses.
 */

```

## Text of Third-Party Software Licenses

\* <http://docs.jquery.com/UI>

### JQuery Validate MIT License

```
/*
 * jQuery validation plug-in 1.6
 *
 * http://bassistance.de/jquery-plugins/jquery-plugin-validation/
 * http://docs.jquery.com/Plugins/Validation
 *
 * Copyright (c) 2006 - 2008 Jörn Zaefferer
 *
 * $Id: jquery.validate.js 6403 2009-06-17 14:27:16Z joern.zaefferer $
 *
 * Dual licensed under the MIT and GPL licenses:
 *   http://www.opensource.org/licenses/mit-license.php
 *   http://www.gnu.org/licenses/gpl.html
 */
```

### Json C MIT License

Copyright (c) 2004, 2005 Metaparadigm Pte Ltd

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

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

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

Portions of Json C:

Copyright (c) 2008-2009 Yahoo! Inc. All rights reserved.  
The copyrights to the contents of this file are licensed under the MIT License  
(<http://www.opensource.org/licenses/mit-license.php>)

### Json Cpp

The json-cpp library and this documentation are in Public Domain.

### Json Lib Apache License v2

JSP Standard Tag Library

JSTL License

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THIS SPECIFICATION TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS LICENSE AGREEMENT ("AGREEMENT"). PLEASE READ THE TERMS AND CONDITIONS OF THIS LICENSE CAREFULLY. BY DOWNLOADING THIS SPECIFICATION, YOU ACCEPT THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY ITS TERMS, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THIS PAGE AND THE DOWNLOADING PROCESS WILL NOT CONTINUE.

JavaServer Pages(TM) Standard Tag Library (JSTL) Specification ("Specification")

Version: 1.0

Status: FCS

Release: June 11, 2002

Copyright 2002 Sun Microsystems, Inc.

4150 Network Circle, Santa Clara, California

95054, U.S.A

All rights reserved.

NOTICE; LIMITED LICENSE GRANTS

Sun hereby grants you a fully-paid, non-exclusive, non-transferable, worldwide, limited license (without the right to sublicense), under the Specification Lead's applicable intellectual property rights to view, download, use and reproduce the Specification only for the purpose of internal evaluation, which shall be understood to include developing applications intended to run on an implementation of the Specification provided that such applications do not themselves implement any portion(s) of the Specification.

Sun also grants you a perpetual, non-exclusive, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) under any applicable copyrights or patent rights it may have in the Specification to create and/or distribute an Independent Implementation of the Specification that: (i) fully implements the Spec(s) including all its required interfaces and functionality; (ii) does not modify, subset, superset or otherwise extend the Licensor Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the Licensor Name Space other than those required/authorized by the Specification or Specifications being implemented; and (iii) passes

## Text of Third-Party Software Licenses

the TCK (including satisfying the requirements of the applicable TCK Users Guide) for such Specification. The foregoing license is expressly conditioned on your not acting outside its scope. No license is granted hereunder for any other purpose.

You need not include limitations (i)-(iii) from the previous paragraph or any other particular pass through requirements in any license You grant concerning the use of your Independent Implementation or products derived from it. However, except with respect to implementations of the Specification (and products derived from them) by the your licensee that satisfy limitations (i)-(iii) from the previous paragraph, You may neither: (a) grant or otherwise pass through to your licensees any licenses under Sun's applicable intellectual property rights; nor (b) authorize your licensees to make any claims concerning their implementation's compliance with the Spec in question.

For the purposes of this Agreement: "Independent Implementation" shall mean an implementation of the Specification that neither derives from any of Sun's source code or binary code materials nor, except with an appropriate and separate license from Sun, includes any of Sun's source code or binary code materials; and "Licensor Name Space" shall mean the public class or interface declarations whose names begin with "java", javax, "com.sun" or their equivalents in any subsequent naming convention adopted by Sun through the Java Community Process, or any recognized successors or replacements thereof.

This Agreement will terminate immediately without notice from Sun if you fail to comply with any material provision of or act outside the scope of the licenses granted above.

### TRADEMARKS

No right, title, or interest in or to any trademarks, service marks, or trade names of Sun or Sun's licensors is granted hereunder. Sun, Sun Microsystems, the Sun logo, Java, the Java Coffee Cup logo, J2EE, and JavaServer Pages are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

### DISCLAIMER OF WARRANTIES

THE SPECIFICATION IS PROVIDED "AS IS". SUN MAKES NO REPRESENTATIONS OR WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A

PARTICULAR PURPOSE, OR NON-INFRINGEMENT, THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE OR THAT ANY PRACTICE OR IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADE SECRETS OR OTHER RIGHTS. This document does not represent any commitment to release or implement any portion of the Specification in any product.

THE SPECIFICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION THEREIN; THESE CHANGES WILL BE INCORPORATED INTO NEW VERSIONS OF THE SPECIFICATION, IF ANY. SUN MAY MAKE IMPROVEMENTS AND/OR CHANGES TO THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THE SPECIFICATION AT ANY TIME. Any use of such changes in the Specification will be governed by the then-current license for the applicable version of the Specification.

#### LIMITATION OF LIABILITY

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUE, PROFITS OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO ANY FURNISHING, PRACTICING, MODIFYING OR ANY USE OF THE SPECIFICATION, EVEN IF SUN AND/OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You will indemnify, hold harmless, and defend Sun and its licensors from any claims arising or resulting from: (i) your use of the Specification; (ii) the use or distribution of your Java application, applet and/or clean room implementation; and/or (iii) any claims that later versions or releases of any Specification furnished to you are incompatible with the Specification provided to you under this license.

#### RESTRICTED RIGHTS LEGEND

U.S. Government: If this Specification is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in the Software and accompanying documentation shall be only as set forth in this license; this is in accordance with 48 C.F.R. 227.7201 through 227.7202-4 (for Department of Defense (DoD) acquisitions) and with 48 C.F.R. 2.101 and 12.212 (for non-DoD acquisitions).

#### REPORT

## Text of Third-Party Software Licenses

You may wish to report any ambiguities, inconsistencies or inaccuracies you may find in connection with your use of the Specification ("Feedback"). To the extent that you provide Sun with any Feedback, you hereby: (i) agree that such Feedback is provided on a non-proprietary and non-confidential basis, and (ii) grant Sun a perpetual, non-exclusive, worldwide, fully paid-up, irrevocable license, with the right to sublicense through multiple levels of sublicensees, to incorporate, disclose, and use without limitation the Feedback for any purpose related to the Specification and future versions, implementations, and test suites thereof.

(LFI#113300/Form ID#011801)

JSP Unified Expression Lang  
Common Dev and Distrib 1.0

### JTidy

Java HTML Tidy - JTidy  
HTML parser and pretty printer

Copyright (c) 1998-2000 World Wide Web Consortium (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

Contributing Author(s) :

Dave Raggett <dsr@w3.org>  
Andy Quick <ac.quick@sympatico.ca> (translation to Java)  
Gary L Peskin <garyp@firsttech.com> (Java development)  
Sami Lempinen <sami@lempinen.net> (release management)

The contributing author(s) would like to thank all those who helped with testing, bug fixes, and patience. This wouldn't have been possible without all of you.

### COPYRIGHT NOTICE:

This software and documentation is provided "as is," and the copyright holders and contributing author(s) make no representations or warranties, express or implied, including but not limited to, warranties of merchantability or fitness for any particular purpose or that the use of the software or documentation will not infringe any third party patents, copyrights, trademarks or other rights.

The copyright holders and contributing author(s) will not be liable for any direct, indirect, special or consequential damages arising out of any use of the software or documentation, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, documentation and executables, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The copyright holders and contributing author(s) specifically permit, without fee, and encourage the use of this source code as a component for supporting the Hypertext Markup Language in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

JUnit

BSD License

Copyright (c) 2000-2006, www.hamcrest.org  
All rights reserved.

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

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

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

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

Lib Art

Lesser GNU Public License v2

Lib Edit

Copyright (c) 1992, 1993  
The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Christos Zoulas of Cornell University.

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

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

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

#### Lib PNG

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

#### COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

libpng versions 1.2.6, August 15, 2004, through 1.2.35, February 14, 2009, are Copyright (c) 2004, 2006-2008 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.2.5 with the following individual added to the list of Contributing Authors

Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5 - October 3, 2002, are Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors

Simon-Pierre Cadieux  
Eric S. Raymond  
Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes

or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane  
Glenn Randers-Pehrson  
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997 Andreas Dilger Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler  
Kevin Bracey  
Sam Bushell  
Magnus Holmgren  
Greg Roelofs  
Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger  
Dave Martindale  
Guy Eric Schalnat  
Paul Schmidt  
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to

## Text of Third-Party Software Licenses

supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

A "png\_get\_copyright" function is available, for convenient use in "about" boxes and the like:

```
printf("%s",png_get_copyright(NULL));
```

Also, the PNG logo (in PNG format, of course) is supplied in the files "pngbar.png" and "pngbar.jpg" (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson  
glennrp at users.sourceforge.net  
February 14, 2009

Lib uuid

```
/*
 * Public include file for the UUID library
 *
 * Copyright (C) 1996, 1997, 1998 Theodore Ts'o.
 *
 * %Begin-Header%
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, and the entire permission notice in its entirety,
 *    including the disclaimer of warranties.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote
 *    products derived from this software without specific prior
 *    written permission.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF
 * WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
 * OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
 * BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
 * USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 * %End-Header%
 */
```

Lib XML2

Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar licence but with different Copyright notices) all the files are:

Copyright (C) 1998-2003 Daniel Veillard. All Rights Reserved.

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

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

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

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

LM Bench  
GNU Public License v2

%M% %I% %E%

The set of programs and documentation known as "lmbench" are distributed under the Free Software Foundation's General Public License with the following additional restrictions (which override any conflicting restrictions in the GPL):

1. You may not distribute results in any public forum, in any publication, or in any other way if you have modified the benchmarks.
2. You may not distribute the results for a fee of any kind. This includes web sites which generate revenue from advertising.

If you have modifications or enhancements that you wish included in future versions, please mail those to me, Larry McVoy, at lm@bitmover.com.

=====

Rationale for the publication restrictions:

In summary:

- a) LMbench is designed to measure enough of an OS that if you do well in all categories, you've covered latency and bandwidth in networking, disks, file systems, VM systems, and memory systems.
- b) Multiple times in the past people have wanted to report partial results.

Without exception, they were doing so to show a skewed view of whatever it was they were measuring (for example, one OS fit small processes into segments and used the segment register to switch them, getting good results, but did not want to report large process context switches because those didn't look as good).

- c) We insist that if you formally report LMbench results, you have to report all of them and make the raw results file easily available. Reporting all of them means in that same publication, a pointer does not count. Formally, in this context, means in a paper, on a web site, etc., but does not mean the exchange of results between OS developers who are tuning a particular subsystem.

We have a lot of history with benchmarking and feel strongly that there is little to be gained and a lot to be lost if we allowed the results to be published in isolation, without the complete story being told.

There has been a lot of discussion about this, with people not liking this restriction, more or less on the freedom principle as far as I can tell. We're not swayed by that, our position is that we are doing the right thing for the OS community and will stick to our guns on this one.

It would be a different matter if there were 3 other competing benchmarking systems out there that did what LMbench does and didn't have the same reporting rules. There aren't and as long as that is the case, I see no reason to change my mind and lots of reasons not to do so. I'm sorry if I'm a pain in the ass on this topic, but I'm doing the right thing for you and the sooner people realize that the sooner we can get on to real work.

Operating system design is a largely an art of balancing tradeoffs. In many cases improving one part of the system has negative effects on other parts of the system. The art is choosing which parts to optimize and which to not optimize. Just like in computer architecture, you can optimize the common instructions (RISC) or the uncommon instructions (CISC), but in either case there is usually a cost to pay (in RISC uncommon instructions are more expensive than common instructions, and in CISC common instructions are more expensive than required). The art lies in knowing which operations are important and optimizing those while minimizing the impact on the rest of the system.

Since lmbench gives a good overview of many important system features, users may see the performance of the system as a whole, and can see where tradeoffs may have been made. This is the driving force behind the publication restriction: any idiot can optimize certain subsystems while completely destroying overall system performance. If said idiot publishes \*only\* the numbers relating to the optimized subsystem, then the costs of the optimization are hidden and readers will mistakenly believe that the optimization is a good idea. By including the publication restriction readers would be able to detect that the optimization improved the subsystem performance while damaging the rest of the system performance and would be able to make an informed decision as to the merits of the optimization.

Note that these restrictions only apply to \*publications\*. We intend and encourage lmbench's use during design, development, and tweaking of systems and applications. If you are tuning the linux or BSD TCP stack, then by all means, use the networking benchmarks to evaluate the performance effects of various

modifications; Swap results with other developers; use the networking numbers in isolation. The restrictions only kick in when you go to \*publish\* the results. If you sped up the TCP stack by a factor of 2 and want to publish a paper with the various tweaks or algorithms used to accomplish this goal, then you can publish the networking numbers to show the improvement. However, the paper \*must\* also include the rest of the standard lmbench numbers to show how your tweaks may (or may not) have impacted the rest of the system. The full set of numbers may be included in an appendix, but they \*must\* be included in the paper.

This helps protect the community from adopting flawed technologies based on incomplete data. It also helps protect the community from misleading marketing which tries to sell systems based on partial (skewed) lmbench performance results.

We have seen many cases in the past where partial or misleading benchmark results have caused great harm to the community, and we want to ensure that our benchmark is not used to perpetrate further harm and support false or misleading claims.

Log4J  
Apache License v2

Merriampark Levenshtein Dist

Levenshtein Distance from <http://www.merriampark.com/ld.htm>

These three implementations are hereby placed in the public domain and are free for anyone to use.

Microsoft Reporting Services

#### MICROSOFT SOFTWARE LICENSE TERMS

#### MICROSOFT WINDOWS SOFTWARE DEVELOPMENT KIT

for Windows Server 2008 and .NET Framework 3.5

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- ?            updates,
- ?            supplements,
- ?            Internet-based services, and
- ?            support services

for this software, unless other terms accompany those items. If so, those terms apply. BY USING THE SOFTWARE, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT USE THE SOFTWARE.

If you comply with these license terms, you have the rights below.

#### 1. INSTALLATION AND USE RIGHTS.

a. Installation and Use. One user may install and use any number of copies of the software on your devices to design, develop and test your programs that run on a Microsoft Windows operating system.

b. Included Microsoft Programs. The software contains other Microsoft programs. These license terms apply to your use of those programs.

2. ADDITIONAL LICENSING REQUIREMENTS AND/OR USE RIGHTS.

a. Distributable Code. The software contains code that you are permitted to distribute in programs you develop if you comply with the terms below.

i. Right to Use and Distribute. The code and text files listed below are "Distributable Code."

? REDIST.TXT Files. You may copy and distribute the object code form of code listed in REDIST.TXT files.

? Sample Code. You may modify, copy, and distribute the source and object code form of code marked as "sample."

? Microsoft Merge Modules. You may copy and distribute the unmodified output of Microsoft Merge Modules.

? Third Party Distribution. You may permit distributors of your programs to copy and distribute the Distributable Code as part of those programs.

ii. Distribution Requirements. For any Distributable Code you distribute, you must

? add significant primary functionality to it in your programs;

? for any Distributable Code having a filename extension of .lib, distribute only the results of running such Distributable Code through a linker with your application;

? distribute Distributable Code included in a setup program only as part of that setup program without modification;

? require distributors and external end users to agree to terms that protect it at least as much as this agreement;

? display your valid copyright notice on your programs;

? for Distributable Code from the Windows Media Services SDK portions of the software, include in your program's Help-About box (or in another obvious place if there is no box) the following copyright notice: "Portions utilize Microsoft Windows Media Technologies. Copyright (c) 2006 Microsoft Corporation. All Rights Reserved"; and

? indemnify, defend, and hold harmless Microsoft from any claims, including attorneys' fees, related to the distribution or use of your programs.

iii. Distribution Restrictions. You may not

? alter any copyright, trademark or patent notice in the Distributable Code;

? use Microsoft's trademarks in your programs' names or in a way that suggests your programs come from or are endorsed by Microsoft;

? include Distributable Code in malicious, deceptive or unlawful programs; or

? modify or distribute the source code of any Distributable Code so that any part of it becomes subject to an Excluded License. An Excluded License is one that requires, as a condition of use, modification or distribution, that

? the code be disclosed or distributed in source code form; or

? others have the right to modify it.

b. Additional Functionality. Microsoft may provide additional functionality for the software. Other license terms and fees may apply.

3. INTERNET-BASED SERVICES. Microsoft provides Internet-based services with the software. It may change or cancel them at any time. You may not use this service in any way that could harm it or impair anyone else's use of it. You may not use the service to try to gain unauthorized access to any service, data, account or network by any means.

4. MICROSOFT .NET BENCHMARK TESTING. The software includes one or more components of the .NET Framework 3.5 ("".NET Components"). You may conduct internal benchmark testing of those components. You may disclose the results of any benchmark test of those components, provided that you comply with the conditions set forth at <http://go.microsoft.com/fwlink/?LinkID=66406>. Notwithstanding any other agreement you may have with Microsoft, if you disclose such benchmark test results, Microsoft shall have the right to disclose the results of benchmark tests it conducts of your products that compete with the applicable .NET Component, provided it complies with the same conditions set forth at <http://go.microsoft.com/fwlink/?LinkID=66406>.

5. Scope of License. The software is licensed, not sold. This agreement only gives you some rights to use the software. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that only allow you to use it in certain ways. For more information, see [www.microsoft.com/licensing/userrights](http://www.microsoft.com/licensing/userrights) <<http://www.microsoft.com/>

licensing/userights>. You may not

- ? work around any technical limitations in the software;
- ? reverse engineer, decompile or disassemble the software, except and only to the extent that applicable law expressly permits, despite this limitation;
- ? make more copies of the software than specified in this agreement or allowed by applicable law, despite this limitation;
- ? publish the software for others to copy;
- ? rent, lease or lend the software; or
- ? use the software for commercial software hosting services.

6. CODE GENERATION AND OPTIMIZATION TOOLS. You may not use the code generation or optimization tools in the software (such as compilers, linkers, assemblers, runtime code generators, and code generating design and modeling tools) to create programs, object code, libraries, assemblies, or executables to run on a platform other than Microsoft operating systems, run-time technologies, or application platforms.

7. BACKUP COPY. You may make one backup copy of the software. You may use it only to reinstall the software.

8. DOCUMENTATION. Any person that has valid access to your computer or internal network may copy and use the documentation for your internal, reference purposes.

9. TRANSFER TO A THIRD PARTY. The first user of the software may transfer it, and this agreement, directly to a third party. Before the transfer, that party must agree that this agreement applies to the transfer and use of the software. The first user must uninstall the software before transferring it separately from the device. The first user may not retain any copies.

10. Export Restrictions. The software is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the software. These laws include restrictions on destinations, end users and end use. For additional information, see [www.microsoft.com/exporting](http://www.microsoft.com/exporting) <<http://www.microsoft.com/exporting>>.

11. SUPPORT SERVICES. Because this software is "as is," we may not provide support services for it.

12. Entire Agreement. This agreement, and the terms for supplements, updates, Internet-based services and support services that you use, are the entire agreement for the software and support services.

13. Applicable Law.

- a. United States. If you acquired the software in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.
- b. Outside the United States. If you acquired the software in any other country, the laws of that country apply.

14. Legal Effect. This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the software. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.

15. Disclaimer of Warranty. The software is licensed "as-is." You bear the risk of using it. Microsoft gives no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this agreement cannot change. To the extent permitted under your local laws, Microsoft excludes the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

16. Limitation on and Exclusion of Remedies and Damages. You can recover from Microsoft and its suppliers only direct damages up to U.S. \$5.00. You cannot recover any other damages, including consequential, lost profits, special, indirect or incidental damages. This limitation applies to

- ? anything related to the software, services, content (including code) on third party Internet sites, or third party programs; and
- ? claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the

## Text of Third-Party Software Licenses

damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this software is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce logiciel ?tant distribu? au Qu?bec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en fran?ais.

**EXON?RATION DE GARANTIE.** Le logiciel vis? par une licence est offert ? tel quel ?. Toute utilisation de ce logiciel est ? votre seule risque et p?ril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez b?n?fici?er de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualit? marchande, d'ad?quation ? un usage particulier et d'absence de contrefa?on sont exclues.

**LIMITATION DES DOMMAGES-INT?R?TS ET EXCLUSION DE RESPONSABILIT? POUR LES DOMMAGES.** Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement ? hauteur de 5,00 \$ US. Vous ne pouvez pr?tendre ? aucune indemnisation pour les autres dommages, y compris les dommages sp?ciaux, indirects ou accessoires et pertes de b?n?fices.

Cette limitation concerne :

? tout ce qui est reli? au logiciel, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers ; et

? les r?clamation?ns au titre de violation de contrat ou de garantie, ou au titre de responsabilit? stricte, de n?gligence ou d'une autre faute dans la limite autoris?e par la loi en vigueur.

Elle s'applique ?galement, m?me si Microsoft connaissait ou devrait conna?tre l'?ventualit? d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilit? pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas ? votre ?gard.

**EFFET JURIDIQUE.** Le pr?sent contrat d?crit certains droits juridiques. Vous pourriez avoir d'autres droits pr?vus par les lois de votre pays. Le pr?sent contrat ne modifie pas les droits que vous conf?rent les lois de votre pays si celles ci ne le permettent pas.

## MinGW

The MinGW base runtime package has been placed in the public domain, and is not governed by copyright. This basically means that you can do what you like with the code.

## MochiKit

MochiKit is dual-licensed software. It is available under the terms of the MIT License, or the Academic Free License version 2.1. The full text of each license is included below.

### MIT License

=====

Copyright (c) 2005 Bob Ippolito. All rights reserved.

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

The above copyright notice and this permission notice shall be included in all copies or

substantial portions of the Software.

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

Academic Free License v. 2.1

=====

Copyright (c) 2005 Bob Ippolito. All rights reserved.

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

- a) to reproduce the Original Work in copies;
- b) to prepare derivative works ("Derivative Works") based upon the Original Work;
- c) to distribute copies of the Original Work and Derivative Works to the public;
- d) to perform the Original Work publicly; and
- e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that

Licensor otherwise would have a right to license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately proceeding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. § 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive

the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licenser promises not to interfere with or be responsible for such uses by You.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

Mod Auth PQSQL

```

/*
 * Based on mod_auth_pgsql - http://www.giuseppetanzilli.it/mod_auth_pgsql2/
 * with some modifications.
 */

/* =====
 * Copyright (c) 1996 Vidya Media Ventures, Inc. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of this source code or a derived source code must
 *    retain the above copyright notice, this list of conditions and the
 *    following disclaimer.
 *
 * 2. Redistributions of this module or a derived module in binary form
 *    must reproduce the above copyright notice, this list of conditions
 *    and the following disclaimer in the documentation and/or other
 *    materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY VIDYA MEDIA VENTURES, INC. ``AS IS'' AND
 * ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

```

## Text of Third-Party Software Licenses

```
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL VIDYA MEDIA VENTURES, INC.  
* OR ITS EMPLOYEES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
* OF THE POSSIBILITY OF SUCH DAMAGE.  
* ======  
*  
* This software is a contribution to and makes use of the Apache HTTP  
* server which is written, maintained and copywritten by the Apache Group.  
* See http://www.apache.org/ for more information.  
*  
* This software makes use of libpq which an interface to the PostgreSQL  
* database. PostgreSQL is copyright (c) 1994 by the Regents of the  
* University of California. As of this writing, more information on  
* PostgreSQL can be found at http://www.postgreSQL.org/  
*  
*/
```

Mono

MIT License

GNU Public License v2

Lesser GNU Public License v2

Mono is made up of many pieces of code, all of them open source, but different pieces of Mono use different licensing terms.

For comments, corrections and updates, please contact [mono@novell.com](mailto:mono@novell.com)

### \* Dual Licensing

Parts of Mono are dual licensed, they are available to the public in GPL or LGPL forms, but we also offer those pieces under commercial terms from Novell for when the GPL and the LGPL are not suitable.

We have tried to pick the licenses that will maximize adoption of Mono, so we tend to use the MIT X11 or LGPL licenses.

Contributions to dual-licensed module require that the author contributes the code under the terms of the MIT X11 code, or to sign an agreement that allows Novell to redistribute the code under other licenses.

Contributions for other modules should be under the same license terms as the rest of the module, or under MIT X11 terms.

If you need further information, please contact [mono@novell.com](mailto:mono@novell.com)

### \* The Modules

#### \*\* mono/mono: the Mono VM

This code is dual licensed under the LGPL or commercial licenses.

The LGPL ensures that Mono can be used in most scenarios, but gives Novell the flexibility to relicense the code for embedded systems, static linking or commercial settings where the LGPL can not be used.

We consider non-LGPL use instances where you use this on an embedded system where the end user is not able to upgrade the Mono VM or Moonlight installation or distribution that is part of your product (Section 6 and 7), you would have to obtain a commercial license from Novell (consider software burned into a ROM, systems where end users would not be able to upgrade, an embedded console, a game console that imposes limitations on the distribution and access to the code, a phone platform that prevents end users from upgrading Moonlight).

Contact mono@novell.com for details on obtaining the Mono runtime under other terms.

**\*\* mono/eglib:** Mono's X11 glib implementation

This is a minimal subset of glib that is to be licensed under the terms of the MIT X11, this means that this code can be used for any purposes by anyone.

**\*\* mcs/mcs, mcs/gmcs**

The C# Compilers (1.0 and 2.0)

These compilers are dual licensed under the GPL and MIT X11 license terms.

**\*\* tests**

Unless explicitly stated, the tests are under the MIT X11 license.

**\*\* mcs/class**

The class libraries developed by the Mono team are licensed under the MIT X11 terms.

In addition to the class libraries developed by the Mono team, there are a number of class libraries that we bundle as part of the distribution that were integrated from third-parties or that contain code that was originally licensed under different terms, these are:

ByteFX.Data: LGPL

Npgsql: LGPL

FirebirdSql.Data.Firebird: Firebird public license.  
See: mcs/class/FirebirdSql.Data.Firebird/license.txt

ICSharpCode.SharpZipLib, GPL with exceptions.  
See: mcs/class/ICSharpCode.SharpZipLib/README

**\*\* mcs/tools**

## Text of Third-Party Software Licenses

These are licensed under the MIT X11 license, except where the GPL is explicitly used.

\*\* mcs/jay

This is a port of Berkeley yacc, so it is available under the BSD license. See the license in the individual C files for details.

\*\* mono/man

Manual pages and Mono documentation are covered by the MIT X11 license.

\* samples

The code in the "samples" directory is released under the MIT X11 license.

\* The Licenses

These are the licenses used in Mono, the files are located:

GNU GPL: details available in the file mcs/LICENSE.GPL  
GNU LGPL: details available in the file mcs/LICENSE.LGPL  
MIT X11: text available in the file mcs/MIT.X11  
MPL: text available in the file mcs/LICENSE.MPL

## Mono Security Protocol

2. Transport Security Layer (TLS) sources license

-- -----

## The MIT License

Copyright (c) 2003 Carlos Guzm?n ?lvarez

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

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

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

## MurmurHash2

/\*\* Murmur hash 2.0.

```

*
* The murmur hash is a relative fast hash function from
* http://murmurhash.googlecode.com/ for platforms with efficient
* multiplication.
*
* This is a re-implementation of the original C code plus some
* additional features.
*
* Public domain.
*
* @author Vilim Holub
* @version 1.0.2

```

## NCurses

```

-----
-- Copyright (c) 1998-2004,2006 Free Software Foundation, Inc. --
-- --
-- Permission is hereby granted, free of charge, to any person obtaining a --
-- copy of this software and associated documentation files (the --
-- "Software"), to deal in the Software without restriction, including --
-- without limitation the rights to use, copy, modify, merge, publish, --
-- distribute, distribute with modifications, sublicense, and/or sell copies --
-- of the Software, and to permit persons to whom the Software is furnished --
-- to do so, subject to the following conditions: --
-- --
-- The above copyright notice and this permission notice shall be included --
-- in all copies or substantial portions of the Software. --
-- --
-- THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS --
-- OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF --
-- MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN --
-- NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, --
-- DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR --
-- OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE --
-- USE OR OTHER DEALINGS IN THE SOFTWARE. --
-- --
-- Except as contained in this notice, the name(s) of the above copyright --
-- holders shall not be used in advertising or otherwise to promote the --
-- sale, use or other dealings in this Software without prior written --
-- authorization. --
-- 
-- $Id: README,v 1.23 2006/04/22 22:19:37 tom Exp $
-----
```

README file for the ncurses package

See the file ANNOUNCE for a summary of ncurses features and ports.  
 See the file INSTALL for instructions on how to build and install ncurses.  
 See the file NEWS for a release history and bug-fix notes.  
 See the file TO-DO for things that still need doing, including known bugs.

Browse the file misc/ncurses-intro.html for narrative descriptions of how to use ncurses and the panel, menu, and form libraries.

Browse the file doc/html/hackguide.html for a tour of the package internals.

ROADMAP AND PACKAGE OVERVIEW:

You should be reading this file in a directory called: ncurses-d.d, where d.d is the current version number (see the dist.mk file in this directory for that). There should be a number of subdirectories, including `c++', `form', `man', `menu', `misc', `ncurses', `panel', `progs', `test', `tack' and `Ada95'. (The 'tack' program may be distributed separately).

A full build/install of this package typically installs several libraries, a handful of utilities, and a database hierarchy. Here is an inventory of the pieces:

The libraries are:

libncurses.a	(normal)
libncurses.so	(shared)
libncurses_g.a	(debug and trace code enabled)
libncurses_p.a	(profiling enabled)
libpanel.a	(normal)
libpanel.so	(shared)
libpanel_g.a	(debug and trace code enabled)
libmenu.a	(normal)
libmenu.so	(shared)
libmenu_g.a	(debug enabled)
libform.a	(normal)
libform.so	(shared)
libform_g.a	(debug enabled)

If you configure using the --enable-widec option, a "w" is appended to the library names (e.g., libncursesw.a), and the resulting libraries support wide-characters, e.g., via a UTF-8 locale. The corresponding header files are compatible with the non-wide-character configuration; wide-character features are provided by ifdef's in the header files. The wide-character library interfaces are not binary-compatible with the non-wide-character version.

The ncurses libraries implement the curses API. The panel, menu and forms libraries implement clones of the SVr4 panel, menu and forms APIs. The source code for these lives in the `ncurses', `panel', `menu', and `form' directories respectively.

In the `c++' directory, you'll find code that defines an interface to the curses, forms, menus and panels library packaged as C++ classes, and a demo program in C++ to test it. These class definition modules are not installed by the 'make install.libs' rule as libncurses++.

In the `Ada95' directory, you'll find code and documentation for an Ada95 binding of the curses API, to be used with the GNAT compiler. This binding is built by a normal top-level `make' if configure detects an usable version of GNAT (3.11 or above). It is not installed automatically. See the Ada95 directory for more build and installation instructions and for documentation of the binding.

To do its job, the ncurses code needs your terminal type to be set in the environment variable TERM (normally set by your OS; under UNIX, getty(1) typically does this, but you can override it in your .profile); and, it needs a database of terminal descriptions in which to look up your terminal type's capabilities.

In older (V7/BSD) versions of curses, the database was a flat text file, /etc/termcap; in newer (USG/USL) versions, the database is a hierarchy of fast-loading binary description blocks under /usr/lib/terminfo. These binary blocks are compiled from an improved editable text representation called `terminfo' format (documented in man/terminfo.5). The ncurses library can use either /etc/termcap or the compiled binary terminfo blocks, but prefers the second form.

In the `misc' directory, there is a text file terminfo.src, in editable terminfo format, which can be used to generate the terminfo binaries (that's what make install.data does). If the package was built with the --enable-termcap option enabled, and the ncurses library cannot find a terminfo description for your terminal, it will fall back to the termcap file supplied with your system (which the ncurses package installation leaves strictly alone).

The utilities are as follows:

tic	-- terminfo source to binary compiler
infocmp	-- terminfo binary to source decompiler/comparator
clear	-- emits clear-screen for current terminal
tput	-- shell-script access to terminal capabilities.
toe--	table of entries utility
tset	-- terminal-initialization utility

The first two (tic and infocmp) are used for manipulating terminfo descriptions; the next two (clear and tput) are for use in shell scripts. The last (tset) is provided for 4.4BSD compatibility. The source code for all of these lives in the `progs' directory.

Detailed documentation for all libraries and utilities can be found in the `man' and `doc' directories. An HTML introduction to ncurses, panels, and menus programming lives in the `doc/html' directory. Manpages in HTML format are under `doc/html/man'.

The `test' directory contains programs that can be used to verify or demonstrate the functions of the ncurses libraries. See test/README for descriptions of these programs. Notably, the `ncurses' utility is designed to help you systematically exercise the library functions.

#### AUTHORS:

Pavel Curtis:  
wrote the original ncurses

Zeyd M. Ben-Halim:  
port of original to Linux and many enhancements.

Thomas Dickey (maintainer for 1.9.9g through 4.1, resuming with FSF's 5.0): configuration scripts, porting, mods to adhere to XSI Curses in the areas of background color, terminal modes. Also memory leak testing, the wresize, default colors and key definition extensions and numerous bug fixes (more than half of those enumerated in NEWS beginning with the internal release 1.8.9).

Florian La Roche (official maintainer for FSF's ncurses 4.2)  
Beginning with release 4.2, ncurses is distributed under an MIT-style license.

## Text of Third-Party Software Licenses

Eric S. Raymond:

the man pages, infocmp(1), tput(1), clear(1), captoinfo(1), tset(1), toe(1), most of tic(1), trace levels, the HTML intro, wgetnstr() and many other entry points, the cursor-movement optimization, the scroll-pack optimizer for vertical motions, the mouse interface and xterm mouse support, and the ncurses test program.

Juergen Pfeifer

The menu and form libraries, C++ bindings for ncurses, menus, forms and panels, as well as the Ada95 binding. Ongoing support for panel.

CONTRIBUTORS:

Alexander V. Lukyanov

for numerous fixes and improvements to the optimization logic.

David MacKenzie

for first-class bug-chasing and methodical testing.

Ross Ridge

for the code that hacks termcap parameterized strings into terminfo.

Warren Tucker and Gerhard Fuernkranz,

for writing and sending the panel library.

Hellmuth Michaelis,

for many patches and testing the optimization code.

Eric Newton, Ulrich Drepper, and Anatoly Ivasyuk:

the C++ code.

Jonathan Ross,

for lessons in using sed.

Keith Bostic (maintainer of 4.4BSD curses)

for help, criticism, comments, bug-finding, and being willing to deep-six BSD curses for this one when it grew up.

Richard Stallman,

for his commitment to making ncurses free software.

Countless other people have contributed by reporting bugs, sending fixes, suggesting improvements, and generally whining about ncurses :-)

BUGS:

See the INSTALL file for bug and developer-list addresses.

The Hacker's Guide in the doc directory includes some guidelines on how to report bugs in ways that will get them fixed most quickly.

NetCat

Netcat is entirely my own creation, although plenty of other code was used as examples. It is freely given away to the Internet community in the hope that it will be useful, with no restrictions except giving credit where it is due. No GPLs, Berkeley copyrights or any of that nonsense. The author assumes NO responsibility for how anyone uses it. If netcat makes you rich somehow and you're feeling generous, mail me a check. If you are affiliated in any way

with Microsoft Network, get a life. Always ski in control. Comments, questions, and patches to nc110-devel@lists.sourceforge.net.

---

Netcat 1.10 is an updated release of Netcat, a simple Unix utility which reads and writes data across network connections using TCP or UDP protocol. It is designed to be a reliable "back-end" tool that can be used directly or easily driven by other programs and scripts. At the same time it is a feature-rich network debugging and exploration tool, since it can create almost any kind of connection you would need and has several interesting built-in capabilities.

Some of netcat's major features are:

- Outbound or inbound connections, TCP or UDP, to or from any ports
- Full DNS forward/reverse checking, with appropriate warnings
- Ability to use any local source port
- Ability to use any locally-configured network source address
- Built-in port-scanning capabilities, with randomizer
- Built-in loose source-routing capability
- Can read command line arguments from standard input
- Slow-send mode, one line every N seconds
- Hex dump of transmitted and received data
- Optional ability to let another program service established connections
- Optional telnet-options responder

A very short list of potential uses:

- Script backends
- Scanning ports and inventorying services, automated probes
- Backup handlers
- File transfers
- Server testing, simulation, debugging, and hijacking
- Firewall testing
- Proxy gatewaying
- Network performance testing
- Address spoofing tests
- Protecting X servers
- 1001 other uses you'll likely come up with

Changes between the 1.00 release and this release:

- Better portability -- updated generic.h and Makefile [thanx folks!]
- Indication of local-end interface address on inbound connections
- That's \*Dave\* Borman's telnet, not Paul Borman...
- Better indication of DNS errors
- Total byte counts printed if -v -v is used
- A bunch of front-end driver companion programs and scripts
- Better handling of stdin arguments-plus-data
- Hex-dump feature
- Telnet responder
- Program exec works inbound or outbound now

Netcat and the associated package is a product of Avian Research, and is freely available in full source form with no restrictions save an obligation to give credit where due. Get it via anonymous FTP at [avian.org:/src/hacks/nc110.tgz](http://avian.org:/src/hacks/nc110.tgz) which is a gzipped tar file and not to be confused with its version 1.00 precursor, nc100.tgz. Other distribution formats can be accommodated upon request. Netcat is also mirrored at the following [faster] sites:

## Text of Third-Party Software Licenses

zippy.telcom.arizona.edu:/pub/mirrors/avian.org/hacks/nc110.tgz  
ftp.sterling.com:/mirrors/avian.org/src/hacks/nc110.tgz  
coast.cs.purdue.edu:/pub/tools/unix/netcat/nc110.tgz  
ftp.rge.com:/pub/security/coast/mirrors/avian.org/netcat/nc110.tgz

\_H\* 960320

### Netlogd

Copyright (c) 1998-2010, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from the U.S. Dept. of Energy). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: (1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. (3) Neither the name of the University of California, Lawrence Berkeley National Laboratory, U.S. Dept. of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

You are under no obligation whatsoever to provide any bug fixes, patches, or upgrades to the features, functionality or performance of the source code ("Enhancements") to anyone; however, if you choose to make your Enhancements available either publicly, or directly to Lawrence Berkeley National Laboratory, without imposing a separate written license agreement for such Enhancements, then you hereby grant the following license: a non-exclusive, royalty-free perpetual license to install, use, modify, prepare derivative works, incorporate into other computer software, distribute, and sublicense such enhancements or derivative works thereof, in binary and source code form.

### Net SNMP

Various copyrights apply to this package, listed in various separate parts below. Please make sure that you read all the parts.

----- Part 1: CMU/UCD copyright notice: (BSD like) -----

Copyright 1989, 1991, 1992 by Carnegie Mellon University

Derivative Work - 1996, 1998-2000 Copyright 1996, 1998-2000 The Regents

of the University of California

All Rights Reserved

Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL CMU OR THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM THE LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----- Part 2: Networks Associates Technology, Inc copyright notice (BSD) -----

Copyright (c) 2001-2003, Networks Associates Technology, Inc All rights reserved.

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

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

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

----- Part 3: Cambridge Broadband Ltd. copyright notice (BSD) -----

Portions of this code are copyright (c) 2001-2003, Cambridge Broadband Ltd. All rights reserved.

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

## Text of Third-Party Software Licenses

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

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

----- Part 4: Sun Microsystems, Inc. copyright notice (BSD) -----

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Use is subject to license terms below.

This distribution may include materials developed by third parties.

Sun, Sun Microsystems, the Sun logo and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

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

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

----- Part 5: Sparta, Inc copyright notice (BSD) -----

Copyright (c) 2003-2009, Sparta, Inc All rights reserved.

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

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

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

----- Part 6: Cisco/BUPTNIC copyright notice (BSD) -----

Copyright (c) 2004, Cisco, Inc and Information Network Center of Beijing University of Posts and Telecommunications. All rights reserved.

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

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

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

## Text of Third-Party Software Licenses

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

----- Part 7: Fabasoft R&D Software GmbH & Co KG copyright notice (BSD) -----

Copyright (c) Fabasoft R&D Software GmbH & Co KG, 2003 oss@fabasoft.com Author: Bernhard Penz

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

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* The name of Fabasoft R&D Software GmbH & Co KG or any of its subsidiaries, brand or product names may not be used to endorse or promote products derived from this software without specific prior written permission.

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

----- Part 8: Apple Inc. copyright notice (BSD) -----

Copyright (c) 2007 Apple Inc. All rights reserved.

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

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

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

----- Part 9: ScienceLogic, LLC copyright notice (BSD) -----

Copyright (c) 2009, ScienceLogic, LLC All rights reserved.

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

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

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

#### Netty Apache License v2

The Netty Project - Copyright 2009 Red Hat, Inc, and is licensed under the Apache License version 2.0 as published by the Apache Software Foundation.

A summary of the individual contributors is given below. Any omission should be sent to Trustin Lee <[trustin@gmail.com](mailto:trustin@gmail.com)>.

SVN Login(s)	Name
-----	-----
amit.bhayani@jboss.com	Amit Bhayani
ataylor	Andy Taylor
beve	Daniel Bevenius
fredbregier	Frederic Bregier
trustin	Trustin Heuiseung Lee
-----	-----

\* JBoss is a registered trademark of Red Hat, Inc.

The Netty Project  
=====

Please visit the Netty web site for more information:

\* <http://www.jboss.org/netty/>

Copyright 2009 Red Hat, Inc.

Red Hat licenses this product to you under the Apache License, version 2.0 (the

## Text of Third-Party Software Licenses

"License"); you may not use this product except in compliance with the License.  
You may obtain a copy of the License at:

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

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

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

-----  
This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

```
* LICENSE:  
  * license/LICENSE.jsr177y.txt (Public Domain)  
* HOMEPAGE:  
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/  
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/
```

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

```
* LICENSE:  
  * license/LICENSE.base64.txt (Public Domain)  
* HOMEPAGE:  
  * http://iharder.sourceforge.net/current/java/base64/
```

This product contains a modified version of 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

```
* LICENSE:  
  * license/LICENSE.jzlib.txt (BSD Style License)  
* HOMEPAGE:  
  * http://www.jcraft.com/jzlib/
```

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

```
* LICENSE:  
  * license/LICENSE.protobuf.txt (New BSD License)  
* HOMEPAGE:  
  * http://code.google.com/p/protobuf/
```

This product optionally depends on 'SLF4J', a simple logging facade for Java, which can be obtained at:

```
* LICENSE:  
  * license/LICENSE.slf4j.txt (MIT License)  
* HOMEPAGE:  
  * http://www.slf4j.org/
```

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

- \* LICENSE:
  - \* license/LICENSE.commons-logging.txt (Apache License 2.0)
- \* HOMEPAGE:
  - \* <http://commons.apache.org/logging/>

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

- \* LICENSE:
  - \* license/LICENSE.log4j.txt (Apache License 2.0)
- \* HOMEPAGE:
  - \* <http://logging.apache.org/log4j/>

This product optionally depends on 'JBoss Logging', a logging framework, which can be obtained at:

- \* LICENSE:
  - \* license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
- \* HOMEPAGE:
  - \* <http://anonsvn.jboss.org/repos/common/common-logging-spi/>

This product optionally depends on 'Apache Felix', an open source OSGi framework implementation, which can be obtained at:

- \* LICENSE:
  - \* license/LICENSE.felix.txt (Apache License 2.0)
- \* HOMEPAGE:
  - \* <http://felix.apache.org/>

## NPG SQL

Copyright (c) 2002-2007, The Npgsql Development Team

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

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

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

## NPG SQL TestSuite Lesser GNU Public License v2.1

```
// Author:
//Dave Page (dpage@postgresql.org)
//Copyright (C) 2002 The nClusterDNProvider Development Team
//npgsql-general@gborg.postgresql.org
```

## Text of Third-Party Software Licenses

```
//http://gborg.postgresql.org/project/npgsql/projdisplay.php
//
// This library is free software; you can redistribute it and/or
// modify it under the terms of the GNU Lesser General Public
// License as published by the Free Software Foundation; either
// version 2.1 of the License, or (at your option) any later version.
//
// This library is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
// Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public
// License along with this library; if not, write to the Free Software
// Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
```

### OCaml

Portions of this code based on Ocaml Manual with License:

The Objective Caml system is copyright ? 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Institut National de Recherche en Informatique et en Automatique (INRIA). INRIA holds all ownership rights to the Objective Caml system.

The Objective Caml system is open source and can be freely redistributed. See the file LICENSE in the distribution for licensing information.

The present documentation is copyright ? 2008 Institut National de Recherche en Informatique et en Automatique (INRIA). The Objective Caml documentation and user's manual may be reproduced and distributed in whole or in part, subject to the following conditions:

- The copyright notice above and this permission notice must be preserved complete on all complete or partial copies.
- Any translation or derivative work of the Objective Caml documentation and user's manual must be approved by the authors in writing before distribution.
- If you distribute the Objective Caml documentation and user's manual in part, instructions for obtaining the complete version of this manual must be included, and a means for obtaining a complete version provided.
- Small portions may be reproduced as illustrations for reviews or quotes in other works without this permission notice if proper citation is given.

In the following, "the Library" refers to all files marked "Copyright INRIA" in the following directories and their sub-directories:

asmrun,byterun,camlp4,config,otherlibs,stdlib,win32caml

and "the Compiler" refers to all files marked "Copyright INRIA" in the following directories and their sub-directories:

asmcomp,boot,bytecomp,debugger,driver,lex,ocamldoc,parsing,tools,toplevel,typing,utils,yacc

The Compiler is distributed under the terms of the Q Public License

version 1.0 with a change to choice of law (included below).

The Library is distributed under the terms of the GNU Library General Public License version 2 (included below).

As a special exception to the Q Public Licence, you may develop application programs, reusable components and other software items that link with the original or modified versions of the Compiler and are not made available to the general public, without any of the additional requirements listed in clause 6c of the Q Public licence.

As a special exception to the GNU Library General Public License, you may link, statically or dynamically, a "work that uses the Library" with a publicly distributed version of the Library to produce an executable file containing portions of the Library, and distribute that executable file under terms of your choice, without any of the additional requirements listed in clause 6 of the GNU Library General Public License. By "a publicly distributed version of the Library", we mean either the unmodified Library as distributed by INRIA, or a modified version of the Library that is distributed under the conditions defined in clause 2 of the GNU Library General Public License. This exception does not however invalidate any other reasons why the executable file might be covered by the GNU Library General Public License.

---

#### THE Q PUBLIC LICENSE version 1.0

Copyright (C) 1999 Troll Tech AS, Norway.  
 Everyone is permitted to copy and  
 distribute this license document.

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model.

This license applies to any software containing a notice placed by the copyright holder saying that it may be distributed under the terms of the Q Public License version 1.0. Such software is herein referred to as the Software. This license covers modification and distribution of the Software, use of third-party application programs based on the Software, and development of free software which uses the Software.

#### Granted Rights

1. You are granted the non-exclusive rights set forth in this license provided you agree to and comply with any and all conditions in this license. Whole or partial distribution of the Software, or software items that link with the Software, in any form signifies acceptance of this license.
2. You may copy and distribute the Software in unmodified form provided that the entire package, including - but not restricted to - copyright, trademark notices and disclaimers, as released by the initial developer of the Software, is distributed.
3. You may make modifications to the Software and distribute your modifications, in a form that is separate from the Software, such as patches. The following restrictions apply to modifications:

## Text of Third-Party Software Licenses

- a. Modifications must not alter or remove any copyright notices in the Software.
  - b. When modifications to the Software are released under this license, a non-exclusive royalty-free right is granted to the initial developer of the Software to distribute your modification in future versions of the Software provided such versions remain available under these terms in addition to any other license(s) of the initial developer.
4. You may distribute machine-executable forms of the Software or machine-executable forms of modified versions of the Software, provided that you meet these restrictions:
- a. You must include this license document in the distribution.
  - b. You must ensure that all recipients of the machine-executable forms are also able to receive the complete machine-readable source code to the distributed Software, including all modifications, without any charge beyond the costs of data transfer, and place prominent notices in the distribution explaining this.
  - c. You must ensure that all modifications included in the machine-executable forms are available under the terms of this license.
5. You may use the original or modified versions of the Software to compile, link and run application programs legally developed by you or by others.
6. You may develop application programs, reusable components and other software items that link with the original or modified versions of the Software. These items, when distributed, are subject to the following requirements:
- a. You must ensure that all recipients of machine-executable forms of these items are also able to receive and use the complete machine-readable source code to the items without any charge beyond the costs of data transfer.
  - b. You must explicitly license all recipients of your items to use and re-distribute original and modified versions of the items in both machine-executable and source code forms. The recipients must be able to do so without any charges whatsoever, and they must be able to re-distribute to anyone they choose.
  - c. If the items are not available to the general public, and the initial developer of the Software requests a copy of the items, then you must supply one.

### Limitations of Liability

In no event shall the initial developers or copyright holders be liable for any damages whatsoever, including - but not restricted to - lost revenue or profits or other direct, indirect, special, incidental or consequential damages, even if they have been advised of the possibility of such damages, except to the extent invariable law, if

any, provides otherwise.

#### No Warranty

The Software and this license document are provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

#### Choice of Law

This license is governed by the Laws of France.

---

#### GNU LIBRARY GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.  
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is  
numbered 2 because it goes with version 2 of the ordinary GPL.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal

permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a

medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked

with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new

## Text of Third-Party Software Licenses

versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### END OF TERMS AND CONDITIONS

## Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the

"copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This library is free software; you can redistribute it and/or  
modify it under the terms of the GNU Library General Public  
License as published by the Free Software Foundation; either  
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
Library General Public License for more details.

You should have received a copy of the GNU Library General Public  
License along with this library; if not, write to the Free  
Software Foundation, Inc., 59 Temple Place - Suite 330, Boston,  
MA 02111-1307, USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the library, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the  
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

OleDb  
Lesser GNU Public License v2.1

The Aster Database OLE DB Provider code is brought to you by:  
Teradata Aster.

Certain code was taken and adapted from the PostgreSQL project.  
This code is Copyrighted by:  
Lingnu Open Source Consulting ltd  
PostgreSQL Global Development Group  
Regents of the University of California

Original GBORG OLE DB provider project by Marek Mosiewicz  
PGArc OLE DB project by Christoph Spoerri (<http://sourceforge.net/projects/pgarc>)

OpenCSV  
Apache License v2

Open LDAP

The OpenLDAP Public License  
Version 2.8, 17 August 2003

## Text of Third-Party Software Licenses

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

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

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

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

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

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

Copyright 1998-2008 The OpenLDAP Foundation  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted only as authorized by the OpenLDAP Public License.

A copy of this license is available in the file LICENSE in the top-level directory of the distribution or, alternatively, at <<http://www.OpenLDAP.org/license.html>>.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Individual files and/or contributed packages may be copyright by

other parties and/or subject to additional restrictions.

This work is derived from the University of Michigan LDAP v3.3 distribution. Information concerning this software is available at <<http://www.umich.edu/~dirsvcs/ldap/ldap.html>>.

This work also contains materials derived from public sources.

Additional information about OpenLDAP can be obtained at <<http://www.openldap.org/>>.

---

Portions Copyright 1998-2008 Kurt D. Zeilenga.

Portions Copyright 1998-2006 Net Boolean Incorporated.

Portions Copyright 2001-2006 IBM Corporation.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted only as authorized by the OpenLDAP Public License.

---

Portions Copyright 1999-2008 Howard Y.H. Chu.

Portions Copyright 1999-2008 Symas Corporation.

Portions Copyright 1998-2003 Hallvard B. Furuseth.

Portions Copyright 2008 Gavin Henry.

Portions Copyright 2008 Suretec Systems.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that this notice is preserved. The names of the copyright holders may not be used to endorse or promote products derived from this software without their specific prior written permission. This software is provided ``as is'' without express or implied warranty.

---

Portions Copyright (c) 1992-1996 Regents of the University of Michigan.  
All rights reserved.

Redistribution and use in source and binary forms are permitted provided that this notice is preserved and that due credit is given to the University of Michigan at Ann Arbor. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. This software is provided ``as is'' without express or implied warranty.

Open SSL

#### LICENSE ISSUES

=====

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

## Text of Third-Party Software Licenses

See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

### OpenSSL License

-----

```
/* =====
* Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* =====
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
```

```
*/
```

#### Original SSLeay License

---

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *    Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed. i.e. this code cannot simply be
 * copied and put under another distribution licence
```

## Text of Third-Party Software Licenses

```
* [including the GNU Public Licence.]  
*/
```

### Open SSH

This file is part of the OpenSSH software.

The licences which components of this software fall under are as follows. First, we will summarize and say that all components are under a BSD licence, or a licence more free than that.

OpenSSH contains no GPL code.

1)

```
* Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland All  
* rights reserved  
*  
* As far as I am concerned, the code I have written for this software can  
* be used freely for any purpose. Any derived versions of this software  
* must be clearly marked as such, and if the derived work is incompatible  
* with the protocol description in the RFC file, it must be called by a  
* name other than "ssh" or "Secure Shell".
```

[Tatu continues]

```
* However, I am not implying to give any licenses to any patents or  
* copyrights held by third parties, and the software includes parts that  
* are not under my direct control. As far as I know, all included source  
* code is used in accordance with the relevant license agreements and can  
* be used freely for any purpose (the GNU license being the most  
* restrictive); see below for details.
```

[However, none of that term is relevant at this point in time. All of these restrictively licenced software components which he talks about have been removed from OpenSSH, i.e.,

- RSA is no longer included, found in the OpenSSL library
- IDEA is no longer included, its use is deprecated
- DES is now external, in the OpenSSL library
- GMP is no longer used, and instead we call BN code from OpenSSL
- Zlib is now external, in a library
- The make-ssh-known-hosts script is no longer included
- TSS has been removed
- MD5 is now external, in the OpenSSL library
- RC4 support has been replaced with ARC4 support from OpenSSL
- Blowfish is now external, in the OpenSSL library

[The licence continues]

Note that any information and cryptographic algorithms used in this software are publicly available on the Internet and at any major bookstore, scientific library, and patent office worldwide. More information can be found e.g. at "<http://www.cs.hut.fi/crypto>".

The legal status of this program is some combination of all these permissions and restrictions. Use only at your own responsibility. You will be responsible for any legal consequences yourself; I am not making any claims whether possessing or using this is legal or not in your country, and I am not

taking any responsibility on your behalf.

#### NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

- 2) The 32-bit CRC compensation attack detector in deattack.c was contributed by CORE SDI S.A. under a BSD-style license.

```
* Cryptographic attack detector for ssh - source code
*
* Copyright (c) 1998 CORE SDI S.A., Buenos Aires, Argentina.
*
* All rights reserved. Redistribution and use in source and binary forms,
* with or without modification, are permitted provided that this copyright
* notice is retained.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
* WARRANTIES ARE DISCLAIMED. IN NO EVENT SHALL CORE SDI S.A. BE LIABLE FOR
* ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL
* DAMAGES RESULTING FROM THE USE OR MISUSE OF THIS SOFTWARE.
*
* Ariel Futoransky <futo@core-sdi.com> <http://www.core-sdi.com>
```

- 3) ssh-keyscan was contributed by David Mazieres under a BSD-style license.

```
* Copyright 1995, 1996 by David Mazieres <dm@lcs.mit.edu>.
*
* Modification and redistribution in source and binary forms is permitted
* provided that due credit is given to the author and the OpenBSD project
* by leaving this copyright notice intact.
```

- 4) The Rijndael implementation by Vincent Rijmen, Antoon Bosselaers and Paulo Barreto is in the public domain and distributed with the following license:

```
* @version 3.0 (December 2000)
*
* Optimised ANSI C code for the Rijndael cipher (now AES)
*
* @author Vincent Rijmen <vincent.rijmen@esat.kuleuven.ac.be> @author
* Antoon Bosselaers <antoon.bosselaers@esat.kuleuven.ac.be> @author Paulo
* Barreto <paulo.barreto@terra.com.br>
```

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

- 5) One component of the ssh source code is under a 3-clause BSD license, held by the University of California, since we pulled these parts from original Berkeley code.

```
* Copyright (c) 1983, 1990, 1992, 1993, 1995 The Regents of the University  
* of California. All rights reserved.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions are  
* met: 1. Redistributions of source code must retain the above copyright  
* notice, this list of conditions and the following disclaimer. 2.  
* Redistributions in binary form must reproduce the above copyright  
* notice, this list of conditions and the following disclaimer in the  
* documentation and/or other materials provided with the distribution. 3.  
* Neither the name of the University nor the names of its contributors may  
* be used to endorse or promote products derived from this software  
* without specific prior written permission.  
*  
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND  
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS  
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF  
* THE POSSIBILITY OF SUCH DAMAGE.
```

- 6) Remaining components of the software are provided under a standard 2-term BSD licence with the following names as copyright holders:

Markus Friedl Theo de Raadt Niels Provos Dug Song Aaron Campbell Damien  
Miller Kevin Steves Daniel Kouril Wesley Griffin Per Allansson Nils Nordman  
Simon Wilkinson

```
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions are  
* met: 1. Redistributions of source code must retain the above copyright  
* notice, this list of conditions and the following disclaimer. 2.  
* Redistributions in binary form must reproduce the above copyright  
* notice, this list of conditions and the following disclaimer in the  
* documentation and/or other materials provided with the distribution.
```

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

-----  
\$OpenBSD: LICENCE,v 1.19 2004/08/30 09:18:08 markus Exp \$

#### OpenSymphony OGNL

##### The OpenSymphony Software License, Version 1.1

(this license is derived and fully compatible with the Apache Software License - see <http://www.apache.org/LICENSE.txt>)

Copyright (c) 2001-2004 The OpenSymphony Group. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:  
"This product includes software developed by the  
OpenSymphony Group (<http://www.opensymphony.com/>)."  
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "OpenSymphony" and "The OpenSymphony Group" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [license@opensymphony.com](mailto:license@opensymphony.com).
5. Products derived from this software may not be called "OpenSymphony" or "XWork", nor may "OpenSymphony" or "XWork" appear in their name, without prior written permission of the OpenSymphony Group.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

## Text of Third-Party Software Licenses

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

OpenSymphony Quartz  
Apache License v2

OpenSymphony XWork

The OpenSymphony Software License, Version 1.1

(this license is derived and fully compatible with the Apache Software License - see <http://www.apache.org/LICENSE.txt>)

Copyright (c) 2001-2004 The OpenSymphony Group. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:  
"This product includes software developed by the  
OpenSymphony Group (<http://www.opensymphony.com/>)."  
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "OpenSymphony" and "The OpenSymphony Group" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [license@opensymphony.com](mailto:license@opensymphony.com) .
5. Products derived from this software may not be called "OpenSymphony" or "XWork", nor may "OpenSymphony" or "XWork" appear in their name, without prior written permission of the OpenSymphony Group.

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

OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OProfile  
GNU Public License v2

This is an alpha release version of oprofile, a transparent low-overhead system-wide profiler.

You can find some documentation in the doc/ directory.

Please visit the oprofile website at : <http://oprofile.sf.net/>

oprofile was written by John Levon <levon@movementarian.org> and Philippe Elie <phil.el@wanadoo.fr>.

Maynard Johnson <maynardj@us.ibm.com> is the current maintainer.

Dave Jones <davej@suse.de> provided bug fixes and support for the AMD Athlon, and AMD Hammer families of CPUs.

Bob Montgomery <bobm@fc.hp.com> provided bug fixes, the initial RTC driver and the initial ia64 driver.

Will Cohen <wcohen@redhat.com> integrated the ia64 driver into the oprofile release, and contributed bug fixes and several cleanups.

Graydon Hoare <graydon@redhat.com> provided P4 port, bug fixes and cleanups.

Ralf Baechle <ralf@linux-mips.org> provided the MIPS port.

Other contributors are listed in the ChangeLog.

Building

-----

Please read the installation instructions in doc/oprofile.html or <http://oprofile.sourceforge.net/doc/install.html>.

Quick start :

(If using CVS: ./autogen.sh first. You need automake 1.5 or higher. You can specify a different version, e.g.

ACLOCAL=aclocal-1.5 AUTOMAKE=automake-1.5 AUTOCONF=autoconf-2.13 AUTOHEADER=autoheader-2.13 ./autogen.sh)

2.4 kernels

./configure --with-linux=/path/to/kernel/source

2.6 kernels

./configure --with-kernel-support

Oracle JDBC

OTN License

Oracle Technology Network Development and Distribution License Terms

**Export Controls on the Programs**

Selecting the "Accept License Agreement" button is a confirmation of your agreement that you comply, now and during the trial term, with each of the following statements:

-You are not a citizen, national, or resident of, and are not under control of, the government of Cuba, Iran, Sudan, Libya, North Korea, Syria, nor any country to which the United States has prohibited export.

#NAME?

or indirectly, to the above mentioned countries nor to citizens, nationals or residents of those countries.

#NAME?

Specially Designated Nationals, Specially Designated Terrorists, and Specially Designated Narcotic Traffickers, nor are you listed on the United States Department of Commerce Table of Denial Orders.

You will not download or otherwise export or re-export the Programs, directly or indirectly, to persons on the above mentioned lists.

You will not use the Programs for, and will not allow the Programs to be used for, any purposes prohibited by United States law, including, without limitation, for the development, design, manufacture or production of nuclear, chemical or biological weapons of mass destruction.

**EXPORT RESTRICTIONS**

You agree that U.S. export control laws and other applicable export and import laws govern your use of the programs, including technical data; additional information can be found on Oracle's Global Trade Compliance web site (<http://www.oracle.com/products/export>).

You agree that neither the programs nor any direct product thereof will be exported, directly, or indirectly, in violation of these laws, or will be used for any purpose prohibited by these laws including, without limitation, nuclear, chemical, or biological weapons proliferation.

**Oracle Employees:** Under no circumstances are Oracle Employees authorized to download software for the purpose of distributing it to customers. Oracle products are available to employees for internal use or demonstration purposes only. In keeping with Oracle's trade compliance obligations under U.S. and applicable multilateral law, failure to comply with this policy could result in disciplinary action up to and including termination.

Note: You are bound by the Oracle Technology Network ("OTN") License Agreement terms. The OTN License Agreement terms also apply to all updates you receive under your Technology Track subscription.

The OTN License Agreement terms below supercede any shrinkwrap license on the

OTN Technology Track software CDs and previous OTN License terms (including the Oracle Program License as modified by the OTN Program Use Certificate).

**Oracle Technology Network Development and Distribution License Agreement**  
We, "us," and "our" refers to Oracle America, Inc., for and on behalf of itself and its subsidiaries and affiliates under common control. "You" and your refers to the individual or entity that wishes to use the programs from Oracle. "Programs" refers to the software product you wish to download and use and program documentation. "License" refers to your right to use the programs under the terms of this agreement. This agreement is governed by the substantive and procedural laws of California. You and Oracle agree to submit to the exclusive jurisdiction of, and venue in, the courts of San Francisco, San Mateo, or Santa Clara counties in California in any dispute arising out of or relating to this agreement.

We are willing to license the programs to you only upon the condition that you accept all of the terms contained in this agreement. Read the terms carefully and select the "Accept" button at the bottom of the page to confirm your acceptance. If you are not willing to be bound by these terms, select the "Do Not Accept" button and the registration process will not continue.

#### **License Rights**

We grant you a nonexclusive, nontransferable limited license to use the programs: (a) for purposes of developing, testing, prototyping and running applications you have developed for your own internal data processing operations; (b) to distribute the programs with applications you have developed to your customers provided that each such licensee agrees to license terms consistent with the terms of this Agreement, you do not charge your end users any additional fees for the use of the programs, and your end users may only use the programs to run your applications for their own business operations; and (c) to use the programs to provide third party demonstrations and training. You are not permitted to use the programs for any purpose other than as permitted under this Agreement. If you want to use the programs for any purpose other than as expressly permitted under this agreement you must contact us, or an Oracle reseller, to obtain the appropriate license. We may audit your use and distribution of the programs. Program documentation is either shipped with the programs, or documentation may accessed online at <http://otn.oracle.com/docs>.

#### **Ownership and Restrictions**

We retain all ownership and intellectual property rights in the programs. You may make a sufficient number of copies of the programs for the licensed use and one copy of the programs for backup purposes.

You may not:

- use the programs for any purpose other than as provided above;
- distribute the programs unless accompanied with your applications;
- charge your end users for use of the programs;
- remove or modify any program markings or any notice of our proprietary rights;

#NAME?

functionality of the programs, except for training your licensed users;

#NAME?

the programs to any individual or entity except as provided under this agreement;

- cause or permit reverse engineering (unless required by law for interoperability), disassembly or decompilation of the programs;

#NAME?

#### Program Distribution

We grant you a nonexclusive, nontransferable right to copy and distribute the programs to your end users provided that you do not charge your end users for use of the programs and provided your end users may only use the programs to run your applications for their business operations. Prior to distributing the programs you shall require your end users to execute an agreement binding them to terms consistent with those contained in this section and the sections of this agreement entitled "License Rights," "Ownership and Restrictions," "Export," "Disclaimer of Warranties and Exclusive Remedies," "No Technical Support," "End of Agreement," "Relationship Between the Parties," and "Open Source." You must also include a provision stating that your end users shall have no right to distribute the programs, and a provision specifying us as a third party beneficiary of the agreement. You are responsible for obtaining these agreements with your end users.

You agree to: (a) defend and indemnify us against all claims and damages caused by your distribution of the programs in breach of this agreements and/or failure to include the required contractual provisions in your end user agreement as stated above; (b) keep executed end user agreements and records of end user information including name, address, date of distribution and identity of programs distributed; (c) allow us to inspect your end user agreements and records upon request; and, (d) enforce the terms of your end user agreements so as to effect a timely cure of any end user breach, and to notify us of any breach of the terms.

#### Export

You agree that U.S. export control laws and other applicable export and import laws govern your use of the programs, including technical data; additional information can be found on Oracle's Global Trade Compliance web site located at <http://www.oracle.com/products/export/index.html?content.html>. You agree that neither the programs nor any direct product thereof will be exported, directly, or indirectly, in violation of these laws, or will be used for any purpose prohibited by these laws including, without limitation, nuclear, chemical, or biological weapons proliferation.

#### Disclaimer of Warranty and Exclusive Remedies

THE PROGRAMS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. WE FURTHER DISCLAIM ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.

IN NO EVENT SHALL WE BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR DATA USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR TORT, EVEN IF WE HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. OUR ENTIRE LIABILITY FOR DAMAGES HEREUNDER SHALL IN NO EVENT EXCEED ONE THOUSAND DOLLARS (U.S. \$1,000).

#### No Technical Support

Our technical support organization will not provide technical support, phone support, or updates to you for the programs licensed under this agreement.

#### Restricted Rights

If you distribute a license to the United States government, the programs, including documentation, shall be considered commercial computer software and you will place a legend, in addition to applicable copyright notices, on the documentation, and on the media label, substantially similar to the following:

**NOTICE OF RESTRICTED RIGHTS**

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, including documentation, shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

**End of Agreement**

You may terminate this agreement by destroying all copies of the programs. We have the right to terminate your right to use the programs if you fail to comply with any of the terms of this agreement, in which case you shall destroy all copies of the programs.

**Relationship Between the Parties**

The relationship between you and us is that of licensee/licensor. Neither party will represent that it has any authority to assume or create any obligation, express or implied, on behalf of the other party, nor to represent the other party as agent, employee, franchisee, or in any other capacity. Nothing in this agreement shall be construed to limit either party's right to independently develop or distribute software that is functionally similar to the other party's products, so long as proprietary information of the other party is not included in such software.

**Open Source**

Open Source software - software available without charge for use, modification and distribution - is often licensed under terms that require the user to make the user's modifications to the Open Source software or any software that the user 'combines' with the Open Source software freely available in source code form. If you use Open Source software in conjunction with the programs, you must ensure that your use does not: (i) create, or purport to create, obligations of us with respect to the Oracle programs; or (ii) grant, or purport to grant, to any third party any rights to or immunities under our intellectual property or proprietary rights in the Oracle programs. For example, you may not develop a software program using an Oracle program and an Open Source program where such use results in a program file(s) that contains code from both the Oracle program and the Open Source program (including without limitation libraries) if the Open Source program is licensed under a license that requires any "modifications" be made freely available. You also may not combine the Oracle program with programs licensed under the GNU General Public License ("GPL") in any manner that could cause, or could be interpreted or asserted to cause, the Oracle program or any modifications thereto to become subject to the terms of the GPL.

**Entire Agreement**

You agree that this agreement is the complete agreement for the programs and licenses, and this agreement supersedes all prior or contemporaneous agreements or representations. If any term of this agreement is found to be invalid or unenforceable, the remaining provisions will remain effective.

Last updated: 01/24/09

Should you have any questions concerning this License Agreement, or if you desire to contact Oracle for any reason, please write:

## Text of Third-Party Software Licenses

Oracle America, Inc.  
500 Oracle Parkway,  
Redwood City, CA 94065

Oracle may contact you to ask if you had a satisfactory experience installing and using this OTN software download.

Perl  
Artistic License

PG Foundry IP4R  
PostgreSQL License

php

```
-----  
      The PHP License, version 3.01  
Copyright (c) 1999 - 2006 The PHP Group. All rights reserved.  
-----
```

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "PHP" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact group@php.net.
4. Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo"
5. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number.  
Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.
6. Redistributions of any form whatsoever must retain the following acknowledgment:  
"This product includes PHP software, freely available from

<<http://www.php.net/software/>>".

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

-----

This software consists of voluntary contributions made by many individuals on behalf of the PHP Group.

The PHP Group can be contacted via Email at [group@php.net](mailto:group@php.net).

For more information on the PHP Group and the PHP project, please see <<http://www.php.net>>.

PHP includes the Zend Engine, freely available at <<http://www.zend.com>>.

#### PlotKit

Copyright (c) 2006, Alastair Tse  
All rights reserved.

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

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

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

## Text of Third-Party Software Licenses

### Popup.js

```
*****
* Portions of this code from:
*
* http://www.lokeshdhakar.com/projects/lightbox/lightbox.js
*
* Lightbox JS: Fullsize Image Overlays
* by Lokesh Dhakar - http://www.huddletogether.com
*
* For more information on this script, visit:
* http://huddletogether.com/projects/lightbox/
*
* Licensed under the Creative Commons Attribution 2.5 License - http://creativecommons.org/licenses/by/2.5/
* (basically, do anything you want, just leave my name and link)
*****
```

### Popt

Copyright (c) 1998 Red Hat Software

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

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

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

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

### Porter Stemmer

Porter Stemmer Algorithm- <http://tartarus.org/~martin/PorterStemmer/>

#1. What is the licensing arrangement for this software?

This question has become very popular recently (the period 2008-2009), despite the clear statement above that ''all these encodings of the algorithm can be used free of charge for any purpose.'' The problem I think is that intellectual property has become such a major issue that some more formal statement is expected. So to restate it:

The software is completely free for any purpose, unless notes at the head of the program text indicates otherwise (which is rare). In any case, the notes about licensing are never more restrictive than the BSD License.

In every case where the software is not written by me (Martin Porter), this licensing arrangement has been endorsed by the contributor, and it is therefore unnecessary to ask the contributor again to confirm it.

I have not asked any contributors (or their employers, if they have them) for proofs that they have the right to distribute their software in this way.

(For anyone taking software from the Snowball website, the position is similar but simpler. There, all the software is issued under the BSD License, and for contributions not written by Martin Porter and Richard Boulton, we have again not asked the authors, or the authors' employers, for proofs that they have such distribution rights.)

PostgreSQL  
PostgreSQL License

PostgreSQL datetime.h  
PostgreSQL License

PostgreSQL JDBC

Copyright (c) 1997-2008, PostgreSQL Global Development Group  
All rights reserved.

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

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

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

PSQL ODBC  
GNU Public License v2

\*\*\*\*\*

## Text of Third-Party Software Licenses

PSQLODBC.DLL - A library to talk to the PostgreSQL DBMS using ODBC.

Copyright (C) 1998                    Insight Distribution Systems  
Copyright (C) 1998 - 2005        The PostgreSQL Global Development Group

Multibyte support was added by Sankyo Unyu Service, (C) 2001.

The code contained in this library is based on code written by Christian Czezatke and Dan McGuirk, (C) 1996.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library (see "license.txt"); if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

How to contact the authors:

email: pgsql-odbc@postgresql.org  
website: <http://pgfoundry.org/projects/psqlodbc>

\*\*\*\*\*\*/

Ptr.h

```
/* Copyright (c) 1993-9, David R. Cheriton. All rights reserved.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions are  
* met:  
*  
* * Redistributions of source code must retain the above copyright  
*   notice, this list of conditions and the following disclaimer.  
*  
* * Redistributions in binary form must reproduce the above copyright  
*   notice, this list of conditions and the following disclaimer in the  
*   documentation and/or other materials provided with the distribution.  
*  
* * Neither the name of David R. Cheriton nor the names of his  
*   contributors may be used to endorse or promote products derived from  
*   this software without specific prior written permission.  
*  
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```

```
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* Copyright (c) 2002-2004 Kealia, Inc. All rights reserved.
*/
```

Public Domain Curses  
MIT License

Welcome to PDCurses!

=====

Public Domain Curses, aka PDCurses, is an implementation of X/Open  
curses for multiple platforms. The latest version can be found at:

<http://pdcurse.sourceforge.net/>

For changes, see the HISTORY file.

Legal Stuff

-----

The core package is in the public domain, but small portions of PDCurses  
are subject to copyright under various licenses. Each directory  
contains a README file, with a section titled "Distribution Status"  
which describes the status of the files in that directory.

If you use PDCurses in an application, an acknowledgement would be  
appreciated, but is not mandatory. If you make corrections or  
enhancements to PDCurses, please forward them to the current maintainer  
for the benefit of other users.

This software is provided AS IS with NO WARRANTY whatsoever.

Distribution Status

-----

All files in this directory except configure, config.guess and  
config.sub are released to the Public Domain. config.guess and  
config.sub are under the GPL; configure is under a free license  
described within it.

-----  
demos/README

-----  
Public Domain, except for rain.c and worm.c, which are under the ncurses  
license (MIT-like).

## Text of Third-Party Software Licenses

### x11/README

-----  
As of April 13, 2006, the files in this directory are released to the Public Domain, except for ScrollBox\*, which are under essentially the MIT X License.

## Python

### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release	Derived from	Year	Owner	GPL-compatible? (1)
0.9.0 thru 1.2		1991-1995	CWI	yes
1.3 thru 1.5.2	1.2	1995-1999	CNRI	yes
1.6	1.5.2	2000	CNRI	no
2.0	1.6	2000	BeOpen.com	no
1.6.1	1.6	2001	CNRI	yes (2)
2.1	2.0+1.6.1	2001	PSF	no
2.0.1	2.0+1.6.1	2001	PSF	yes
2.1.1	2.1+2.0.1	2001	PSF	yes
2.2	2.1.1	2001	PSF	yes
2.1.2	2.1.1	2002	PSF	yes
2.1.3	2.1.2	2002	PSF	yes
2.2.1	2.2	2002	PSF	yes
2.2.2	2.2.1	2002	PSF	yes
2.2.3	2.2.2	2003	PSF	yes
2.3	2.2.2	2002-2003	PSF	yes
2.3.1	2.3	2002-2003	PSF	yes
2.3.2	2.3.1	2002-2003	PSF	yes
2.3.3	2.3.2	2002-2003	PSF	yes
2.3.4	2.3.3	2004	PSF	yes
2.3.5	2.3.4	2005	PSF	yes

2.4	2.3	2004	PSF	yes
2.4.1	2.4	2005	PSF	yes
2.4.2	2.4.1	2005	PSF	yes
2.4.3	2.4.2	2006	PSF	yes
2.4.4	2.4.3	2006	PSF	yes
2.5	2.4	2006	PSF	yes
2.5.1	2.5	2007	PSF	yes
2.5.2	2.5.2	2008	PSF	yes

## Footnotes:

- (1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.
- (2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

---

## PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

---

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

## Text of Third-Party Software Licenses

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

---

#### BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant

permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonglabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

---

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal

## Text of Third-Party Software Licenses

intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions.

Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

### CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

---

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

This is Python version 2.5.2

---

Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Python Software Foundation.  
All rights reserved.

Copyright (c) 2000 BeOpen.com.  
All rights reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.  
All rights reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum.  
All rights reserved.

License information

-----

See the file "LICENSE" for information on the history of this software, terms & conditions for usage, and a DISCLAIMER OF ALL WARRANTIES.

This Python distribution contains no GNU General Public Licensed (GPLed) code so it may be used in proprietary projects just like prior Python distributions. There are interfaces to some GNU code but these are entirely optional.

All trademarks referenced herein are property of their respective holders.

What's new in this release?

-----

See the file "Misc/NEWS".

If you don't read instructions

-----

Congratulations on getting this far. :-)

To start building right away (on UNIX): type "./configure" in the current directory and when it finishes, type "make". This creates an executable "./python"; to install in /usr/local, first do "su root" and then "make install".

The section 'Build instructions' below is still recommended reading.

What is Python anyway?

-----

Python is an interpreted, interactive object-oriented programming language suitable (amongst other uses) for distributed application development, scripting, numeric computing and system testing. Python is often compared to Tcl, Perl, Java, JavaScript, Visual Basic or Scheme. To find out more about what Python can do for you, point your browser to <http://www.python.org/>.

How do I learn Python?

-----

The official tutorial is still a good place to start; see <http://docs.python.org/> for online and downloadable versions, as well as a list of other introductions, and reference documentation.

There's a quickly growing set of books on Python. See <http://wiki.python.org/moin/PythonBooks> for a list.

## Text of Third-Party Software Licenses

### Documentation

-----

All documentation is provided online in a variety of formats. In order of importance for new users: Tutorial, Library Reference, Language Reference, Extending & Embedding, and the Python/C API. The Library Reference is especially of immense value since much of Python's power is described there, including the built-in data types and functions!

All documentation is also available online at the Python web site (<http://docs.python.org/>, see below). It is available online for occasional reference, or can be downloaded in many formats for faster access. The documentation is available in HTML, PostScript, PDF, and LaTeX formats; the LaTeX version is primarily for documentation authors, translators, and people with special formatting requirements.

Unfortunately, new-style classes (new in Python 2.2) have not yet been integrated into Python's standard documentation. A collection of pointers to what has been written is at:

<http://www.python.org/doc/newstyle.html>

### Web sites

-----

New Python releases and related technologies are published at <http://www.python.org/>. Come visit us!

There's also a Python community web site at <http://starship.python.net/>.

### Newsgroups and Mailing Lists

-----

Read `comp.lang.python`, a high-volume discussion newsgroup about Python, or `comp.lang.python.announce`, a low-volume moderated newsgroup for Python-related announcements. These are also accessible as mailing lists: see <http://www.python.org/community/lists.html> for an overview of these and many other Python-related mailing lists.

Archives are accessible via the Google Groups Usenet archive; see <http://groups.google.com/>. The mailing lists are also archived, see <http://www.python.org/community/lists.html> for details.

### Bug reports

-----

To report or search for bugs, please use the Python Bug Tracker at <http://bugs.python.org>.

### Patches and contributions

To submit a patch or other contribution, please use the Python Patch Manager at <http://bugs.python.org>. Guidelines for patch submission may be found at <http://www.python.org/dev/patches/>.

If you have a proposal to change Python, it's best to submit a Python Enhancement Proposal (PEP) first. All current PEPs, as well as guidelines for submitting a new PEP, are listed at <http://www.python.org/dev/peps/>.

#### Questions

-----

For help, if you can't find it in the manuals or on the web site, it's best to post to the comp.lang.python or the Python mailing list (see above). If you specifically don't want to involve the newsgroup or mailing list, send questions to [help@python.org](mailto:help@python.org) (a group of volunteers who answer questions as they can). The newsgroup is the most efficient way to ask public questions.

#### Build instructions

=====

Before you can build Python, you must first configure it. Fortunately, the configuration and build process has been automated for Unix and Linux installations, so all you usually have to do is type a few commands and sit back. There are some platforms where things are not quite as smooth; see the platform specific notes below. If you want to build for multiple platforms sharing the same source tree, see the section on VPATH below.

Start by running the script "`./configure`", which determines your system configuration and creates the Makefile. (It takes a minute or two -- please be patient!) You may want to pass options to the configure script -- see the section below on configuration options and variables. When it's done, you are ready to run make.

To build Python, you normally type "make" in the toplevel directory. If you have changed the configuration, the Makefile may have to be rebuilt. In this case you may have to run make again to correctly build your desired target. The interpreter executable is built in the top level directory.

Once you have built a Python interpreter, see the subsections below on testing and installation. If you run into trouble, see the next section.

Previous versions of Python used a manual configuration process that involved editing the file Modules/Setup. While this file still exists and manual configuration is still supported, it is rarely needed any more: almost all modules are automatically built as appropriate under guidance of the setup.py script, which is run by Make after the interpreter has been built.

#### Troubleshooting

-----

See also the platform specific notes in the next section.

If you run into other trouble, see the FAQ  
(<http://www.python.org/doc/faq>) for hints on what can go wrong, and how to fix it.

If you rerun the configure script with different options, remove all object files by running "make clean" before rebuilding. Believe it or not, "make clean" sometimes helps to clean up other inexplicable problems as well. Try it before sending in a bug report!

If the configure script fails or doesn't seem to find things that should be there, inspect the config.log file.

If you get a warning for every file about the -Olimit option being no longer supported, you can ignore it. There's no foolproof way to know whether this option is needed; all we can do is test whether it is accepted without error. On some systems, e.g. older SGI compilers, it is essential for performance (specifically when compiling ceval.c, which has more basic blocks than the default limit of 1000). If the warning bothers you, edit the Makefile to remove "-Olimit 1500" from the OPT variable.

If you get failures in test\_long, or sys.maxint gets set to -1, you are probably experiencing compiler bugs, usually related to optimization. This is a common problem with some versions of gcc, and some vendor-supplied compilers, which can sometimes be worked around by turning off optimization. Consider switching to stable versions (gcc 2.95.2, gcc 3.x, or contact your vendor.)

From Python 2.0 onward, all Python C code is ANSI C. Compiling using old K&R-C-only compilers is no longer possible. ANSI C compilers are available for all modern systems, either in the form of updated compilers from the vendor, or one of the free compilers (gcc).

If "make install" fails mysteriously during the "compiling the library" step, make sure that you don't have any of the PYTHONPATH or PYTHONHOME environment variables set, as they may interfere with the newly built executable which is compiling the library.

#### Unsupported systems

---

A number of features are not supported in Python 2.5 anymore. Some support code is still present, but will be removed in Python 2.6. If you still need to use current Python versions on these systems, please send a message to [python-dev@python.org](mailto:python-dev@python.org) indicating that you volunteer to support this system. For a more detailed discussion regarding no-longer-supported and resupporting platforms, as well as a list of platforms that became or will be unsupported, see PEP 11.

More specifically, the following systems are not supported any longer:

- SunOS 4
- DYNIX
- dgux
- Minix
- NeXT

- Irix 4 and --with-sgi-dl
- Linux 1
- Systems defining \_\_d6\_pthread\_create (configure.in)
- Systems defining PY\_PTHREAD\_D4, PY\_PTHREAD\_D6, or PY\_PTHREAD\_D7 in thread\_pthread.h
- Systems using --with-dl-dld
- Systems using --without-universal-newlines
- MacOS 9

The following systems are still supported in Python 2.5, but support will be dropped in 2.6:

- Systems using --with-wctype-functions
- Win9x, WinME

Notice on install in Windows 98 and Windows Me

Following Microsoft's closing of Extended Support for Windows 98/ME (July 11, 2006), Python 2.6 will stop supporting these platforms. Python development and maintainability becomes easier (and more reliable) when platform specific code targeting OSes with few users and no dedicated expert developers is taken out. The vendor also warns that the OS versions listed above "can expose customers to security risks" and recommends upgrade.

Platform specific notes

(Some of these may no longer apply. If you find you can build Python on these platforms without the special directions mentioned here, submit a documentation bug report to SourceForge (see Bug Reports above) so we can remove them!)

Unix platforms: If your vendor still ships (and you still use) Berkeley DB 1.85 you will need to edit Modules/Setup to build the bsddb185 module and add a line to sitecustomize.py which makes it the default. In Modules/Setup a line like

```
bsddb185 bsddbmodule.c
```

should work. (You may need to add -I, -L or -l flags to direct the compiler and linker to your include files and libraries.)

XXX I think this next bit is out of date:

64-bit platforms: The modules audioop, imageop and rgbiimg don't work. The setup.py script disables them on 64-bit installations. Don't try to enable them in the Modules/Setup file. They contain code that is quite wordsize sensitive. (If you have a fix, let us know!)

Solaris: When using Sun's C compiler with threads, at least on Solaris 2.5.1, you need to add the "-mt" compiler option (the simplest way is probably to specify the compiler with this option as the "CC" environment variable when running the configure script).

When using GCC on Solaris, beware of binutils 2.13 or GCC versions built using it. This mistakenly enables the -zcombreloc option which creates broken shared libraries on Solaris. binutils 2.12 works, and the binutils maintainers are aware of the problem. Binutils 2.13.1 only partially fixed things. It appears that 2.13.2 solves the problem completely. This problem is known to occur with Solaris 2.7 and 2.8, but may also affect earlier and later versions of the OS.

When the dynamic loader complains about errors finding shared libraries, such as

```
ld.so.1: ./python: fatal: libstdc++.so.5: open failed:  
No such file or directory
```

you need to first make sure that the library is available on your system. Then, you need to instruct the dynamic loader how to find it. You can choose any of the following strategies:

1. When compiling Python, set LD\_RUN\_PATH to the directories containing missing libraries.
2. When running Python, set LD\_LIBRARY\_PATH to these directories.
3. Use crle(8) to extend the search path of the loader.
4. Modify the installed GCC specs file, adding -R options into the \*link: section.

The complex object fails to compile on Solaris 10 with gcc 3.4 (at least up to 3.4.3). To work around it, define Py\_HUGE\_VAL as HUGE\_VAL(), e.g.:

```
make CPPFLAGS='-D"Py_HUGE_VAL=HUGE_VAL()" -I. -I$(srcdir)/Include'  
../python setup.py CPPFLAGS='-D"Py_HUGE_VAL=HUGE_VAL()"'
```

**Linux:** A problem with threads and fork() was tracked down to a bug in the pthreads code in glibc version 2.0.5; glibc version 2.0.7 solves the problem. This causes the popen2 test to fail; problem and solution reported by Pablo Bleyer.

**Red Hat Linux:** Red Hat 9 built Python2.2 in UCS-4 mode and hacked Tcl to support it. To compile Python2.3 with Tkinter, you will need to pass --enable-unicode=ucs4 flag to ./configure.

There's an executable /usr/bin/python which is Python 1.5.2 on most older Red Hat installations; several key Red Hat tools require this version. Python 2.1.x may be installed as /usr/bin/python2. The Makefile installs Python as /usr/local/bin/python, which may or may not take precedence over /usr/bin/python, depending on how you have set up \$PATH.

**FreeBSD 3.x and probably platforms with NCurses that use libmytinfo or similar:** When using cursesmodule, the linking is not done in the correct order with the defaults. Remove "-ltermcap" from the readline entry in Setup, and use as curses entry: "curses cursesmodule.c -lmytinfo -lncurses -ltermcap" - "mytinfo" (so called on FreeBSD) should be the name of the auxiliary library required on your platform. Normally, it would be linked automatically, but not necessarily in the correct order.

**BSDI:** BSDI versions before 4.1 have known problems with threads, which can cause strange errors in a number of modules (for instance, the 'test\_signal' test script will hang forever.) Turning off threads (with --with-threads=no) or upgrading to BSDI 4.1 solves this problem.

**DEC Unix:** Run configure with --with-dec-threads, or with --with-threads=no if no threads are desired (threads are on by default). When using GCC, it is possible to get an internal compiler error if optimization is used. This was reported for GCC 2.7.2.3 on selectmodule.c. Manually compile the affected file without optimization to solve the problem.

**DEC Ultrix:** compile with GCC to avoid bugs in the native compiler, and pass SHELL=/bin/sh5 to Make when installing.

**AIX:** A complete overhaul of the shared library support is now in place. See Misc/AIX-NOTES for some notes on how it's done. (The optimizer bug reported at this place in previous releases has been worked around by a minimal code change.) If you get errors about pthread\_\* functions, during compile or during testing, try setting CC to a thread-safe (reentrant) compiler, like "cc\_r". For full C++ module support, set CC="xlc\_r" (or CC="xlc" without thread support).

**AIX 5.3:** To build a 64-bit version with IBM's compiler, I used the following:

```
export PATH=/usr/bin:/usr/vacpp/bin
./configure --with-gcc="xlc_r -q64" --with-cxx="xlc_r -q64" \
            --disable-ipv6 AR="ar -X64"
make
```

**HP-UX:** When using threading, you may have to add -D\_REENTRANT to the OPT variable in the top-level Makefile; reported by Pat Knight, this seems to make a difference (at least for HP-UX 10.20) even though pyconfig.h defines it. This seems unnecessary when using HP/UX 11 and later - threading seems to work "out of the box".

**HP-UX ia64:** When building on the ia64 (Itanium) platform using HP's compiler, some experience has shown that the compiler's optimiser produces a completely broken version of python (see <http://www.python.org/sf/814976>). To work around this, edit the Makefile and remove -O from the OPT line.

To build a 64-bit executable on an Itanium 2 system using HP's compiler, use these environment variables:

```
CC=cc
CXX=aCC
BASECFLAGS="+DD64"
LDFLAGS="+DD64 -lxnet"
```

and call configure as:

```
./configure --without-gcc
```

then \*unset\* the environment variables again before running

## Text of Third-Party Software Licenses

make. (At least one of these flags causes the build to fail if it remains set.) You still have to edit the Makefile and remove -O from the OPT line.

HP PA-RISC 2.0: A recent bug report (<http://www.python.org/sf/546117>) suggests that the C compiler in this 64-bit system has bugs in the optimizer that break Python. Compiling without optimization solves the problems.

SCO: The following apply to SCO 3 only; Python builds out of the box on SCO 5 (or so we've heard).

1) Everything works much better if you add `-U__STDC__` to the defs. This is because all the SCO header files are broken. Anything that isn't mentioned in the C standard is conditionally excluded when `__STDC__` is defined.

2) Due to the U.S. export restrictions, SCO broke the crypt stuff out into a separate library, `libcrypt_i.a` so the LIBS needed be set to:

```
LIBS=' -lsocket -lcrypt_i'
```

UnixWare: There are known bugs in the math library of the system, as well as problems in the handling of threads (calling fork in one thread may interrupt system calls in others). Therefore, `test_math` and tests involving threads will fail until those problems are fixed.

QNX: Chris Herborth ([chrish@qnx.com](mailto:chrish@qnx.com)) writes:  
configure works best if you use GNU bash; a port is available on [ftp.qnx.com](ftp://ftp.qnx.com) in `/usr/free`. I used the following process to build, test and install Python 1.5.x under QNX:

1) `CONFIG_SHELL=/usr/local/bin/bash CC=cc RANLIB=: \ ./configure --verbose --without-gcc --with-libm=""`

2) edit Modules/Setup to activate everything that makes sense for your system... tested here at QNX with the following modules:

```
array, audioop, binascii, cPickle, cStringIO, cmath,
crypt, curses, errno, fcntl, gdbm, grp, imageop,
_locale, math, md5, new, operator, parser, pcre,
posix, pwd, readline, regex, reop, rgbiimg, rotor,
select, signal, socket, soundex, strop, struct,
syslog, termios, time, timing, zlib, audioop, rgbiimg
```

3) `make SHELL=/usr/local/bin/bash`

or, if you feel the need for speed:

```
make SHELL=/usr/local/bin/bash OPT="-5 -O1+nrt"
```

4) `make SHELL=/usr/local/bin/bash test`

Using GNU readline 2.2 seems to behave strangely, but I think that's a problem with my readline 2.2 port. :-\

5) `make SHELL=/usr/local/bin/bash install`

If you get SIGSEGVs while running Python (I haven't yet, but I've only run small programs and the test cases), you're probably running out of stack; the default 32k could be a little tight. To increase the stack size, edit the Makefile to read: LDFLAGS = -N 48k

**BeOS:** See Misc/BeOS-NOTES for notes about compiling/installing Python on BeOS R3 or later. Note that only the PowerPC platform is supported for R3; both PowerPC and x86 are supported for R4.

**Cray T3E:** Mark Hadfield ([m.hadfield@niwa.co.nz](mailto:m.hadfield@niwa.co.nz)) writes:  
 Python can be built satisfactorily on a Cray T3E but based on my experience with the NIWA T3E (2002-05-22, version 2.2.1) there are a few bugs and gotchas. For more information see a thread on comp.lang.python in May 2002 entitled "Building Python on Cray T3E".

- 1) Use Cray's cc and not gcc. The latter was reported not to work by Konrad Hinsen. It may work now, but it may not.
- 2) To set sys.platform to something sensible, pass the following environment variable to the configure script:

```
MACHDEP=unicosmk
```

- 2) Run configure with option "--enable-unicode=ucs4".
- 3) The Cray T3E does not support dynamic linking, so extension modules have to be built by adding (or uncommenting) lines in Modules/Setup. The minimum set of modules is

```
posix, new, _sre, unicodedata
```

On NIWA's vanilla T3E system the following have also been included successfully:

```
_codecs, _locale, _socket, _symtable, _testcapi, _weakref
array, binascii, cmath, cPickle, crypt, cStringIO, dbm
errno, fcntl, grp, math, md5, operator, parser, pcre, pwd
regex, rotor, select, struct, strop, syslog, termios
time, timing, xreadlines
```

- 4) Once the python executable and library have been built, make will execute setup.py, which will attempt to build remaining extensions and link them dynamically. Each of these attempts will fail but should not halt the make process. This is normal.
- 5) Running "make test" uses a lot of resources and causes problems on our system. You might want to try running tests singly or in small groups.

**SGI:** SGI's standard "make" utility (/bin/make or /usr/bin/make) does not check whether a command actually changed the file it is supposed to build. This means that whenever you say "make" it will redo the link step. The remedy is to use SGI's much smarter "smake" utility (/usr/sbin/smake), or GNU make. If you set the first line of the Makefile to #!/usr/sbin/smake

smake will be invoked by make (likewise for GNU make).

WARNING: There are bugs in the optimizer of some versions of SGI's compilers that can cause bus errors or other strange behavior, especially on numerical operations. To avoid this, try building with "make OPT=".

OS/2: If you are running Warp3 or Warp4 and have IBM's VisualAge C/C++ compiler installed, just change into the pc\os2vacpp directory and type NMAKE. Threading and sockets are supported by default in the resulting binaries of PYTHON15.DLL and PYTHON.EXE.

Monterey (64-bit AIX): The current Monterey C compiler (Visual Age) uses the OBJECT\_MODE={32|64} environment variable to set the compilation mode to either 32-bit or 64-bit (32-bit mode is the default). Presumably you want 64-bit compilation mode for this 64-bit OS. As a result you must first set OBJECT\_MODE=64 in your environment before configuring (./configure) or building (make) Python on Monterey.

Reliant UNIX: The thread support does not compile on Reliant UNIX, and there is a (minor) problem in the configure script for that platform as well. This should be resolved in time for a future release.

MacOSX: The tests will crash on both 10.1 and 10.2 with SEGV in test\_re and test\_sre due to the small default stack size. If you set the stack size to 2048 before doing a "make test" the failure can be avoided. If you're using the tcsh or csh shells, use "limit stacksize 2048" and for the bash shell (the default as of OSX 10.3), use "ulimit -s 2048".

On naked Darwin you may want to add the configure option "--disable-toolbox-glue" to disable the glue code for the Carbon interface modules. The modules themselves are currently only built if you add the --enable-framework option, see below.

On a clean OSX /usr/local does not exist. Do a  
"sudo mkdir -m 775 /usr/local"  
before you do a make install. It is probably not a good idea to do "sudo make install" which installs everything as superuser, as this may later cause problems when installing distutils-based additions.

Some people have reported problems building Python after using "fink" to install additional unix software. Disabling fink (remove all references to /sw from your .profile or .login) should solve this.

You may want to try the configure option "--enable-framework" which installs Python as a framework. The location can be set as argument to the --enable-framework option (default /Library/Frameworks). A framework install is probably needed if you want to use any Aqua-based GUI toolkit (whether Tkinter, wxPython, Carbon, Cocoa or anything else).

You may also want to try the configure option "--enable-universalsdk" which builds Python as a universal binary with support for the i386 and PPC architectures. This requires Xcode 2.1 or later to build.

See Mac/OSX/README for more information on framework and universal builds.

Cygwin: With recent (relative to the time of writing, 2001-12-19) Cygwin installations, there are problems with the interaction of dynamic linking and fork(). This manifests itself in build failures during the execution of setup.py.

There are two workarounds that both enable Python (albeit without threading support) to build and pass all tests on NT/2000 (and most likely XP as well, though reports of testing on XP would be appreciated).

The workarounds:

- (a) the band-aid fix is to link the \_socket module statically rather than dynamically (which is the default).

To do this, run "./configure --with-threads=no" including any other options you need (--prefix, etc.). Then in Modules/Setup uncomment the lines:

```
#SSL=/usr/local/ssl
#_socket socketmodule.c \
#       -DUSE_SSL -I$(SSL)/include -I$(SSL)/include/openssl \
#       -L$(SSL)/lib -lssl -lcrypto
```

and remove "local/" from the SSL variable. Finally, just run "make"!

- (b) The "proper" fix is to rebase the Cygwin DLLs to prevent base address conflicts. Details on how to do this can be found in the following mail:

<http://sources.redhat.com/ml/cygwin/2001-12/msg00894.html>

It is hoped that a version of this solution will be incorporated into the Cygwin distribution fairly soon.

Two additional problems:

- (1) Threading support should still be disabled due to a known bug in Cygwin pthreads that causes test\_threadedtempfile to hang.
- (2) The \_curses module does not build. This is a known Cygwin ncurses problem that should be resolved the next time that this package is released.

On older versions of Cygwin, test\_poll may hang and test\_strerror may fail.

The situation on 9X/Me is not accurately known at present. Some time ago, there were reports that the following regression tests failed:

```
test_pwd
test_select (hang)
test_socket
```

Due to the test\_select hang on 9X/Me, one should run the regression test using the following:

```
make TESTOPTS=' -l -x test_select' test
```

News regarding these platforms with more recent Cygwin versions would be appreciated!

AtheOS: From Octavian Cerna <tavy at ylabs.com>:

Before building:

Make sure you have shared versions of the libraries you want to use with Python. You will have to compile them yourself, or download precompiled packages.

Recommended libraries:

```
ncurses-4.2
readline-4.2a
zlib-1.1.4
```

Build:

```
$ ./configure --prefix=/usr/python
$ make
```

Python is always built as a shared library, otherwise dynamic loading would not work.

Testing:

```
$ make test
```

Install:

```
# make install
# pkgmanager -a /usr/python
```

AtheOS issues:

- large file support: due to a stdio bug in glibc/libio, access to large files may not work correctly. fseeko() tries to seek to a negative offset. ftello() returns a negative offset, it looks like a 32->64bit sign-extension issue. The lowlevel functions (open, lseek, etc) are OK.
- sockets: AF\_UNIX is defined in the C library and in Python, but not implemented in the system.
- select: poll is available in the C library, but does not work (It does not return POLLNVAL for bad fds and hangs).
- posix: statvfs and fstatvfs always return ENOSYS.
- disabled modules:
  - mmap: not yet implemented in AtheOS
  - nis: broken (on an unconfigured system  
yp\_get\_default\_domain() returns junk instead of

```

        error)
- dl: dynamic loading doesn't work via dlopen()
- resource: getrlimit and setrlimit are not yet
  implemented

- if you are getting segmentation faults, you probably are
  low on memory. AtheOS doesn't handle very well an
  out-of-memory condition and simply SEGVs the process.

```

Tested on:

```

AtheOS-0.3.7
gcc-2.95
binutils-2.10
make-3.78

```

#### Configuring the bsddb and dbm modules

Beginning with Python version 2.3, the PyBsddb package <<http://pybsddb.sf.net/>> was adopted into Python as the bsddb package, exposing a set of package-level functions which provide backwards-compatible behavior. Only versions 3.3 through 4.4 of Sleepycat's libraries provide the necessary API, so older versions aren't supported through this interface. The old bsddb module has been retained as bsddb185, though it is not built by default. Users wishing to use it will have to tweak Modules/Setup to build it. The dbm module will still be built against the Sleepycat libraries if other preferred alternatives (ndbm, gdbm) are not found.

#### Building the sqlite3 module

To build the sqlite3 module, you'll need the sqlite3 or libsqlite3 packages installed, including the header files. Many modern operating systems distribute the headers in a separate package to the library - often it will be the same name as the main package, but with a -dev or -devel suffix.

The version of pysqlite2 that's including in Python needs sqlite3 3.0.8 or later. setup.py attempts to check that it can find a correct version.

#### Configuring threads

As of Python 2.0, threads are enabled by default. If you wish to compile without threads, or if your thread support is broken, pass the --with-threads=no switch to configure. Unfortunately, on some platforms, additional compiler and/or linker options are required for threads to work properly. Below is a table of those options, collected by Bill Janssen. We would love to automate this process more, but the information below is not enough to write a patch for the configure.in file, so manual intervention is required. If you patch the configure.in file and are confident that the patch works, please send in the patch. (Don't bother patching the configure script itself -- it is regenerated each time the configure.in file changes.)

#### Compiler switches for threads

## Text of Third-Party Software Licenses

.....

The definition of `_REENTRANT` should be configured automatically, if that does not work on your system, or if `_REENTRANT` is defined incorrectly, please report that as a bug.

OS/Compiler/threads (POSIX is draft 10, DCE is draft 4)	Switches for use with threads compile & link
SunOS 5.{1-5}/{gcc,SunPro cc}/solaris	-mt
SunOS 5.5/{gcc,SunPro cc}/POSIX	(nothing)
DEC OSF/1 3.x/cc/DCE (butenhof@zko.dec.com)	-threads
Digital UNIX 4.x/cc/DCE (butenhof@zko.dec.com)	-threads
Digital UNIX 4.x/cc/POSIX (butenhof@zko.dec.com)	-pthread
AIX 4.1.4/cc_r/d7 (buhrt@quest.net)	(nothing)
AIX 4.1.4/cc_r4/DCE (buhrt@quest.net)	(nothing)
IRIX 6.2/cc/POSIX (robertl@cwi.nl)	(nothing)

Linker (ld) libraries and flags for threads

.....

OS/threads	Libraries/switches for use with threads
SunOS 5.{1-5}/solaris	-lthread
SunOS 5.5/POSIX	-lpthread
DEC OSF/1 3.x/DCE (butenhof@zko.dec.com)	-lpthreads -lmach -lc_r -lc
Digital UNIX 4.x/DCE (butenhof@zko.dec.com)	-lpthreads -lpthread -lmach -lexc -lc
Digital UNIX 4.x/POSIX (butenhof@zko.dec.com)	-lpthread -lmach -lexc -lc
AIX 4.1.4/{draft7,DCE} (buhrt@quest.net)	(nothing)
IRIX 6.2/POSIX (jph@emilia.engr.sgi.com)	-lpthread

Building a shared libpython

-----

Starting with Python 2.3, the majority of the interpreter can be built into a shared library, which can then be used by the interpreter executable, and by applications embedding Python. To enable this feature, configure with `--enable-shared`.

If you enable this feature, the same object files will be used to create a static library. In particular, the static library will contain object files using position-independent code (PIC) on platforms where PIC flags are needed for the shared library.

Configuring additional built-in modules

-----  
Starting with Python 2.1, the setup.py script at the top of the source distribution attempts to detect which modules can be built and automatically compiles them. Autodetection doesn't always work, so you can still customize the configuration by editing the Modules/Setup file; but this should be considered a last resort. The rest of this section only applies if you decide to edit the Modules/Setup file. You also need this to enable static linking of certain modules (which is needed to enable profiling on some systems).

This file is initially copied from Setup.dist by the configure script; if it does not exist yet, create it by copying Modules/Setup.dist yourself (configure will never overwrite it). Never edit Setup.dist -- always edit Setup or Setup.local (see below). Read the comments in the file for information on what kind of edits are allowed. When you have edited Setup in the Modules directory, the interpreter will automatically be rebuilt the next time you run make (in the toplevel directory).

Many useful modules can be built on any Unix system, but some optional modules can't be reliably autodetected. Often the quickest way to determine whether a particular module works or not is to see if it will build: enable it in Setup, then if you get compilation or link errors, disable it -- you're either missing support or need to adjust the compilation and linking parameters for that module.

On SGI IRIX, there are modules that interface to many SGI specific system libraries, e.g. the GL library and the audio hardware. These modules will not be built by the setup.py script.

In addition to the file Setup, you can also edit the file Setup.local. (the makesetup script processes both). You may find it more convenient to edit Setup.local and leave Setup alone. Then, when installing a new Python version, you can copy your old Setup.local file.

#### Setting the optimization/debugging options

-----

If you want or need to change the optimization/debugging options for the C compiler, assign to the OPT variable on the toplevel make command; e.g. "make OPT=-g" will build a debugging version of Python on most platforms. The default is OPT=-O; a value for OPT in the environment when the configure script is run overrides this default (likewise for CC; and the initial value for LIBS is used as the base set of libraries to link with).

When compiling with GCC, the default value of OPT will also include the -Wall and -Wstrict-prototypes options.

Additional debugging code to help debug memory management problems can be enabled by using the --with-pydebug option to the configure script.

For flags that change binary compatibility, use the EXTRA\_CFLAGS variable.

## Text of Third-Party Software Licenses

### Profiling

If you want C profiling turned on, the easiest way is to run configure with the CC environment variable to the necessary compiler invocation. For example, on Linux, this works for profiling using gprof(1) :

```
CC="gcc -pg" ./configure
```

Note that on Linux, gprof apparently does not work for shared libraries. The Makefile/Setup mechanism can be used to compile and link most extension modules statically.

### Testing

To test the interpreter, type "make test" in the top-level directory. This runs the test set twice (once with no compiled files, once with the compiled files left by the previous test run). The test set produces some output. You can generally ignore the messages about skipped tests due to optional features which can't be imported. If a message is printed about a failed test or a traceback or core dump is produced, something is wrong. On some Linux systems (those that are not yet using glibc 6), test.strftime fails due to a non-standard implementation of strftime() in the C library. Please ignore this, or upgrade to glibc version 6.

**IMPORTANT:** If the tests fail and you decide to mail a bug report, \*don't\* include the output of "make test". It is useless. Run the failing test manually, as follows:

```
./python ./Lib/test/test_whatever.py
```

(substituting the top of the source tree for '.' if you built in a different directory). This runs the test in verbose mode.

### Installing

To install the Python binary, library modules, shared library modules (see below), include files, configuration files, and the manual page, just type

```
make install
```

This will install all platform-independent files in subdirectories of the directory given with the --prefix option to configure or to the `prefix' Make variable (default /usr/local). All binary and other platform-specific files will be installed in subdirectories if the directory given by --exec-prefix or the `exec\_prefix' Make variable (defaults to the --prefix directory) is given.

If DESTDIR is set, it will be taken as the root directory of the installation, and files will be installed into \$(DEstdir)\$prefix, \$(DEstdir)\$exec\_prefix, etc.

All subdirectories created will have Python's version number in their name, e.g. the library modules are installed in "/usr/local/lib/python<version>/" by default, where <version> is the <major>.<minor> release number (e.g. "2.1"). The Python binary is installed as "python<version>" and a hard link named "python" is created. The only file not installed with a version number in its name is the manual page, installed as "/usr/local/man/man1/python.1" by default.

If you have a previous installation of Python that you don't want to replace yet, use

```
make altinstall
```

This installs the same set of files as "make install" except it doesn't create the hard link to "python<version>" named "python" and it doesn't install the manual page at all.

The only thing you may have to install manually is the Python mode for Emacs found in Misc/python-mode.el. (But then again, more recent versions of Emacs may already have it.) Follow the instructions that came with Emacs for installation of site-specific files.

On Mac OS X, if you have configured Python with --enable-framework, you should use "make frameworkinstall" to do the installation. Note that this installs the Python executable in a place that is not normally on your PATH, you may want to set up a symlink in /usr/local/bin.

#### Configuration options and variables

---

Some special cases are handled by passing options to the configure script.

**WARNING:** if you rerun the configure script with different options, you must run "make clean" before rebuilding. Exceptions to this rule: after changing --prefix or --exec-prefix, all you need to do is remove Modules/getpath.o.

**--with(out)-gcc:** The configure script uses gcc (the GNU C compiler) if it finds it. If you don't want this, or if this compiler is installed but broken on your platform, pass the option --without-gcc. You can also pass "CC=cc" (or whatever the name of the proper C compiler is) in the environment, but the advantage of using --without-gcc is that this option is remembered by the config.status script for its --recheck option.

**--prefix, --exec-prefix:** If you want to install the binaries and the Python library somewhere else than in /usr/local/{bin,lib}, you can pass the option --prefix=DIRECTORY; the interpreter binary will be installed as DIRECTORY/bin/python and the library files as DIRECTORY/lib/python/\*. If you pass --exec-prefix=DIRECTORY (as well) this overrides the installation prefix for architecture-dependent files (like the interpreter binary). Note that --prefix=DIRECTORY also affects the default module search path (sys.path), when Modules/config.c is compiled. Passing make the option

```
prefix=DIRECTORY (and/or exec_prefix=DIRECTORY) overrides the
prefix set at configuration time; this may be more convenient
than re-running the configure script if you change your mind
about the install prefix.

--with-readline: This option is no longer supported.  GNU
readline is automatically enabled by setup.py when present.

--with-threads: On most Unix systems, you can now use multiple
threads, and support for this is enabled by default. To
disable this, pass --with-threads=no. If the library required
for threads lives in a peculiar place, you can use
--with-thread DIRECTORY. IMPORTANT: run "make clean" after
changing (either enabling or disabling) this option, or you
will get link errors! Note: for DEC Unix use
--with-dec-threads instead.

--with-sgi-dl: On SGI IRIX 4, dynamic loading of extension modules is
supported by the "dl" library by Jack Jansen, which is
ftp'able from ftp://ftp.cwi.nl/pub/dynload/dl-1.6.tar.Z.
This is enabled (after you've ftp'ed and compiled the dl
library) by passing --with-sgi-dl= DIRECTORY where DIRECTORY
is the absolute pathname of the dl library. (Don't bother on
IRIX 5, it already has dynamic linking using SunOS style
shared libraries.) THIS OPTION IS UNSUPPORTED.

--with-dl-dld: Dynamic loading of modules is rumored to be supported
on some other systems: VAX (Ultrix), Sun3 (SunOS 3.4), Sequent
Symmetry (Dynix), and Atari ST. This is done using a
combination of the GNU dynamic loading package
(ftp://ftp.cwi.nl/pub/dynload/dl-dld-1.1.tar.Z) and an
emulation of the SGI dl library mentioned above (the emulation
can be found at
ftp://ftp.cwi.nl/pub/dynload/dld-3.2.3.tar.Z). To
enable this, ftp and compile both libraries, then call
configure, passing it the option
--with-dl-dld=DL_DIRECTORY,DLD_DIRECTORY where DL_DIRECTORY is
the absolute pathname of the dl emulation library and
DLD_DIRECTORY is the absolute pathname of the GNU dld library.
(Don't bother on SunOS 4 or 5, they already have dynamic
linking using shared libraries.) THIS OPTION IS UNSUPPORTED.

--with-libm, --with-libc: It is possible to specify alternative
versions for the Math library (default -lm) and the C library
(default the empty string) using the options
--with-libm=STRING and --with-libc=STRING, respectively. For
example, if your system requires that you pass -lc_s to the C
compiler to use the shared C library, you can pass
--with-libc=-lc_s. These libraries are passed after all other
libraries, the C library last.

--with-libs='libs': Add 'libs' to the LIBS that the python interpreter
is linked against.

--with-cxx-main=<compiler>: If you plan to use C++ extension modules,
then -- on some platforms -- you need to compile python's main()
function with the C++ compiler. With this option, make will use
<compiler> to compile main() *and* to link the python executable.
It is likely that the resulting executable depends on the C++
```

runtime library of <compiler>. (The default is --without-cxx-main.)

There are platforms that do not require you to build Python with a C++ compiler in order to use C++ extension modules. E.g., x86 Linux with ELF shared binaries and GCC 3.x, 4.x is such a platform. We recommend that you configure Python --without-cxx-main on those platforms because a mismatch between the C++ compiler version used to build Python and to build a C++ extension module is likely to cause a crash at runtime.

The Python installation also stores the variable CXX that determines, e.g., the C++ compiler distutils calls by default to build C++ extensions. If you set CXX on the configure command line to any string of non-zero length, then configure won't change CXX. If you do not preset CXX but pass --with-cxx-main=<compiler>, then configure sets CXX=<compiler>. In all other cases, configure looks for a C++ compiler by some common names (c++, g++, gcc, CC, cxx, cc++, cl) and sets CXX to the first compiler it finds. If it does not find any C++ compiler, then it sets CXX="".

Similarly, if you want to change the command used to link the python executable, then set LINKCC on the configure command line.

```
--with-pydebug: Enable additional debugging code to help track down
memory management problems. This allows printing a list of all
live objects when the interpreter terminates.

--with(out)-universal-newlines: enable reading of text files with
foreign newline convention (default: enabled). In other words,
any of \r, \n or \r\n is acceptable as end-of-line character.
If enabled import and execfile will automatically accept any newline
in files. Python code can open a file with open(file, 'U') to
read it in universal newline mode. THIS OPTION IS UNSUPPORTED.

--with-tsc: Profile using the Pentium timestamping counter (TSC).

--with-system-ffi: Build the _ctypes extension module using an ffi
library installed on the system.
```

#### Building for multiple architectures (using the VPATH feature)

---

If your file system is shared between multiple architectures, it usually is not necessary to make copies of the sources for each architecture you want to support. If the make program supports the VPATH feature, you can create an empty build directory for each architecture, and in each directory run the configure script (on the appropriate machine with the appropriate options). This creates the necessary subdirectories and the Makefiles therein. The Makefiles contain a line VPATH=... which points to a directory containing the actual sources. (On SGI systems, use "smake -J1" instead of "make" if you use VPATH -- don't try gnumake.)

For example, the following is all you need to build a minimal Python in /usr/tmp/python (assuming ~guido/src/python is the toplevel

## Text of Third-Party Software Licenses

directory and you want to build in /usr/tmp/python) :

```
$ mkdir /usr/tmp/python
$ cd /usr/tmp/python
$ ~guido/src/python/configure
[...]
$ make
[...]
$
```

Note that configure copies the original Setup file to the build directory if it finds no Setup file there. This means that you can edit the Setup file for each architecture independently. For this reason, subsequent changes to the original Setup file are not tracked automatically, as they might overwrite local changes. To force a copy of a changed original Setup file, delete the target Setup file. (The makesetup script supports multiple input files, so if you want to be fancy you can change the rules to create an empty Setup.local if it doesn't exist and run it with arguments \$(srccdir)/Setup Setup.local; however this assumes that you only need to add modules.)

## Building on non-UNIX systems

---

For Windows (2000/NT/ME/98/95), assuming you have MS VC++ 7.1, the project files are in PCbuild, the workspace is pcbuild.dsw. See PCbuild\readme.txt for detailed instructions.

For other non-Unix Windows compilers, in particular MS VC++ 6.0 and for OS/2, enter the directory "PC" and read the file "readme.txt".

For the Mac, a separate source distribution will be made available, for use with the CodeWarrior compiler. If you are interested in Mac development, join the PythonMac Special Interest Group (<http://www.python.org/sigs/pythonmac-sig/>, or send email to [pythonmac-sig-request@python.org](mailto:pythonmac-sig-request@python.org)).

Of course, there are also binary distributions available for these platforms -- see <http://www.python.org/>.

To port Python to a new non-UNIX system, you will have to fake the effect of running the configure script manually (for Mac and PC, this has already been done for you). A good start is to copy the file pyconfig.h.in to pyconfig.h and edit the latter to reflect the actual configuration of your system. Most symbols must simply be defined as 1 only if the corresponding feature is present and can be left alone otherwise; however the \*\_t type symbols must be defined as some variant of int if they need to be defined at all.

For all platforms, it's important that the build arrange to define the preprocessor symbol NDEBUG on the compiler command line in a release build of Python (else assert() calls remain in the code, hurting release-build performance). The Unix, Windows and Mac builds already do this.

## Miscellaneous issues

---

## Emacs mode

-----

There's an excellent Emacs editing mode for Python code; see the file Misc/python-mode.el. Originally written by the famous Tim Peters, it is now maintained by the equally famous Barry Warsaw (it's no coincidence that they now both work on the same team). The latest version, along with various other contributed Python-related Emacs goodies, is online at <http://www.python.org/emacs/python-mode>. And if you are planning to edit the Python C code, please pick up the latest version of CC Mode <http://www.python.org/emacs/cc-mode>; it contains a "python" style used throughout most of the Python C source files. (Newer versions of Emacs or XEmacs may already come with the latest version of python-mode.)

## Tkinter

-----

The setup.py script automatically configures this when it detects a usable Tcl/Tk installation. This requires Tcl/Tk version 8.0 or higher.

For more Tkinter information, see the Tkinter Resource page:  
<http://www.python.org/topics/tkinter/>

There are demos in the Demo/tkinter directory.

Note that there's a Python module called "Tkinter" (capital T) which lives in Lib/lib-tk/Tkinter.py, and a C module called "\_tkinter" (lower case t and leading underscore) which lives in Modules/\_tkinter.c. Demos and normal Tk applications import only the Python Tkinter module -- only the latter imports the C \_tkinter module. In order to find the C \_tkinter module, it must be compiled and linked into the Python interpreter -- the setup.py script does this. In order to find the Python Tkinter module, sys.path must be set correctly -- normal installation takes care of this.

## Distribution structure

-----

Most subdirectories have their own README files. Most files have comments.

BeOS/	Files specific to the BeOS port
Demo/	Demonstration scripts, modules and programs
Doc/	Documentation sources (LaTeX)
Grammar/	Input for the parser generator
Include/	Public header files
LICENSE	Licensing information
Lib/	Python library modules
Mac/	Macintosh specific resources
Makefile.pre.in	Source from which config.status creates the Makefile.pre
Misc/	Miscellaneous useful files
Modules/	Implementation of most built-in modules
Objects/	Implementation of most built-in object types
PC/	Files specific to PC ports (DOS, Windows, OS/2)

## Text of Third-Party Software Licenses

PCbuild/	Build directory for Microsoft Visual C++
Parser/	The parser and tokenizer and their input handling
Python/	The byte-compiler and interpreter
README	The file you're reading now
Tools/	Some useful programs written in Python
pyconfig.h.in	Source from which pyconfig.h is created (GNU autoheader output)
configure	Configuration shell script (GNU autoconf output)
configure.in	Configuration specification (input for GNU autoconf)
install-sh	Shell script used to install files
setup.py	Python script used to build extension modules

The following files will (may) be created in the toplevel directory by the configuration and build processes:

Makefile	Build rules
Makefile.pre	Build rules before running Modules/makesetup
buildno	Keeps track of the build number
config.cache	Cache of configuration variables
pyconfig.h	Configuration header
config.log	Log from last configure run
config.status	Status from last run of the configure script
getbuildinfo.o	Object file from Modules/getbuildinfo.c
libpython<version>.a	The library archive
python	The executable interpreter
tags, TAGS	Tags files for vi and Emacs

That's all, folks!

--Guido van Rossum (home page: <http://www.python.org/~guido/>)

Python Nose  
Lesser GNU Public License v2

License and copyright

nose is copyright Jason Pellerin 2005-2008

This program is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Python Logilab  
GNU Public License v2

```

# This program is free software; you can redistribute it and/or modify it under
# the terms of the GNU General Public License as published by the Free Software
# Foundation; either version 2 of the License, or (at your option) any later
# version.
#
# This program is distributed in the hope that it will be useful, but WITHOUT
# ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
# FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License along with
# this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
"""Python Abstract Syntax Tree New Generation

```

The aim of this module is to provide a common base representation of python source code for projects such as pychecker, pyreverse, pylint... Well, actually the development of this library is essentially governed by pylint's needs.

It extends class defined in the compiler.ast [1] module with some additional methods and attributes. Instance attributes are added by a builder object, which can either generate extended ast (let's call them astng;) by visiting an existant ast tree or by inspecting living object. Methods are added by monkey patching ast classes.

Main modules are:

- \* nodes and scoped\_nodes for more information about methods and attributes added to different node classes
- \* the manager contains a high level object to get astng trees from source files and living objects. It maintains a cache of previously constructed tree for quick access
- \* builder contains the class responsible to build astng trees

```

:author:    Sylvain Thenault
:copyright: 2003-2007 LOGILAB S.A. (Paris, FRANCE)
:contact:   http://www.logilab.fr/ -- mailto:python-projects@logilab.org
:copyright: 2003-2007 Sylvain Thenault
:contact:   mailto:thenault@gmail.com

```

Readline  
GNU Public License v2

RRDTool  
GNU Public License v2

RRDTOOL - Round Robin Database Tool  
A tool for fast logging of numerical data graphical display  
of this data.

Copyright (c) 1998-2008 Tobias Oetiker  
All rights reserved.

GNU GPL License

=====

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

FLOSS License Exception

=====

(Adapted from <http://www.mysql.com/company/legal/licensing/foss-exception.html>)

I want specified Free/Libre and Open Source Software ("FLOSS") applications to be able to use specified GPL-licensed RRDtool libraries (the "Program") despite the fact that not all FLOSS licenses are compatible with version 2 of the GNU General Public License (the "GPL").

As a special exception to the terms and conditions of version 2.0 of the GPL:

You are free to distribute a Derivative Work that is formed entirely from the Program and one or more works (each, a "FLOSS Work") licensed under one or more of the licenses listed below, as long as:

1. You obey the GPL in all respects for the Program and the Derivative Work, except for identifiable sections of the Derivative Work which are not derived from the Program, and which can reasonably be considered independent and separate works in themselves,
2. all identifiable sections of the Derivative Work which are not derived from the Program, and which can reasonably be considered independent and separate works in themselves,
1. are distributed subject to one of the FLOSS licenses listed below, and
2. the object code or executable form of those sections are accompanied by the complete corresponding machine-readable source code for those sections on the same medium and under the same FLOSS license as the corresponding object code or executable forms of those sections, and
3. any works which are aggregated with the Program or with a Derivative Work on a volume of a storage or distribution medium in accordance with the GPL, can reasonably be considered independent and separate works in themselves which are not derivatives of either the Program, a Derivative Work or a FLOSS Work.

If the above conditions are not met, then the Program may only be copied, modified, distributed or used under the terms and conditions of the GPL.

FLOSS License List

```
=====
License nameVersion(s) /Copyright Date
Academic Free License2.0
Apache Software License1.0/1.1/2.0
Apple Public Source License2.0
Artistic licenseFrom Perl 5.8.0
BSD license"July 22 1999"
Common Public License1.0
GNU Library or "Lesser" General Public License (LGPL) 2.0/2.1
IBM Public License, Version 1.0
Jabber Open Source License1.0
MIT License (As listed in file MIT-License.txt)-
Mozilla Public License (MPL) 1.0/1.1
Open Software License2.0
OpenSSL license (with original SSLeay license) "2003" ("1998")
PHP License3.0
Python license (CNRI Python License)-
Python Software Foundation License2.1.1
Sleepycat License"1999"
W3C License"2001"
X11 License"2001"
Zlib/libpng License-
Zope Public License2.0/2.1
```

Ruby  
GNU Public License v2

Ruby is copyrighted free software by Yukihiro Matsumoto <matz@netlab.jp>. You can redistribute it and/or modify it under either the terms of the GPL version 2 (see the file *GPL*), or the conditions below:

1. You may make and give away verbatim copies of the source form of the software without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may modify your copy of the software in any way, provided that you do at least ONE of the following:
  - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or by allowing the author to include your modifications in the software.
  - b) use the modified software only within your corporation or organization.
  - c) give non-standard binaries non-standard names, with instructions on where to get the original software distribution.
  - d) make other distribution arrangements with the author.
3. You may distribute the software in object code or binary form, provided that you do at least ONE of the following:
  - a) distribute the binaries and library files of the software, together with instructions (in the manual page or equivalent) on where to get the original distribution.

## Text of Third-Party Software Licenses

- b) accompany the distribution with the machine-readable source of the software.
  - c) give non-standard binaries non-standard names, with instructions on where to get the original software distribution.
  - d) make other distribution arrangements with the author.
4. You may modify and include the part of the software into any other software (possibly commercial). But some files in the distribution are not written by the author, so that they are not under these terms.
- For the list of those files and their copying conditions, see the file **LEGAL**.
5. The scripts and library files supplied as input to or produced as output from the software do not automatically fall under the copyright of the software, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this software.
6. THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

### Silk Icons

Create Commons 2.5

<http://www.famfamfam.com/lab/icons/silk/>

### LICENSE

I also love to hear of my work being used, feel encouraged to send an email with a link or screenshot of the icons in their new home to mjames at gmail dot com. This work is licensed under a Creative Commons Attribution 2.5 License. This means you may use it for any purpose, and make any changes you like. All I ask is that you include a link back to this page in your credits (although a giant link on every page of your website really isn't needed, contact me to discuss specifics).

The icons can also be used under Creative Commons Attribution 3.0 License (Hi Debian folks!) with the following requirements:

As an author, I would appreciate a reference to my authorship of the Silk icon set contents within a readme file or equivalent documentation for the software which includes the set or a subset of the icons contained within.

Simple XML  
Apache License v2

slf4j

Copyright (c) 2004-2008 QOS.ch  
All rights reserved.

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

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

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

Spring Framework  
Apache License v2

Spring Framework 3.1  
Copyright (c) 2002-2011 SpringSource, a division of VMware, Inc.

This product is licensed to you under the Apache License, Version 2.0 (the "License"). You may not use this product except in compliance with the License.

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE file.

String Template

StringTemplate 3.2.1  
September 22, 2009

Terence Parr, parrt at cs usfca edu  
ANTLR project lead and supreme dictator for life  
University of San Francisco

ST (StringTemplate) is a java template engine (with ports for C# and Python) for generating source code, web pages, emails, or any other formatted text output. ST is particularly good at multi-targeted code generators, multiple site skins, and internationalization/localization. It evolved over years of effort developing jGuru.com. ST also generates this website and powers the ANTLR v3 code generator. Its distinguishing characteristic is that it strictly enforces model-view separation unlike other engines.

The main website is:

<http://www.stringtemplate.org>

The documentation is in the wiki:

## Text of Third-Party Software Licenses

<http://www.antlr.org/wiki/display/ST/StringTemplate+Documentation>

Here are the 3.2.1 release notes:

<http://www.antlr.org/wiki/display/ST/3.2.1+Release+Notes>

Per the license in LICENSE.txt, this software is not guaranteed to work and might even destroy all life on this planet.

See the CHANGES.txt file.

[The "BSD licence"]  
Copyright (c) 2003-2008 Terence Parr  
All rights reserved.

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

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

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

Sun JDK

Java JDK 6 License

Sun Microsystems, Inc. Binary Code License Agreement

for the JAVA SE DEVELOPMENT KIT (JDK), VERSION 6

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE SOFTWARE IDENTIFIED BELOW TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THIS SOFTWARE, YOU ACCEPT THE TERMS OF THE AGREEMENT. INDICATE ACCEPTANCE BY SELECTING THE "ACCEPT" BUTTON AT THE BOTTOM OF THE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY ALL THE TERMS, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THE AGREEMENT AND THE DOWNLOAD OR INSTALL PROCESS WILL NOT

CONTINUE.

1. DEFINITIONS. "Software" means the identified above in binary form, any other machine readable materials (including, but not limited to, libraries, source files, header files, and data files), any updates or error corrections provided by Sun, and any user manuals, programming guides and other documentation provided to you by Sun under this Agreement. "General Purpose Desktop Computers and Servers" means computers, including desktop, laptop and tablet computers, or servers, used for general computing functions under end user control (such as but not specifically limited to email, general purpose Internet browsing, and office suite productivity tools). The use of Software in systems and solutions that provide dedicated functionality (other than as mentioned above) or designed for use in embedded or function-specific software applications, for example but not limited to: Software embedded in or bundled with industrial control systems, wireless mobile telephones, wireless handheld devices, kiosks, TV/STB, Blu-ray Disc devices, telematics and network control switching equipment, printers and storage management systems, and other related systems are excluded from this definition and not licensed under this Agreement. "Programs" means Java technology applets and applications intended to run on the Java Platform Standard Edition (Java SE) platform on Java-enabled General Purpose Desktop Computers and Servers.

2. LICENSE TO USE. Subject to the terms and conditions of this Agreement, including, but not limited to the Java Technology Restrictions of the Supplemental License Terms, Sun grants you a non-exclusive, non-transferable, limited license without license fees to reproduce and use internally Software complete and unmodified for the sole purpose of running Programs. Additional licenses for developers and/or publishers are granted in the Supplemental License Terms.

3. RESTRICTIONS. Software is confidential and copyrighted. Title to Software and all associated intellectual property rights is retained by Sun and/or its licensors. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer Software. You acknowledge that Licensed Software is not designed or intended for use in the design, construction, operation or maintenance of any nuclear facility. Sun Microsystems, Inc. disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Sun or its licensors is granted under this Agreement. Additional restrictions for developers and/or publishers licenses are set forth in the Supplemental License Terms.

4. LIMITED WARRANTY. Sun warrants to you that for a period of ninety (90) days from the date of purchase, as evidenced by a copy of the receipt, the media on which Software is furnished (if any) will be free of defects in materials and workmanship under normal use. Except for the foregoing,

## Text of Third-Party Software Licenses

Software is provided "AS IS". Your exclusive remedy and Sun's entire liability under this limited warranty will be at Sun's option to replace Software media or refund the fee paid for Software. Any implied warranties on the Software are limited to 90 days. Some states do not allow limitations on duration of an implied warranty, so the above may not apply to you. This limited warranty gives you specific legal rights. You may have others, which vary from state to state.

5. DISCLAIMER OF WARRANTY. UNLESS SPECIFIED IN THIS AGREEMENT, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

6. LIMITATION OF LIABILITY. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event will Sun's liability to you, whether in contract, tort (including negligence), or otherwise, exceed the amount paid by you for Software under this Agreement. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose. Some states do not allow the exclusion of incidental or consequential damages, so some of the terms above may not be applicable to you.

7. TERMINATION. This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate immediately without notice from Sun if you fail to comply with any provision of this Agreement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right. Upon Termination, you must destroy all copies of Software.

8. EXPORT REGULATIONS. All Software and technical data delivered under this Agreement are subject to US export control laws and may be subject to export or import regulations in other countries. You agree to comply strictly with all such laws and regulations and acknowledge that you have the responsibility to obtain such licenses to export, re-export, or import as may be required after delivery to you.

9. TRADEMARKS AND LOGOS. You acknowledge and agree as between you and Sun that Sun owns the SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET trademarks and all SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET-related trademarks, service marks, logos and other brand designations ("Sun Marks"),

and you agree to comply with the Sun Trademark and Logo Usage Requirements currently located at <http://www.sun.com/policies/trademarks>. Any use you make of the Sun Marks inures to Sun's benefit.

10. U.S. GOVERNMENT RESTRICTED RIGHTS. If Software is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in Software and accompanying documentation will be only as set forth in this Agreement; this is in accordance with 48 CFR 227.7201 through 227.7202-4 (for Department of Defense (DOD) acquisitions) and with 48 CFR 2.101 and 12.212 (for non-DOD acquisitions).

11. GOVERNING LAW. Any action related to this Agreement will be governed by California law and controlling U.S. federal law. No choice of law rules of any jurisdiction will apply.

12. SEVERABILITY. If any provision of this Agreement is held to be unenforceable, this Agreement will remain in effect with the provision omitted, unless omission would frustrate the intent of the parties, in which case this Agreement will immediately terminate.

13. INTEGRATION. This Agreement is the entire agreement between you and Sun relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

#### SUPPLEMENTAL LICENSE TERMS

These Supplemental License Terms add to or modify the terms of the Binary Code License Agreement. Capitalized terms not defined in these Supplemental Terms shall have the same meanings ascribed to them in the Binary Code License Agreement. These Supplemental Terms shall supersede any inconsistent or conflicting terms in the Binary Code License Agreement, or in any license contained within the Software.

A. Software Internal Use and Development License Grant. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the Software README file incorporated herein by reference, including, but not limited to the Java Technology Restrictions of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license without fees to reproduce internally and use internally the Software complete and unmodified for the purpose of designing, developing, and testing your Programs.

## Text of Third-Party Software Licenses

B. License to Distribute Software. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the Software README file, including, but not limited to the Java Technology Restrictions of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license without fees to reproduce and distribute the Software, provided that (i) you distribute the Software complete and unmodified and only bundled as part of, and for the sole purpose of running, your Programs, (ii) the Programs add significant and primary functionality to the Software, (iii) you do not distribute additional software intended to replace any component(s) of the Software, (iv) you do not remove or alter any proprietary legends or notices contained in the Software, (v) you only distribute the Software subject to a license agreement that protects Sun's interests consistent with the terms contained in this Agreement, and (vi) you agree to defend and indemnify Sun and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software.

C. License to Distribute Redistributables. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the Software README file, including but not limited to the Java Technology Restrictions of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license without fees to reproduce and distribute those files specifically identified as redistributable in the Software "README" file ("Redistributables") provided that: (i) you distribute the Redistributables complete and unmodified, and only bundled as part of Programs, (ii) the Programs add significant and primary functionality to the Redistributables, (iii) you do not distribute additional software intended to supersede any component(s) of the Redistributables (unless otherwise specified in the applicable README file), (iv) you do not remove or alter any proprietary legends or notices contained in or on the Redistributables, (v) you only distribute the Redistributables pursuant to a license agreement that protects Sun's interests consistent with the terms contained in the Agreement, (vi) you agree to defend and indemnify Sun and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software.

D. Java Technology Restrictions. You may not create, modify, or change the behavior of, or authorize your licensees to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Sun in any naming convention designation.

E. Distribution by Publishers. This section pertains to your distribution of the Software with your printed book or magazine (as those terms are commonly used in the industry) relating to Java technology ("Publication"). Subject to and conditioned upon your compliance with the restrictions and obligations contained in the Agreement, in addition to the license granted in Paragraph 1 above, Sun hereby grants to you a non-exclusive, nontransferable limited right to reproduce complete and unmodified copies of the Software on electronic media (the "Media") for the sole purpose of inclusion and distribution with your Publication(s), subject to the following terms: (i) You may not distribute the Software on a stand-alone basis; it must be distributed with your Publication(s); (ii) You are responsible for downloading the Software from the applicable Sun web site; (iii) You must refer to the Software as JavaTM SE Development Kit 6; (iv) The Software must be reproduced in its entirety and without any modification whatsoever (including, without limitation, the Binary Code License and Supplemental License Terms accompanying the Software and proprietary rights notices contained in the Software); (v) The Media label shall include the following information: Copyright 2006, Sun Microsystems, Inc. All rights reserved. Use is subject to license terms. Sun, Sun Microsystems, the Sun logo, Solaris, Java, the Java Coffee Cup logo, J2SE, and all trademarks and logos based on Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. This information must be placed on the Media label in such a manner as to only apply to the Sun Software; (vi) You must clearly identify the Software as Sun's product on the Media holder or Media label, and you may not state or imply that Sun is responsible for any third-party software contained on the Media; (vii) You may not include any third party software on the Media which is intended to be a replacement or substitute for the Software; (viii) You shall indemnify Sun for all damages arising from your failure to comply with the requirements of this Agreement. In addition, you shall defend, at your expense, any and all claims brought against Sun by third parties, and shall pay all damages awarded by a court of competent jurisdiction, or such settlement amount negotiated by you, arising out of or in connection with your use, reproduction or distribution of the Software and/or the Publication. Your obligation to provide indemnification under this section shall arise provided that Sun: (a) provides you prompt notice of the claim; (b) gives you sole control of the defense and settlement of the claim; (c) provides you, at your expense, with all available information, assistance and authority to defend; and (d) has not compromised or settled such claim without your prior written consent; and (ix) You shall provide Sun with a written notice for each Publication; such notice shall include the following information: (1) title of Publication, (2) author(s), (3) date of Publication, and (4) ISBN or ISSN numbers. Such notice shall be sent to Sun Microsystems, Inc., 4150 Network Circle, M/S USCA12-110, Santa Clara, California 95054, U.S.A , Attention: Contracts Administration.

## Text of Third-Party Software Licenses

F. Source Code. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of this Agreement. Source code may not be redistributed unless expressly provided for in this Agreement.

G. Third Party Code. Additional copyright notices and license terms applicable to portions of the Software are set forth in the THIRDPARTYLICENSEREADME.txt file. In addition to any terms and conditions of any third party opensource/freeware license identified in the THIRDPARTYLICENSEREADME.txt file, the disclaimer of warranty and limitation of liability provisions in paragraphs 5 and 6 of the Binary Code License Agreement shall apply to all Software in this distribution.

H. Termination for Infringement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right.

I. Installation and Auto-Update. The Software's installation and auto-update processes transmit a limited amount of data to Sun (or its service provider) about those specific processes to help Sun understand and optimize them. Sun does not associate the data with personally identifiable information. You can find more information about the data Sun collects at <http://java.com/data/>.

For inquiries please contact: Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A.

### Java JDK 5 License

#### Sun Microsystems, Inc. Binary Code License Agreement

for the JAVA 2 PLATFORM STANDARD EDITION DEVELOPMENT KIT 5.0

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE SOFTWARE IDENTIFIED BELOW TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THIS SOFTWARE, YOU ACCEPT THE TERMS OF THE AGREEMENT. INDICATE ACCEPTANCE BY SELECTING THE "ACCEPT" BUTTON AT THE BOTTOM OF THE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY ALL THE TERMS, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THE AGREEMENT AND THE DOWNLOAD OR INSTALL PROCESS WILL NOT CONTINUE.

1. DEFINITIONS. "Software" means the identified above in binary form, any other machine readable materials (including, but not limited to, libraries, source files, header files, and data files), any updates or error corrections provided by Sun, and any user manuals, programming guides and other documentation provided to you by Sun under this Agreement. "General Purpose Desktop

Computers and Servers" means computers, including desktop, laptop and tablet computers, or servers, used for general computing functions under end user control (such as but not specifically limited to email, general purpose Internet browsing, and office suite productivity tools). The use of Software in systems and solutions that provide dedicated functionality (other than as mentioned above) or designed for use in embedded or function-specific software applications, for example but not limited to: Software embedded in or bundled with industrial control systems, wireless mobile telephones, wireless handheld devices, kiosks, TV/STB, Blu-ray Disc devices, telematics and network control switching equipment, printers and storage management systems, and other related systems are excluded from this definition and not licensed under this Agreement. Programs means Java technology applets and applications intended to run on the Java Platform Standard Edition (Java SE) platform on Java-enabled General Purpose Desktop Computers and Servers.

2. LICENSE TO USE. Subject to the terms and conditions of this Agreement, including, but not limited to the Java Technology Restrictions of the Supplemental License Terms, Sun grants you a non-exclusive, non-transferable, limited license without license fees to reproduce and use internally Software complete and unmodified for the sole purpose of running Programs. Additional licenses for developers and/or publishers are granted in the Supplemental License Terms.

3. RESTRICTIONS. Software is confidential and copyrighted. Title to Software and all associated intellectual property rights is retained by Sun and/or its licensors. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer Software. You acknowledge that Licensed Software is not designed or intended for use in the design, construction, operation or maintenance of any nuclear facility. Sun Microsystems, Inc. disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Sun or its licensors is granted under this Agreement. Additional restrictions for developers and/or publishers licenses are set forth in the Supplemental License Terms.

4. LIMITED WARRANTY. Sun warrants to you that for a period of ninety (90) days from the date of purchase, as evidenced by a copy of the receipt, the media on which Software is furnished (if any) will be free of defects in materials and workmanship under normal use. Except for the foregoing, Software is provided "AS IS". Your exclusive remedy and Sun's entire liability under this limited warranty will be at Sun's option to replace Software media or refund the fee paid for Software. Any implied warranties on the Software are limited to 90 days. Some states do not allow limitations on duration of an implied warranty, so the above may not apply to you. This limited warranty gives you specific legal rights. You may have others, which vary from state to state.

## Text of Third-Party Software Licenses

5. DISCLAIMER OF WARRANTY. UNLESS SPECIFIED IN THIS AGREEMENT, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

6. LIMITATION OF LIABILITY. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event will Sun's liability to you, whether in contract, tort (including negligence), or otherwise, exceed the amount paid by you for Software under this Agreement. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose. Some states do not allow the exclusion of incidental or consequential damages, so some of the terms above may not be applicable to you.

7. TERMINATION. This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate immediately without notice from Sun if you fail to comply with any provision of this Agreement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right. Upon Termination, you must destroy all copies of Software.

8. EXPORT REGULATIONS. All Software and technical data delivered under this Agreement are subject to US export control laws and may be subject to export or import regulations in other countries. You agree to comply strictly with all such laws and regulations and acknowledge that you have the responsibility to obtain such licenses to export, re-export, or import as may be required after delivery to you.

9. TRADEMARKS AND LOGOS. You acknowledge and agree as between you and Sun that Sun owns the SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET trademarks and all SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET-related trademarks, service marks, logos and other brand designations ("Sun Marks"), and you agree to comply with the Sun Trademark and Logo Usage Requirements currently located at <http://www.sun.com/policies/trademarks>. Any use you make of the Sun Marks inures to Sun's benefit.

10. U.S. GOVERNMENT RESTRICTED RIGHTS. If Software is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in Software and accompanying documentation will be only as set forth in this

Agreement; this is in accordance with 48 CFR 227.7201 through 227.7202-4 (for Department of Defense (DOD) acquisitions) and with 48 CFR 2.101 and 12.212 (for non-DOD acquisitions).

11. GOVERNING LAW. Any action related to this Agreement will be governed by California law and controlling U.S. federal law. No choice of law rules of any jurisdiction will apply.

12. SEVERABILITY. If any provision of this Agreement is held to be unenforceable, this Agreement will remain in effect with the provision omitted, unless omission would frustrate the intent of the parties, in which case this Agreement will immediately terminate.

13. INTEGRATION. This Agreement is the entire agreement between you and Sun relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

#### SUPPLEMENTAL LICENSE TERMS

These Supplemental License Terms add to or modify the terms of the Binary Code License Agreement. Capitalized terms not defined in these Supplemental Terms shall have the same meanings ascribed to them in the Binary Code License Agreement. These Supplemental Terms shall supersede any inconsistent or conflicting terms in the Binary Code License Agreement, or in any license contained within the Software.

A. Software Internal Use and Development License Grant. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the Software README file incorporated herein by reference, including, but not limited to the Java Technology Restrictions of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license without fees to reproduce internally and use internally the Software complete and unmodified for the purpose of designing, developing, and testing your Programs.

B. License to Distribute Software. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the Software README file, including, but not limited to the Java Technology Restrictions of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license without fees to reproduce and distribute the Software, provided that (i) you distribute the Software complete and unmodified and only bundled as part of, and for the sole purpose of running, your Programs, (ii) the Programs add significant and primary functionality to the Software, (iii) you do not distribute

## Text of Third-Party Software Licenses

additional software intended to replace any component(s) of the Software, (iv) you do not remove or alter any proprietary legends or notices contained in the Software, (v) you only distribute the Software subject to a license agreement that protects Sun's interests consistent with the terms contained in this Agreement, and (vi) you agree to defend and indemnify Sun and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software.

C. License to Distribute Redistributables. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the Software README file, including but not limited to the Java Technology Restrictions of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license without fees to reproduce and distribute those files specifically identified as redistributable in the Software "README" file ("Redistributables") provided that: (i) you distribute the Redistributables complete and unmodified, and only bundled as part of Programs, (ii) the Programs add significant and primary functionality to the Redistributables, (iii) you do not distribute additional software intended to supersede any component(s) of the Redistributables (unless otherwise specified in the applicable README file), (iv) you do not remove or alter any proprietary legends or notices contained in or on the Redistributables, (v) you only distribute the Redistributables pursuant to a license agreement that protects Sun's interests consistent with the terms contained in the Agreement, (vi) you agree to defend and indemnify Sun and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software.

D. Java Technology Restrictions. You may not create, modify, or change the behavior of, or authorize your licensees to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Sun in any naming convention designation.

E. Distribution by Publishers. This section pertains to your distribution of the Software with your printed book or magazine (as those terms are commonly used in the industry) relating to Java technology ("Publication"). Subject to and conditioned upon your compliance with the restrictions and obligations contained in the Agreement, in addition to the license granted in Paragraph 1 above, Sun hereby grants to you a non-exclusive, nontransferable limited right to reproduce complete and unmodified copies of the Software on electronic media (the "Media") for the sole purpose of inclusion and distribution with your Publication(s), subject to the following terms: (i) You may not distribute the

Software on a stand-alone basis; it must be distributed with your Publication(s); (ii) You are responsible for downloading the Software from the applicable Sun web site; (iii) You must refer to the Software as JavaTM 2 Platform Standard Edition Development Kit 5.0; (iv) The Software must be reproduced in its entirety and without any modification whatsoever (including, without limitation, the Binary Code License and Supplemental License Terms accompanying the Software and proprietary rights notices contained in the Software); (v) The Media label shall include the following information: Copyright 2006, Sun Microsystems, Inc. All rights reserved. Use is subject to license terms. Sun, Sun Microsystems, the Sun logo, Solaris, Java, the Java Coffee Cup logo, J2SE, and all trademarks and logos based on Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. This information must be placed on the Media label in such a manner as to only apply to the Sun Software; (vi) You must clearly identify the Software as Sun's product on the Media holder or Media label, and you may not state or imply that Sun is responsible for any third-party software contained on the Media; (vii) You may not include any third party software on the Media which is intended to be a replacement or substitute for the Software; (viii) You shall indemnify Sun for all damages arising from your failure to comply with the requirements of this Agreement. In addition, you shall defend, at your expense, any and all claims brought against Sun by third parties, and shall pay all damages awarded by a court of competent jurisdiction, or such settlement amount negotiated by you, arising out of or in connection with your use, reproduction or distribution of the Software and/or the Publication. Your obligation to provide indemnification under this section shall arise provided that Sun: (a) provides you prompt notice of the claim; (b) gives you sole control of the defense and settlement of the claim; (c) provides you, at your expense, with all available information, assistance and authority to defend; and (d) has not compromised or settled such claim without your prior written consent; and (ix) You shall provide Sun with a written notice for each Publication; such notice shall include the following information: (1) title of Publication, (2) author(s), (3) date of Publication, and (4) ISBN or ISSN numbers. Such notice shall be sent to Sun Microsystems, Inc., 4150 Network Circle, M/S USCA12-110, Santa Clara, California 95054, U.S.A , Attention: Contracts Administration.

F. Source Code. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of this Agreement. Source code may not be redistributed unless expressly provided for in this Agreement.

G. Third Party Code. Additional copyright notices and license terms applicable to portions of the Software are set forth in the THIRDPARTYLICENSEREADME.txt file. In addition to any terms and conditions of any third party opensource/freeware license identified in the THIRDPARTYLICENSEREADME.txt file, the disclaimer of warranty

## Text of Third-Party Software Licenses

and limitation of liability provisions in paragraphs 5 and 6 of the Binary Code License Agreement shall apply to all Software in this distribution.

H. Termination for Infringement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right.

I. Installation and Auto-Update. The Software's installation and auto-update processes transmit a limited amount of data to Sun (or its service provider) about those specific processes to help Sun understand and optimize them. Sun does not associate the data with personally identifiable information. You can find more information about the data Sun collects at <http://java.com/data/>.

For inquiries please contact: Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A.  
(LFI#143333/Form ID#011801)

DO NOT TRANSLATE OR LOCALIZE.

%% The following software may be included in this product: CS CodeViewer v1.0;  
Use of any of this software is governed by the terms of the license below:  
Copyright 1999 by CoolServlets.com.

Any errors or suggested improvements to this class can be reported as instructed on CoolServlets.com. We hope you enjoy this program... your comments will encourage further development! This software is distributed under the terms of the BSD License. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

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

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

%% The following software may be included in this product: Crimson v1.1.1 ; Use of any of this software is governed by the terms of the license below:

```

/*
 * The Apache Software License, Version 1.1
 *
 *
 * Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 * if any, must include the following acknowledgment:
 * "This product includes software developed by the
 * Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgment may appear in the software itself,
 * if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Crimson" and "Apache Software Foundation" must
 * not be used to endorse or promote products derived from this
 * software without prior written permission. For written
 * permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 * nor may "Apache" appear in their name, without prior written
 * permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * =====
 * This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation and was
 * originally based on software copyright (c) 1999, International
 * Business Machines, Inc., http://www.ibm.com. For more
 * information on the Apache Software Foundation, please see
 * <http://www.apache.org/>.
 */

```

%% The following software may be included in this product: Xalan J2; Use of any of this software is governed by the terms of the license below:

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any

## Text of Third-Party Software Licenses

additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licenser regarding such Contributions.

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

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

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

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

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

%% The following software may be included in this product: NSIS 1.0j; Use of any of this software is governed by the terms of the license below:  
Copyright (C) 1999-2000 Nullsoft, Inc.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.  
Justin Frankel justin@nullsoft.com"

%% Some Portions licensed from IBM are available at:  
<http://www.ibm.com/software/globalization/icu/>

%% Portions Copyright Eastman Kodak Company 1992

%% Lucida is a registered trademark or trademark of Bigelow & Holmes in the U.S. and other countries.

%% Portions licensed from Telligent, Inc.

%% The following software may be included in this product: IAIK PKCS Wrapper; Use of any of this software is governed by the terms of the license below:

Copyright (c) 2002 Graz University of Technology. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by IAIK of Graz University of Technology."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

## Text of Third-Party Software Licenses

4. The names "Graz University of Technology" and "IAIK of Graz University of Technology" must not be used to endorse or promote products derived from this software without prior written permission.

5. Products derived from this software may not be called "IAIK PKCS Wrapper", nor may "IAIK" appear in their name, without prior written permission of Graz University of Technology.

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

%% The following software may be included in this product: Document Object Model (DOM) v. Level 3; Use of any of this software is governed by the terms of the license below:

### W3C SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

- 1.The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- 2.Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- 3.Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission.

Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

---

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

%% The following software may be included in this product: Xalan, Xerces; Use of any of this software is governed by the terms of the license below: /\*

```
* The Apache Software License, Version 1.1
*
*
* Copyright (c) 1999-2003 The Apache Software Foundation. All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. The end-user documentation included with the redistribution,
*    if any, must include the following acknowledgment:
*      "This product includes software developed by the
*      Apache Software Foundation (http://www.apache.org/)."
*    Alternately, this acknowledgment may appear in the software itself,
*    if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Xerces" and "Apache Software Foundation" must
*    not be used to endorse or promote products derived from this
*    software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*    nor may "Apache" appear in their name, without prior written
*    permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
```

## Text of Third-Party Software Licenses

```
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* =====
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., http://www.ibm.com. For more
* information on the Apache Software Foundation, please see http://www.apache.org
*
```

%% The following software may be included in this product: W3C XML Conformance Test Suites v. 20020606; Use of any of this software is governed by the terms of the license below:

### W3C SOFTWARE NOTICE AND LICENSE

Copyright 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.  
<http://www.w3.org/Consortium/Legal/>

This W3C work (including software, documents, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code: "Copyright [\$date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>"
3. Notice of any changes or modifications to the W3C files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on August 14 1998 so as to improve compatibility with GPL. This version ensures that W3C software licensing terms are no more restrictive than GPL and consequently W3C software may be distributed in GPL packages. See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

%% The following software may be included in this product: W3C XML Schema Test Collection v. 1.16.2; Use of any of this software is governed by the terms of the license below: W3C DOCUMENT NOTICE AND LICENSE

Copyright 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.  
<http://www.w3.org/Consortium/Legal/>

Public documents on the W3C site are provided by the copyright holders under the following license. The software or Document Type Definitions (DTDs) associated with W3C specifications are governed by the Software Notice. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice of the form: "Copyright [\$date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>" (Hypertext is preferred, but atextual representation is permitted.)
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or

## Text of Third-Party Software Licenses

derivatives is sometimes granted by the W3C to individuals complying with those requirements. THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

---

This formulation of W3C's notice and license became active on April 05 1999 so as to account for the treatment of DTDs, schema's and bindings. See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to [site-policy@w3.org](mailto:site-policy@w3.org). webmaster (last updated by reagle on 1999/04/99.)

%% The following software may be included in this product: Mesa 3-D graphics library v. 5; Use of any of this software is governed by the terms of the license below:

```
core Mesa code    include/GL/gl.h      Brian Paul
Mesa GLX driver   include/GL/glx.h     Brian Paul
Mesa Ext registry include/GL/glext.h   SGI
SGI Free B        include/GL/glxext.h
```

Mesa license:

The Mesa distribution consists of several components. Different copyrights and licenses apply to different components. For example, GLUT is copyrighted by Mark Kilgard, some demo programs are copyrighted by SGI, some of the Mesa device drivers are copyrighted by their authors. See below for a list of Mesa's components and the copyright/license for each.

The core Mesa library is licensed according to the terms of the XFree86copyright (an MIT-style license). This allows integration with the XFree86/DRIproject. Unless otherwise stated, the Mesa source code and documentation is licensed as follows:

Copyright (C) 1999-2003 Brian Paul All Rights Reserved.

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

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

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

SGI FREE SOFTWARE LICENSE B (Version 1.1 [02/22/2000])

1. Definitions.

1.1 "Additional Notice Provisions" means such additional provisions as appear in the Notice in Original Code under the heading "Additional Notice Provisions."

1.2 "Covered Code" means the Original Code or Modifications, or any combination thereof.

1.3 "Hardware" means any physical device that accepts input, processes input, stores the results of processing, and/or provides output.

1.4 "Larger Work" means a work that combines Covered Code or portions thereof with code not governed by the terms of this License.

1.5 "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.6 "License" means this document.

1.7 "Licensed Patents" means patent claims Licensable by SGI that are infringed by the use or sale of Original Code or any Modifications provided by SGI, or any combination thereof.

1.8 "Modifications" means any addition to or deletion from the substance or structure of the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is: A. Any addition to the contents of a file containing Original Code and/or addition to or deletion from the contents of a file containing previous Modifications. B. Any new file that contains any part of the Original Code or previous Modifications.

1.9 "Notice" means any notice in Original Code or Covered Code, as required by and in compliance with this License.

1.10 "Original Code" means source code of computer software code that is described in the source code Notice required by Exhibit A as Original Code, and updates and error corrections specifically thereto.

1.11 "Recipient" means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 8. For legal entities, "Recipient" includes any entity that controls, is controlled by, or is under common control with Recipient. For purposes of this definition, "control" of an entity means (a) the power, direct or indirect, to direct or manage such entity, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

1.12 "Recipient Patents" means patent claims Licensable by a Recipient that are

infringed by the use or sale of Original Code or any Modifications provided by SGI, or any combination thereof.

1.13 "SGI" means Silicon Graphics, Inc.

1.14 "SGI Patents" means patent claims Licensable by SGI other than the Licensed Patents.

2. License Grant and Restrictions.

2.1 SGI License Grant. Subject to the terms of this License and any third party intellectual property claims, for the duration of intellectual property protections inherent in the Original Code, SGI hereby grants Recipient a worldwide, royalty-free, non-exclusive license, to do the following: (i) under copyrights Licensable by SGI, to reproduce, distribute, create derivative works from, and, to the extent applicable, display and perform the Original Code and/or any Modifications provided by SGI alone and/or as part of a Larger Work; and (ii) under any Licensable Patents, to make, have made, use, sell, offer for sale, import and/or otherwise transfer the Original Code and/or any Modifications provided by SGI. Recipient accepts the terms and conditions of this License by undertaking any of the aforementioned actions. The patent license shall apply to the Covered Code if, at the time any related Modification is added, such addition of the Modification causes such combination to be covered by the Licensed Patents . The patent license in Section 2.1(ii) shall not apply to any other combinations that include the Modification. No patent license is provided under SGI Patents for infringements of SGI Patents by Modifications not provided by SGI or combinations of Original Code and Modifications not provided by SGI.

2.2 Recipient License Grant. Subject to the terms of this License and any third party intellectual property claims, Recipient hereby grants SGI and any other Recipients a worldwide, royalty-free, non-exclusive license, under any Recipient Patents, to make, have made, use, sell, offer for sale, import and/or otherwise transfer the Original Code and/or any Modifications provided by SGI.

2.3 No License For Hardware Implementations. The licenses granted in Section 2.1 and 2.2 are not applicable to implementation in Hardware of the algorithms embodied in the Original Code or any Modifications provided by SGI .

3. Redistributions.

3.1 Retention of Notice/Copy of License. The Notice set forth in Exhibit A, below, must be conspicuously retained or included in any and all redistributions of Covered Code. For distributions of the Covered Code in source code form, the Notice must appear in every file that can include a text comments field; in executable form, the Notice and a copy of this License must appear in related documentation or collateral where the Recipient's rights relating to Covered Code are described. Any Additional Notice Provisions which actually appears in the Original Code must also be retained or included in any and all redistributions of Covered Code.

3.2 Alternative License. Provided that Recipient is in compliance with the terms of this License, Recipient may, so long as without derogation of any of SGI's rights in and to the Original Code, distribute the source code and/or executable version(s) of Covered Code under (1) this License; (2) a license identical to this License but for only such changes as are necessary in order to clarify Recipient's role as licensor of Modifications; and/or (3) a license of Recipient's choosing, containing terms different from this License, provided that the license terms include this Section 3 and Sections 4, 6, 7, 10, 12, and

13, which terms may not be modified or superseded by any other terms of such license. If Recipient elects to use any license other than this License, Recipient must make it absolutely clear that any of its terms which differ from this License are offered by Recipient alone, and not by SGI. It is emphasized that this License is a limited license, and, regardless of the license form employed by Recipient in accordance with this Section 3.2, Recipient may relicense only such rights, in Original Code and Modifications by SGI, as it has actually been granted by SGI in this License.

**3.3 Indemnity.** Recipient hereby agrees to indemnify SGI for any liability incurred by SGI as a result of any such alternative license terms Recipient offers.

**4. Termination.** This License and the rights granted hereunder will terminate automatically if Recipient breaches any term herein and fails to cure such breach within 30 days thereof. Any sublicense to the Covered Code that is properly granted shall survive any termination of this License, absent termination by the terms of such sublicense. Provisions that, by their nature, must remain in effect beyond the termination of this License, shall survive.

**5. No Trademark Or Other Rights.** This License does not grant any rights to: (i) any software apart from the Covered Code, nor shall any other rights or licenses not expressly granted hereunder arise by implication, estoppel or otherwise with respect to the Covered Code; (ii) any trade name, trademark or service mark whatsoever, including without limitation any related right for purposes of endorsement or promotion of products derived from the Covered Code, without prior written permission of SGI; or (iii) any title to or ownership of the Original Code, which shall at all times remains with SGI. All rights in the Original Code not expressly granted under this License are reserved.

**6. Compliance with Laws; Non-Infringement.** There are various worldwide laws, regulations, and executive orders applicable to dispositions of Covered Code, including without limitation export, re-export, and import control laws, regulations, and executive orders, of the U.S. government and other countries, and Recipient is reminded it is obliged to obey such laws, regulations, and executive orders. Recipient may not distribute Covered Code that (i) in any way infringes (directly or contributorily) any intellectual property rights of any kind of any other person or entity or (ii) breaches any representation or warranty, express, implied or statutory, to which, under any applicable law, it might be deemed to have been subject.

**7. Claims of Infringement.** If Recipient learns of any third party claim that any disposition of Covered Code and/or functionality wholly or partially infringes the third party's intellectual property rights, Recipient will promptly notify SGI of such claim.

**8. Versions of the License.** SGI may publish revised and/or new versions of the License from time to time, each with a distinguishing version number. Once Covered Code has been published under a particular version of the License, Recipient may, for the duration of the license, continue to use it under the terms of that version, or choose to use such Covered Code under the terms of any subsequent version published by SGI. Subject to the provisions of Sections 3 and 4 of this License, only SGI may modify the terms applicable to Covered Code created under this License.

**9. DISCLAIMER OF WARRANTY.** COVERED CODE IS PROVIDED "AS IS." ALL EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS ARE DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. SGI ASSUMES NO RISK AS

## Text of Third-Party Software Licenses

TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE. SHOULD THE SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, SGI ASSUMES NO COST OR LIABILITY FOR SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY IS AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT SUBJECT TO THIS DISCLAIMER.

10. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES NOR LEGAL THEORY, WHETHER TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE OR STRICT LIABILITY), CONTRACT, OR OTHERWISE, SHALL SGI OR ANY SGI LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, LOSS OF DATA, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SGI's NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THAT EXCLUSION AND LIMITATION MAY NOT APPLY TO RECIPIENT.

11. Indemnity. Recipient shall be solely responsible for damages arising, directly or indirectly, out of its utilization of rights under this License. Recipient will defend, indemnify and hold harmless Silicon Graphics, Inc. from and against any loss, liability, damages, costs or expenses (including the payment of reasonable attorneys fees) arising out of Recipient's use, modification, reproduction and distribution of the Covered Code or out of any representation or warranty made by Recipient.

12. U.S. Government End Users. The Covered Code is a "commercial item" consisting of "commercial computer software" as such terms are defined in title 48 of the Code of Federal Regulations and all U.S. Government End Users acquire only the rights set forth in this License and are subject to the terms of this License.

13. Miscellaneous. This License represents the complete agreement concerning the its subject matter. If any provision of this License is held to be unenforceable, such provision shall be reformed so as to achieve as nearly as possible the same legal and economic effect as the original provision and the remainder of this License will remain in effect. This License shall be governed by and construed in accordance with the laws of the United States and the State of California as applied to agreements entered into and to be performed entirely within California between California residents. Any litigation relating to this License shall be subject to the exclusive jurisdiction of the Federal Courts of the Northern District of California (or, absent subject matter jurisdiction in such courts, the courts of the State of California), with venue lying exclusively in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys fees and ex penses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation that provides that the language of a contract shall be construed against the drafter shall not apply to this License.

Exhibit A License Applicability. Except to the extent portions of this file are made subject to an alternative license as permitted in the SGI Free Software License B, Version 1.1 (the "License"), the contents of this file are subject only to the provisions of the License. You may not use this file except in compliance with the License. You may obtain a copy of the License at Silicon Graphics, Inc., attn: Legal Services, 1600 Amphitheatre Parkway, Mountain View, CA 94043-1351, or at: <http://oss.sgi.com/projects/FreeB>. Note that, as provided in the License, the Software is distributed on an "AS IS" basis, with ALL

EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.

Original Code. The Original Code is: [name of software, version number, and release date], developed by Silicon Graphics, Inc. The Original Code is Copyright (c) [dates of first publication, as appearing in the Notice in the Original Code] Silicon Graphics, Inc. Copyright in any portions created by third parties is as indicated elsewhere herein. All Rights Reserved.

Additional Notice Provisions: [such additional provisions, if any, as appear in the Notice in the Original Code under the heading "Additional Notice Provisions"]

% The following software may be included in this product: Byte Code Engineering Library (BCEL) v. 5; Use of any of this software is governed by the terms of the license below:

#### Apache Software License

/

---

#### The Apache Software License, Version 1.1

Copyright (c) 2001 The Apache Software Foundation. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials providedwith the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).". Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Apache" and "Apache Software Foundation"and "Apache BCEL" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [apache@apache.org](mailto:apache@apache.org).
5. Products derived from this software may not be called"Apache", "Apache BCEL", nor may "Apache" appear in their name,without prior written permission of the Apache Software Foundation.

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

=====

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

% The following software may be included in this product: Regexp, Regular Expression Package v. 1.2; Use of any of this software is governed by the terms of the license below: The Apache Software License, Version 1.1

Copyright (c) 2001 The Apache Software Foundation. All rights reserved.

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

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

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

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).". Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" and "Apache Turbine" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [apache@apache.org](mailto:apache@apache.org).

5. Products derived from this software may not be called "Apache", "Apache Turbine", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

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

=====

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

% The following software may be included in this product: CUP Parser Generator for Java v. 0.10k; Use of any of this software is governed by the terms of the license below: CUP Parser Generator Copyright Notice, License, and Disclaimer

Copyright 1996-1999 by Scott Hudson, Frank Flannery, C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

% The following software may be included in this product: JLex: A Lexical Analyzer Generator for Java v. 1.2.5; Use of any of this software is governed by the terms of the license below: JLEX COPYRIGHT NOTICE, LICENSE AND DISCLAIMER.

Copyright 1996-2003 by Elliot Joel Berk and C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Java is a trademark of Oracle Corporation. References to the Java programming language in relation to JLex are not meant to imply that Oracle endorses this product.

% The following software may be included in this product: SAX v. 2.0.1; Use of any of this software is governed by the terms of the license below:  
Copyright Status

SAX is free!

In fact, it's not possible to own a license to SAX, since it's been placed in the public domain.

No Warranty

Because SAX is released to the public domain, there is no warranty for the design or for the software implementation, to the extent permitted by applicable

## Text of Third-Party Software Licenses

law. Except when otherwise stated in writing the copyright holders and/or other parties provide SAX "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of SAX is with you. Should SAX prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute SAX, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use SAX (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the SAX to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

### Copyright Disclaimers

This page includes statements to that effect by David Megginson, who would have been able to claim copyright for the original work.

#### SAX 1.0

Version 1.0 of the Simple API for XML (SAX), created collectively by the membership of the XML-DEV mailing list, is hereby released into the public domain.

No one owns SAX: you may use it freely in both commercial and non-commercial applications, bundle it with your software distribution, include it on a CD-ROM, list the source code in a book, mirror the documentation at your own web site, or use it in any other way you see fit.

David Megginson, [sax@megginson.com](mailto:sax@megginson.com)  
1998-05-11

#### SAX 2.0

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, [david@megginson.com](mailto:david@megginson.com)  
2000-05-05

%% The following software may be included in this product: Cryptix; Use of any of this software is governed by the terms of the license below:

#### Cryptix General License

Copyright © 1995-2003 The Cryptix Foundation Limited. All rights reserved.

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

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

2. Redistributions in binary form must reproduce the above copyright notice, this

list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

%% The following software may be included in this product: W3C XML Schema Test Collection; Use of any of this software is governed by the terms of the license below:

#### W3C DOCUMENT NOTICE AND LICENSE

Copyright 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

<http://www.w3.org/Consortium/Legal/>

Public documents on the W3C site are provided by the copyright holders under the following license. The software or Document Type Definitions (DTDs) associated with W3C specifications are governed by the Software Notice. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice of the form: "Copyright [\$date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>" (Hypertext is preferred, but a textual representation is permitted.)
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,

## Text of Third-Party Software Licenses

WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

-----

This formulation of W3C's notice and license became active on April 05 1999 so as to account for the treatment of DTDs, schema's and bindings. See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org. webmaster (last updated by reagle on 1999/04/99.)

% The following software may be included in this product: Stax API; Use of any of this software is governed by the terms of the license below:

Streaming API for XML (JSR-173) Specification  
Reference Implementation  
License Agreement

READ THE TERMS OF THIS (THE "AGREEMENT") CAREFULLY BEFORE VIEWING OR USING THE SOFTWARE LICENSED HEREUNDER. BY VIEWING OR USING THE SOFTWARE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCESSING THE SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO ORIGINAL CONTRIBUTOR, DEFINED HEREIN.

### 1.0 DEFINITIONS.

1.1. "BEA" means BEA Systems, Inc., the licensor of the Original Code.

1.2. "Contributor" means BEA and each entity that creates or contributes to the creation of Modifications.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof and corresponding documentation released with the source code.

1.4. "Executable" means Covered Code in any form other than Source Code.

1.5. "FCS" means first commercial shipment of a product.

1.6. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

(a) Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

(b) Any new file that contains any part of the Original Code or previous Modifications.

1.7. "Original Code" means Source Code of computer software code Reference Implementation.

1.8. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent for which the grantor has the right to grant a license.

1.9. "Reference Implementation" means the prototype or "proof of concept" implementation of the Specification developed and made available for license by or on behalf of BEA.

1.10. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated documentation, interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice.

1.11. "Specification" means the written specification for the Streaming API for XML , Java technology developed pursuant to the Java Community Process.

1.12. "Technology Compatibility Kit" or "TCK" means the documentation, testing tools and test suites associated with the Specification as may be revised by BEA from time to time, that is provided so that an implementer of the Specification may determine if its implementation is compliant with the Specification.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this Agreement or a future version of this Agreement issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2.0 SOURCE CODE LICENSE.

2.1. Copyright Grant. Subject to the terms of this Agreement, each Contributor hereby grants You a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Covered Code of such Contributor, if any, and such derivative works, in Source Code and Executable form.

2.2. Patent Grant. Subject to the terms of this Agreement, each Contributor hereby grants You a non-exclusive, worldwide, royalty-free patent license under the Patent Claims to make, use, sell, offer to sell, import and otherwise transfer the Covered Code prepared and provided by such Contributor, if any, in Source Code and Executable form. This patent license shall apply to the Covered Code if, at the time a Modification is added by the Contributor, such addition of the Modification causes such combination to be covered by the Patent Claims. The patent license shall not apply to any other combinations which include the Modification.

2.3. Conditions to Grants. You understand that although each Contributor grants the licenses to the Covered Code prepared by it, no assurances are

provided by any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a thirdparty patent license is required to allow You to distribute Covered Code, it is Your responsibility to acquire that license before distributing such code.

2.4. Contributors' Representation. Each Contributor represents that to its knowledge it has sufficient copyright rights in the Covered Code it provides , if any, to grant the copyright license set forth in this Agreement.

### 3.0 DISTRIBUTION RESTRICTIONS.

#### 3.1. Application of Agreement.

The Modifications which You create or to which You contribute are governed by the terms of this Agreement, including without limitation Section 2.0. The Source Code version of Covered Code may be distributed only under the terms of this Agreement or a future version of this Agreement released under Section 6.1, and You must include a copy of this Agreement with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this Agreement or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.3.

#### 3.2. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by BEA and including the name of BEA in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

%% The following software may be included in this product: X Window System; Use of any of this software is governed by the terms of the license below:  
Copyright The Open Group

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

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

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

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this

Software without prior written authorization from The Open Group.

Portions also covered by other licenses as noted in the above URL.

%% The following software may be included in this product: dom4j v. 1.6; Use of any of this software is governed by the terms of the license below:

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

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.

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

3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact [dom4j-info@metastuff.com](mailto:dom4j-info@metastuff.com).

4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.

5. Due credit should be given to the DOM4J Project - <http://www.dom4j.org>

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

Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved.

%% The following software may be included in this product: Retroweaver; Use of any of this software is governed by the terms of the license below:

Copyright (c) February 2004, Toby Reyelts All rights reserved.

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

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

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

## Text of Third-Party Software Licenses

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

%% The following software may be included in this product: stripper; Use of any of this software is governed by the terms of the license below:

Stripper : debug information stripper Copyright (c) 2003 Kohsuke Kawaguchi All rights reserved.

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

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

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

%% The following software may be included in this product: libpng official PNG reference library; Use of any of this software is governed by the terms of the license below:

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

libpng version 1.2.6, December 3, 2004, is Copyright (c) 2004 Glenn Randers-Pehrson, and is distributed according to the same disclaimer and license as libpng-1.2.5 with the following individual added to the list of Contributing Authors Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5 - October 3, 2002, are Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors Simon-Pierre Cadieux Eric S. Raymond Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors: Tom Lane Glenn Randers-Pehrson Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997 Andreas Dilger Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors: John Bowler Kevin Bracey Sam Bushell Magnus Holmgren Greg Roelofs Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger  
Dave Martindale  
Guy Eric Schalnat  
Paul Schmidt  
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee,

## Text of Third-Party Software Licenses

and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

A "png\_get\_copyright" function is available, for convenient use in "about" boxes and the like:

```
printf("%s",png_get_copyright(NULL));
```

Also, the PNG logo (in PNG format, of course) is supplied in the files "pngbar.png" and "pngbar.jpg" (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson  
glenrnp at users.sourceforge.net  
December 3, 2004

%% The following software may be included in this product: Libungif - An uncompressed GIF library; Use of any of this software is governed by the terms of the license below:

The GIFLIB distribution is Copyright (c) 1997 Eric S.Raymond

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

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

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

%% The following software may be included in this product: Ant; Use of any of this software is governed by the terms of the license below: License The Apache Software License Version 2.0

The Apache Software License Version 2.0 applies to all releases of Ant starting with ant 1.6.1

```
/*
 *          Apache License
 *      Version 2.0, January 2004
 *      http://www.apache.org/licenses/
 *
 *      TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
 *
 *      1. Definitions.
```

\*  
 \* "License" shall mean the terms and conditions for use, reproduction,  
 \* and distribution as defined by Sections 1 through 9 of this document.  
 \*  
 \* "Licensor" shall mean the copyright owner or entity authorized by  
 \* the copyright owner that is granting the License.  
 \*  
 \* "Legal Entity" shall mean the union of the acting entity and all  
 \* other entities that control, are controlled by, or are under common  
 \* control with that entity. For the purposes of this definition,  
 \* "control" means (i) the power, direct or indirect, to cause the  
 \* direction or management of such entity, whether by contract or  
 \* otherwise, or (ii) ownership of fifty percent (50%) or more of the  
 \* outstanding shares, or (iii) beneficial ownership of such entity.  
 \*  
 \* "You" (or "Your") shall mean an individual or Legal Entity  
 \* exercising permissions granted by this License.  
 \*  
 \* "Source" form shall mean the preferred form for making modifications,  
 \* including but not limited to software source code, documentation  
 \* source, and configuration files.  
 \*  
 \* "Object" form shall mean any form resulting from mechanical  
 \* transformation or translation of a Source form, including but  
 \* not limited to compiled object code, generated documentation,  
 \* and conversions to other media types.  
 \*  
 \* "Work" shall mean the work of authorship, whether in Source or  
 \* Object form, made available under the License, as indicated by a  
 \* copyright notice that is included in or attached to the work  
 \* (an example is provided in the Appendix below).  
 \*  
 \* "Derivative Works" shall mean any work, whether in Source or Object  
 \* form, that is based on (or derived from) the Work and for which the  
 \* editorial revisions, annotations, elaborations, or other modifications  
 \* represent, as a whole, an original work of authorship. For the purposes  
 \* of this License, Derivative Works shall not include works that remain  
 \* separable from, or merely link (or bind by name) to the interfaces of,  
 \* the Work and Derivative Works thereof.  
 \*  
 \* "Contribution" shall mean any work of authorship, including  
 \* the original version of the Work and any modifications or additions  
 \* to that Work or Derivative Works thereof, that is intentionally  
 \* submitted to Licensor for inclusion in the Work by the copyright owner  
 \* or by an individual or Legal Entity authorized to submit on behalf of  
 \* the copyright owner. For the purposes of this definition, "submitted"  
 \* means any form of electronic, verbal, or written communication sent  
 \* to the Licensor or its representatives, including but not limited to  
 \* communication on electronic mailing lists, source code control systems,  
 \* and issue tracking systems that are managed by, or on behalf of, the  
 \* Licensor for the purpose of discussing and improving the Work, but  
 \* excluding communication that is conspicuously marked or otherwise  
 \* designated in writing by the copyright owner as "Not a Contribution."  
 \*  
 \* "Contributor" shall mean Licensor and any individual or Legal Entity  
 \* on behalf of whom a Contribution has been received by Licensor and  
 \* subsequently incorporated within the Work.  
 \*  
 \* 2. Grant of Copyright License. Subject to the terms and conditions of

\*       this License, each Contributor hereby grants to You a perpetual,  
\*       worldwide, non-exclusive, no-charge, royalty-free, irrevocable  
\*       copyright license to reproduce, prepare Derivative Works of,  
\*       publicly display, publicly perform, sublicense, and distribute the  
\*       Work and such Derivative Works in Source or Object form.  
\*  
\*       3. Grant of Patent License. Subject to the terms and conditions of  
\*       this License, each Contributor hereby grants to You a perpetual,  
\*       worldwide, non-exclusive, no-charge, royalty-free, irrevocable  
\*       (except as stated in this section) patent license to make, have made,  
\*       use, offer to sell, sell, import, and otherwise transfer the Work,  
\*       where such license applies only to those patent claims licensable  
\*       by such Contributor that are necessarily infringed by their  
\*       Contribution(s) alone or by combination of their Contribution(s)  
\*       with the Work to which such Contribution(s) was submitted. If You  
\*       institute patent litigation against any entity (including a  
\*       cross-claim or counterclaim in a lawsuit) alleging that the Work  
\*       or a Contribution incorporated within the Work constitutes direct  
\*       or contributory patent infringement, then any patent licenses  
\*       granted to You under this License for that Work shall terminate  
\*       as of the date such litigation is filed.  
\*  
\*       4. Redistribution. You may reproduce and distribute copies of the  
\*       Work or Derivative Works thereof in any medium, with or without  
\*       modifications, and in Source or Object form, provided that You  
\*       meet the following conditions:  
\*  
\*       (a) You must give any other recipients of the Work or  
\*           Derivative Works a copy of this License; and  
\*  
\*       (b) You must cause any modified files to carry prominent notices  
\*           stating that You changed the files; and  
\*  
\*       (c) You must retain, in the Source form of any Derivative Works  
\*           that You distribute, all copyright, patent, trademark, and  
\*           attribution notices from the Source form of the Work,  
\*           excluding those notices that do not pertain to any part of  
\*           the Derivative Works; and  
\*  
\*       (d) If the Work includes a "NOTICE" text file as part of its  
\*           distribution, then any Derivative Works that You distribute must  
\*           include a readable copy of the attribution notices contained  
\*           within such NOTICE file, excluding those notices that do not  
\*           pertain to any part of the Derivative Works, in at least one  
\*           of the following places: within a NOTICE text file distributed  
\*           as part of the Derivative Works; within the Source form or  
\*           documentation, if provided along with the Derivative Works; or,  
\*           within a display generated by the Derivative Works, if and  
\*           wherever such third-party notices normally appear. The contents  
\*           of the NOTICE file are for informational purposes only and  
\*           do not modify the License. You may add Your own attribution  
\*           notices within Derivative Works that You distribute, alongside  
\*           or as an addendum to the NOTICE text from the Work, provided  
\*           that such additional attribution notices cannot be construed  
\*           as modifying the License.  
\*  
\*       You may add Your own copyright statement to Your modifications and  
\*       may provide additional or different license terms and conditions  
\*       for use, reproduction, or distribution of Your modifications, or

\* for any such Derivative Works as a whole, provided Your use,  
 \* reproduction, and distribution of the Work otherwise complies with  
 \* the conditions stated in this License.

\* 5. Submission of Contributions. Unless You explicitly state otherwise,  
 \* any Contribution intentionally submitted for inclusion in the Work  
 \* by You to the Licensor shall be under the terms and conditions of  
 \* this License, without any additional terms or conditions.  
 \* Notwithstanding the above, nothing herein shall supersede or modify  
 \* the terms of any separate license agreement you may have executed  
 \* with Licensor regarding such Contributions.

\* 6. Trademarks. This License does not grant permission to use the trade  
 \* names, trademarks, service marks, or product names of the Licensor,  
 \* except as required for reasonable and customary use in describing the  
 \* origin of the Work and reproducing the content of the NOTICE file.

\* 7. Disclaimer of Warranty. Unless required by applicable law or  
 \* agreed to in writing, Licensor provides the Work (and each  
 \* Contributor provides its Contributions) on an "AS IS" BASIS,  
 \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or  
 \* implied, including, without limitation, any warranties or conditions  
 \* of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A  
 \* PARTICULAR PURPOSE. You are solely responsible for determining the  
 \* appropriateness of using or redistributing the Work and assume any  
 \* risks associated with Your exercise of permissions under this License.

\* 8. Limitation of Liability. In no event and under no legal theory,  
 \* whether in tort (including negligence), contract, or otherwise,  
 \* unless required by applicable law (such as deliberate and grossly  
 \* negligent acts) or agreed to in writing, shall any Contributor be  
 \* liable to You for damages, including any direct, indirect, special,  
 \* incidental, or consequential damages of any character arising as a  
 \* result of this License or out of the use or inability to use the  
 \* Work (including but not limited to damages for loss of goodwill,  
 \* work stoppage, computer failure or malfunction, or any and all  
 \* other commercial damages or losses), even if such Contributor  
 \* has been advised of the possibility of such damages.

\* 9. Accepting Warranty or Additional Liability. While redistributing  
 \* the Work or Derivative Works thereof, You may choose to offer,  
 \* and charge a fee for, acceptance of support, warranty, indemnity,  
 \* or other liability obligations and/or rights consistent with this  
 \* License. However, in accepting such obligations, You may act only  
 \* on Your own behalf and on Your sole responsibility, not on behalf  
 \* of any other Contributor, and only if You agree to indemnify,  
 \* defend, and hold each Contributor harmless for any liability  
 \* incurred by, or claims asserted against, such Contributor by reason  
 \* of your accepting any such warranty or additional liability.

\* END OF TERMS AND CONDITIONS

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

\* To apply the Apache License to your work, attach the following  
 \* boilerplate notice, with the fields enclosed by brackets "[]"  
 \* replaced with your own identifying information. (Don't include  
 \* the brackets!) The text should be enclosed in the appropriate  
 \* comment syntax for the file format. We also recommend that a

## Text of Third-Party Software Licenses

```
*      file or class name and description of purpose be included on the
*      same "printed page" as the copyright notice for easier
*      identification within third-party archives.
*
*      Copyright [yyyy] Apache Software Foundation
*
*      Licensed under the Apache License, Version 2.0 (the "License");
*      you may not use this file except in compliance with the License.
*      You may obtain a copy of the License at
*
*          http://www.apache.org/licenses/LICENSE-2.0
*
*      Unless required by applicable law or agreed to in writing, software
*      distributed under the License is distributed on an "AS IS" BASIS,
*      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
*      See the License for the specific language governing permissions and
*      limitations under the License.
*/

```

You can download the original license file [here](#).

The License is accompanied by a NOTICE

```
=====
== NOTICE file corresponding to the section 4 d of
== the Apache License, Version 2.0,
== in this case for the Apache Ant distribution.
=====
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This product includes also software developed by : - the W3C consortium  
(<http://www.w3c.org>) , - the SAX project (<http://www.saxproject.org>)

Please read the different LICENSE files present in the root directory of this distribution.

The names "Ant" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission.  
For written permission, please contact apache@apache.org.

The Apache Software License, Version 1.1

The Apache Software License, Version 1.1, applies to all versions of up to ant1.6.0 included.

```
/*
* =====
*           The Apache Software License, Version 1.1
* =====
*
*   Copyright (C) 2000-2003 The Apache Software Foundation. All
*   rights reserved.
*
* Redistribution and use in source and binary forms, with or without modifica-
* tion, are permitted provided that the following conditions are met:
*
* 1. Redistributions of source code must retain the above copyright notice,
```

```

*   this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright notice,
*   this list of conditions and the following disclaimer in the documentation
*   and/or other materials provided with the distribution.
*
* 3. The end-user documentation included with the redistribution, if any, must
*   include the following acknowledgment: "This product includes software
*   developed by the Apache Software Foundation (http://www.apache.org/)."
*   Alternately, this acknowledgment may appear in the software itself, if
*   and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Ant" and "Apache Software Foundation" must not be used to
*   endorse or promote products derived from this software without prior
*   written permission. For written permission, please contact
*   apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache", nor may
*   "Apache" appear in their name, without prior written permission of the
*   Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES,
* INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
* FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
* INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLU-
* DING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
* OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
* ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* This software consists of voluntary contributions made by many individuals
* on behalf of the Apache Software Foundation. For more information on the
* Apache Software Foundation, please see http://www.apache.org.
*
*/

```

%% The following software may be included in this product: XML Resolver library; Use of any of this software is governed by the terms of the license below:

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control

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

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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell,

sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

## Text of Third-Party Software Licenses

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licenser provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

%% The following software may be included in this product: ICU4J; Use of any of this software is governed by the terms of the license below:

ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE Copyright (c)

1995-2003 International Business Machines Corporation and others All rights reserved Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

% The following software may be included in this product: NekoHTML; Use of any of this software is governed by the terms of the license below: The CyberNeko Software License, Version 1.0

(C) Copyright 2002,2003, Andy Clark. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:  
 "This product includes software developed by Andy Clark."  
 Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "CyberNeko" and "NekoHTML" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact andy@cyberneko.net.
5. Products derived from this software may not be called "CyberNeko", nor may "CyberNeko" appear in their name, without prior written permission of the author.

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

## Text of Third-Party Software Licenses

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

=====

This license is based on the Apache Software License, version 1.1

%% The following software may be included in this product: Jing; Use of any of this software is governed by the terms of the license below: Jing Copying Conditions

Copyright (c) 2001-2003 Thai Open Source Software Center Ltd All rights reserved.

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

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

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

%% The following software may be included in this product: RelaxNGCC; Use of any of this software is governed by the terms of the license below:

Copyright (c) 2000-2003 Daisuke Okajima and Kohsuke Kawaguchi.  
All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (<http://relaxngcc.sf.net/>)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names of the copyright holders must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact the copyright holders.

5. Products derived from this software may not be called "RELAXNGCC", nor may "RELAXNGCC" appear in their name, without prior written permission of the copyright holders.

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

%% The following software may be included in this product: RELAX NG Object Model/Parser; Use of any of this software is governed by the terms of the license below: The MIT License

Copyright (c)

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

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

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

%% The following software may be included in this product: XFree86-VidMode Extension; Use of any of this software is governed by the terms of the license below: Version 1.1 of Project Licence.

## Text of Third-Party Software Licenses

Copyright (C) 1994-2004 The Project, Inc. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution, and in the same place and form as other copyright, license and disclaimer information.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by The XFree86 Project, Inc (<http://www.xfree86.org/>) and its contributors", in the same place and form as other third-party acknowledgments. Alternately, this acknowledgment may appear in the software itself, in the same form and location as other such third-party acknowledgments.
4. Except as contained in this notice, the name of The XFree86 Project, Inc shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from TheXFree86 Project, Inc.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE XFREE86PROJECT, INC OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

%% The following software may be included in this product: RelaxNGCC; Use of any of this software is governed by the terms of the license below: This is version 2003-May-08 of the Info-ZIP copyright and license. The definitive version of this document should be available at <ftp://ftp.info-zip.org/pub/infozip/license.html> indefinitely.

Copyright (c) 1990-2003 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen, Paul von Behren, Rich

Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, definition, disclaimer, and this list of conditions.
2. Redistributions in binary form (compiled executables) must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution. The sole exception to this condition is redistribution of a standard UnZipSFX binary (including SFXWiz) as part of a self-extracting archive; that is permitted without inclusion of this license, as long as the normal SFX banner has not been removed from the binary or disabled.
3. Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s).
4. Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "UnZipSFX," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases.

%% The following software may be included in this product: XML Security; Use of any of this software is governed by the terms of the license below: The Apache Software License, Version 1.1 PDF

Copyright (C) 2002 The Apache Software Foundation.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).". Alternately, this acknowledgment may appear in the software itself, if and wherever such

third-party acknowledgments normally appear.

4. The names "Apache Forrest" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org>.

%% The following software may be included in this product: Regexp, Regular Expression Package v. 1.2; Use of any of this software is governed by the terms of the license below: The Apache Software License, Version 1.1 Copyright (c) 2001 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

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

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

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).". Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" and "Apache Turbine" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", "Apache Turbine", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

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

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

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

---

%% The following software may be included in this product: zlib; Use of any of this software is governed by the terms of the license below:

`zlib.h -- interface of the 'zlib' general purpose compression library  
version 1.1.3, July 9th, 1998`

Copyright (C) 1995-1998 Jean-loup Gailly and Mark Adler

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

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly <a href="mailto:jloup@gzip.org">jloup@gzip.org</a>	Mark Adler <a href="mailto:madler@alumni.caltech.edu">madler@alumni.caltech.edu</a>
------------------------------------------------------------------------	----------------------------------------------------------------------------------------

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files <ftp://ds.internic.net/rfc/rfc1950.txt> (zlib format), [rfc1951.txt](ftp://ds.internic.net/rfc/rfc1951.txt) (deflate format) and [rfc1952.txt](ftp://ds.internic.net/rfc/rfc1952.txt) (gzip format)

%% The following software may be included in this product: Mozilla Rhino. Use of any of this software is governed by the terms of the license below:

```
* The contents of this file are subject to the Netscape Public
* License Version 1.1 (the "License"); you may not use this file
* except in compliance with the License. You may obtain a copy of
* the License at http://www.mozilla.org/NPL/
*
* Software distributed under the License is distributed on an "AS
* IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or
* implied. See the License for the specific language governing
* rights and limitations under the License.
*
* The Original Code is Rhino code, released
* May 6, 1999.
```

```
*  
* The Initial Developer of the Original Code is Netscape  
* Communications Corporation. Portions created by Netscape are  
* Copyright (C) 1997-2000 Netscape Communications Corporation. All  
* Rights Reserved.  
*  
* Contributor(s):  
*  
* Kemal Bayram  
* Patrick Beard  
* Norris Boyd  
* Igor Bukanov, igor@mir2.org  
* Brendan Eich  
* Ethan Hugg  
* Roger Lawrence  
* Terry Lucas  
* Mike McCabe  
* Milen Nankov  
* Attila Szegedi, szegedia@freemail.hu  
* Ian D. Stewart  
* Andi Vajda  
* Andrew Wason  
*/
```

%% The following software may be included in this product: Apache Derby. Use  
of any of this software is governed by the terms of the license below:

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

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

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

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

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

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

## Text of Third-Party Software Licenses

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

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

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

%% The following software may be included in this product: 7-Zip. Use of any of this software is governed by the terms of the license below:

~~~~~

## Text of Third-Party Software Licenses

License for use and distribution  
~~~~~

7-Zip Copyright (C) 1999-2007 Igor Pavlov.

Licenses for files are:

- 1) 7z.dll: GNU LGPL + AES code license + unRAR restriction
- 2) 7za.exe, 7z.sfx and 7zCon.sfx: GNU LGPL + AES code license
- 3) All other files: GNU LGPL

The GNU LGPL + AES code license + unRAR restriction means that you must follow GNU LGPL rules, AES code license rules and unRAR restriction rules.

The GNU LGPL + AES code license means that you must follow both GNU LGPL rules and AES code license rules.

Note:

You can use 7-Zip on any computer, including a computer in a commercial organization. You don't need to register or pay for 7-Zip.

GNU LGPL information

-----

## GNU Lesser General Public License

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

## Text of Third-Party Software Licenses

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of

these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1

and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents

of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not

## Text of Third-Party Software Licenses

accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version

ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and an idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
```

```
This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.
```

```
You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
```

## Text of Third-Party Software Licenses

Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in  
the library `Frob' (a library for tweaking knobs) written  
by James Random Hacker.

signature of Ty Coon, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

### unRAR restriction

-----

The unRAR sources cannot be used to re-create the RAR compression algorithm,  
which is proprietary. Distribution of modified unRAR sources in separate form  
or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver.

### AES code license

-----

Copyright (c) 2001, Dr Brian Gladman

### LICENSE TERMS

The free distribution and use of this software in both source and binary form is allowed (with or without changes) provided that:

1. distributions of this source code include the above copyright notice, this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other associated materials;
3. the copyright holder's name is not used to endorse products built using this software without specific written permission.

### DISCLAIMER

This software is provided 'as is' with no explicit or implied warranties in respect of its properties, including, but not limited to, correctness and fitness for purpose.

\*\*\*\*\*

%%The following software may be included in this product:  
UPX

Use of any of this software is governed by the terms of the license below:

-----BEGIN PGP SIGNED MESSAGE-----

```
oooooo    ooo  oooooooooo.    oooooooo  oooooo
`888'    `8'  `888   `Y88.  `88888     d8'
888      8   888   .d88'    Y888..8P
888      8   88800088P'    `8888'
888      8   888       .8PY888.
`88.     .8'  888       d8'  `888b
`YbodP'   o888o       o888o  o88888o
```

The Ultimate Packer for eXecutables  
Copyright (c) 1996-2000 Markus Oberhumer & Laszlo Molnar  
<http://wildsau.idv.uni-linz.ac.at/mfx/upx.html>  
<http://www.nexus.hu/upx>  
<http://upx.tsx.org>

PLEASE CAREFULLY READ THIS LICENSE AGREEMENT, ESPECIALLY IF YOU PLAN  
TO MODIFY THE UPX SOURCE CODE OR USE A MODIFIED UPX VERSION.

#### ABSTRACT

=====

UPX and UCL are copyrighted software distributed under the terms  
of the GNU General Public License (hereinafter the "GPL").

The stub which is imbedded in each UPX compressed program is part  
of UPX and UCL, and contains code that is under our copyright. The  
terms of the GNU General Public License still apply as compressing  
a program is a special form of linking with our stub.

As a special exception we grant the free usage of UPX for all  
executables, including commercial programs.

See below for details and restrictions.

#### COPYRIGHT

=====

UPX and UCL are copyrighted software. All rights remain with the authors.

UPX is Copyright (C) 1996-2000 Markus Franz Xaver Johannes Oberhumer  
UPX is Copyright (C) 1996-2000 Laszlo Molnar

UCL is Copyright (C) 1996-2000 Markus Franz Xaver Johannes Oberhumer

#### GNU GENERAL PUBLIC LICENSE

=====

## Text of Third-Party Software Licenses

UPX and the UCL library are free software; you can redistribute them and/or modify them under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

UPX and UCL are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; see the file COPYING.

### SPECIAL EXCEPTION FOR COMPRESSED EXECUTABLES

---

The stub which is imbedded in each UPX compressed program is part of UPX and UCL, and contains code that is under our copyright. The terms of the GNU General Public License still apply as compressing a program is a special form of linking with our stub.

Hereby Markus F.X.J. Oberhuber and Laszlo Molnar grant you special permission to freely use and distribute all UPX compressed programs (including commercial ones), subject to the following restrictions:

1. You must compress your program with a completely unmodified UPX version; either with our precompiled version, or (at your option) with a self compiled version of the unmodified UPX sources as distributed by us.
2. This also implies that the UPX stub must be completely unmodified, i.e. the stub imbedded in your compressed program must be byte-identical to the stub that is produced by the official unmodified UPX version.
3. The decompressor and any other code from the stub must exclusively get used by the unmodified UPX stub for decompressing your program at program startup. No portion of the stub may get read, copied, called or otherwise get used or accessed by your program.

### ANNOTATIONS

---

- You can use a modified UPX version or modified UPX stub only for programs that are compatible with the GNU General Public License.
- We grant you special permission to freely use and distribute all UPX compressed programs. But any modification of the UPX stub (such as, but not limited to, removing our copyright string or making your program non-decompressible) will immediately revoke your right to use and distribute a UPX compressed program.
- UPX is not a software protection tool; by requiring that you use the unmodified UPX version for your proprietary programs we make sure that any user can decompress your program. This protects both you and your users as nobody can hide malicious code - any program that cannot be decompressed is highly suspicious by definition.
- You can integrate all or part of UPX and UCL into projects that

are compatible with the GNU GPL, but obviously you cannot grant any special exceptions beyond the GPL for our code in your project.

- We want to actively support manufacturers of virus scanners and similar security software. Please contact us if you would like to incorporate parts of UPX or UCL into such a product.

Markus F.X.J. Oberhuber  
markus.oberhuber@jk.uni-linz.ac.at

Laszlo Molnar  
ml1050@cdata.tvnet.hu

Linz, Austria, 25 Feb 2000

#### Additional License(s)

The UPX license file is at <http://upx.sourceforge.net/upx-license.html>.

\*\*\*\*\*

%%The following software may be included in this product:  
LZMA Software Development Kit

Use of any of this software is governed by the terms of the license below:

#### License

LZMA SDK is available under any of the following licenses:

1. GNU Lesser General Public License (GNU LGPL)
2. Common Public License (CPL)
3. Simplified license for unmodified code (read SPECIAL EXCEPTION)
4. Proprietary license

This means that you can select one of these four options and follow rules of that license.

SPECIAL EXCEPTION: Igor Pavlov, as the author of this code, expressly permit you statically or dynamically to link your code (or bind by name) to the files from LZMA SDK without subjecting your linked code to the terms of the CPL or GNU LGPL. Any modification or addition to any file in the LZMA SDK, however, is subject to the GNU LGPL or CPL terms.

This SPECIAL EXCEPTION allows you to use LZMA SDK in applications with proprietary code, provided you keep the LZMA SDK code unmodified.

SPECIAL EXCEPTION #2: Igor Pavlov, as the author of this code, expressly permits you to use LZMA SDK 4.43 under the same terms and conditions contained in the License Agreement you have for any previous version of LZMA SDK developed by Igor Pavlov.

SPECIAL EXCEPTION #2 allows holders of proprietary licenses to use latest version of LZMA SDK as update for previous versions.

GNU LGPL and CPL are pretty similar and both these licenses are classified as free software licenses at <http://www.gnu.org/> and OSI-approved at <http://www.opensource.org/>.

LZMA SDK also is available under a proprietary license which can include:

## Text of Third-Party Software Licenses

1. The right to modify code from the LZMA SDK without subjecting the modified code to the terms of the CPL or GNU LGPL
2. Technical support for LZMA SDK via email

To request such a proprietary license, or for any additional consultations, send an email message, using the 7-Zip support page: Send message to LZMA developer

The source code of 7-Zip is released under the terms of the GNU GPL. You can download the source code of 7-Zip at 7-Zip's Source Forge Page

### Additional License(s)

The license included with the software differs slightly from the version posted on the website. Specifically it includes SPECIAL EXCEPTION #3, which is not present in the license on the website. The license from the software archive follows:

#### LICENSE

-----

LZMA SDK is available under any of the following licenses:

- 1) GNU Lesser General Public License (GNU LGPL)
- 2) Common Public License (CPL)
- 3) Simplified license for unmodified code (read SPECIAL EXCEPTION)
- 4) Proprietary license

It means that you can select one of these four options and follow rules of that license.

1,2) GNU LGPL and CPL licenses are pretty similar and both these licenses are classified as

- "Free software licenses" at <http://www.gnu.org/>
- "OSI-approved" at <http://www.opensource.org/>

#### 3) SPECIAL EXCEPTION

Igor Pavlov, as the author of this code, expressly permits you to statically or dynamically link your code (or bind by name) to the files from LZMA SDK without subjecting your linked code to the terms of the CPL or GNU GPL. Any modifications or additions to files from LZMA SDK, however, are subject to the GNU GPL or CPL terms.

SPECIAL EXCEPTION allows you to use LZMA SDK in applications with closed code, while you keep LZMA SDK code unmodified.

SPECIAL EXCEPTION #2: Igor Pavlov, as the author of this code, expressly permits you to use this code under the same terms and conditions contained in the License Agreement you have for any previous version of LZMA SDK developed by Igor Pavlov.

SPECIAL EXCEPTION #2 allows owners of proprietary licenses to use latest version of LZMA SDK as update for previous versions.

SPECIAL EXCEPTION #3: Igor Pavlov, as the author of this code, expressly permits you to use code of the following files: BranchTypes.h, LzmaTypes.h, LzmaTest.c, LzmaStateTest.c, LzmaAlone.cpp, LzmaAlone.cs, LzmaAlone.java as public domain code.

4) Proprietary license

LZMA SDK also can be available under a proprietary license which can include:

- 1) Right to modify code without subjecting modified code to the terms of the CPL or GNU LGPL
- 2) Technical support for code

To request such proprietary license or any additional consultations, send email message from that page:<http://www.7-zip.org/support.html>

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

You should have received a copy of the Common Public License along with this library.

Swig

SWIG is distributed under the following terms:

I.

Copyright (c) 1995-1998  
The University of Utah and the Regents of the University of California  
All Rights Reserved

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that (1) The above copyright notice and the following two paragraphs appear in all copies of the source code and (2) redistributions including binaries reproduces these notices in the supporting documentation. Substantial modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated in all files where they apply.

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

THE AUTHOR, THE UNIVERSITY OF CALIFORNIA, AND THE UNIVERSITY OF UTAH SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE,

## Text of Third-Party Software Licenses

SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

### II.

This software includes contributions that are Copyright (c) 1998-2005 University of Chicago.  
All rights reserved.

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

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

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

### III.

This software includes contributions that are Copyright (c) 2005-2006 Arizona Board of Regents (University of Arizona).  
All Rights Reserved

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that (1) The above copyright notice and the following two paragraphs appear in all copies of the source code and (2) redistributions including binaries reproduces these notices in the supporting documentation. Substantial modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated in all files where they apply.

THIS SOFTWARE IS PROVIDED BY THE UNIVERSITY OF ARIZONA AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE UNIVERSITY OF ARIZONA OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SysStat  
GNU Public License v2

Trove  
Lesser GNU Public License v2.1

The Trove library is licensed under the Lesser GNU Public License, which is included with the distribution in a file called LICENSE.txt.

Other license arrangements are possible, for a fee: contact ericdf@users.sourceforge.net for terms/pricing.

The PrimeFinder and HashFunctions classes in Trove are subject to the following license restrictions:

Copyright (c) 1999 CERN - European Organization for Nuclear Research.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

TrueZip  
Apache License v2

Uncomplicated Firewall  
GNU Public License v3

Unix ODBC  
Lesser GNU Public License v2

UTF8 CPP

// Copyright 2006 Nemanja Trifunovic

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

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and

## Text of Third-Party Software Licenses

all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

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

\*/

```
#ifndef UTF8_FOR_CPP_2675DCD0_9480_4c0c_B92A_CC14C027B731
#define UTF8_FOR_CPP_2675DCD0_9480_4c0c_B92A_CC14C027B731

#include "utf8/checked.h"
#include "utf8/unchecked.h"

#endif // header guard
```

Web Services Desc Lang for Java  
Common Public License 1.0

Webworks/Quadrabay

[Home](#) | [Contact Us](#) | [1-877-693-2967](#) | [Support Login](#)

[Support Central](#)  
[Resource Center](#) [View All](#) [?WebWork's Blog](#) [?Newsletter \(The Works\)](#) [?Webinar Library](#) [?Video Library](#) [?Whitepaper Library](#) [?Events](#)  
[Documentation](#) [?ePublisher Manuals](#) [?Wiki](#) [?Tech Notes](#)  
[My Support](#) [?Support Login](#) [?Support Guidelines](#) [?Request Support Login](#) [?Support FAQ](#)  
[ePublisher](#) [?System Requirements](#) [?Releases](#) [?Videos](#) [?Webinars](#) [?License Terms](#) [?Downloads](#)  
[Latest Release](#)

[ePublisher End User License Terms](#)

[Electronic End User License Agreement](#)

NOTICE TO END USER: THIS ELECTRONIC END USER LICENSE AGREEMENT ("EEULA") IS A CONTRACT BETWEEN YOU (either as an individual person or a single legal entity, who will be referred to in this EEULA as "You") AND QUADRABAY CORPORATION ("QUADRABAY"). BY INDICATING YOUR ACCEPTANCE BELOW, YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS EEULA.

This EEULA accompanies and upon acceptance becomes a legal agreement related to a Quadrabay software product ("Software") and related explanatory materials ("Documentation"). The term "Software" shall also include any upgrades, modified versions, updates, additions and copies of the Software licensed to you by Quadrabay. (The Software and Documentation collectively may be referred to herein as this "package.")

This EEULA is a license agreement and not an agreement for sale. Quadralay continues to own the copy of the Software and the physical media contained in this package and any other copy that you are authorized to make pursuant to this Agreement. Unless otherwise agreed in writing by Quadralay, this EEULA applies to any Software or Documentation obtained by you from Quadralay, whether by means of physical media or in electronic form by download. Subject to your payment of the applicable license fees, and further subject to your acceptance of this EEULA, Quadralay grants to you a nonexclusive license to use the Software and Documentation, on the following terms and conditions:

1. Use of the Software.
  - A. You may install the Software in a single location on a hard disk or other storage device of a computer, multiple computers or a server.
  - B. You agree to pay Quadralay the applicable charges for this nonexclusive license as specified in the applicable Quadralay schedule of prices or fees.
  - C. You acknowledge and understand that the fee schedule under this EEULA is scaleable, such that license fees are determined based upon the number of users to whom you give access, as defined in subparagraph 1.D.
  - D. A "user" under this EEULA shall include any one who installs and uses the software.
  - E. This clause removed (9/1/09)
  - F. Business Use. This License is personal to the Licensee. Unless covered under the terms of a separate Agreement with Quadralay, Licensee shall only use the Software for its internal business purposes. Licensee shall not sell, rent, lend, lease, license, sublicense, time share, assign, act as an application service provider or bureau service or otherwise collect fees or grant access to persons outside the Licensee's organization, except as permitted pursuant to Clause 3 (Transfer of Software).
  - G. You may display, modify, reproduce and distribute any visual or text content that is the result of the Software's publishing operation to the destination where output files are created (but not the Software or Documentation), in whole or in part, that is included with the Software (unless a specific notice indicates to the contrary). Such content may not be used in the production of lewd, obscene or pornographic material. You hereby agree to indemnify, to hold harmless and to defend Quadralay against any claims or lawsuits, including attorney's fees, that arise from the manner in which you display, modify, reproduce or distribute any such content.
  - H. You may make one backup copy of the Software, provided your backup copy is not installed or used on any computer. Backup copies must include the copyright, trademark, and other proprietary markings that are found on the original.
  - I. HOME USE. For the Software labeled "ePublisher Express" or "ePublisher Pro", the primary user of the computer on which the Software is installed or used may also install the Software on one home or portable computer for use by him or her in connection with your business, but not for use for or on behalf of third parties or for personal use.
  - J. TRAINING. This software is not to be used for any public or on-site training classes unless authorized in writing.
2. Copyright. The Software is owned by Quadralay and its suppliers, and its structure, organization and code are the valuable trade secrets of Quadralay and its suppliers. The Software is also protected by United States Copyright Law and International Treaty provisions. You may not copy the Software or the Documentation, except as set forth in the "Use of the Software" section. Any whole or partial copies that you are permitted to make pursuant to this EEULA must contain the same copyright and other proprietary notices that appear on or in the Software or Documentation. You agree not to modify, adapt or translate the Software except as may be expressly permitted under the European Directive on the Legal Protection of Computer Programs (14 May 1991) ("the Directive") without possibility of waiver. You also agree not to reverse-engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software or reverse-engineer templates, except as may be expressly required to be permitted under the Directive. Trademarks shall be used in accordance with accepted trademark practice, including identification of trademark owner's name. Trademarks can only be used to identify printed output produced by the Software. Such use of any trademark does not give you any rights of ownership in that trademark. Except

## Text of Third-Party Software Licenses

as stated above, this Agreement does not grant you any intellectual property rights in the Software.

3. Transfer. You may not rent, lease, sublicense or lend the Software or Documentation. You may, however, transfer all your rights to use the Software and Documentation to another person or legal entity provided that (i) you transfer this Agreement, the Software, including all copies, updates and prior versions and all copies of font software converted into other formats, and the Documentation to such person or entity, (ii) you retain no copies, including copies stored on a computer, and (iii) that the receiving party agrees by prior notice in writing to Quadralay to be bound by the terms and conditions of this EULA.

4. Multiple Environment Software/Multiple Language Software/Dual Media Software/Multiple Copies/Upgrades. If this package contains, or, in connection with the acquisition of the Software contained in this package you receive, two or more operating environment versions of the Software (e.g. two or more language translation versions of the Software, the same Software on two or more media [e.g., diskettes and a CD-ROM]), and/or you otherwise receive two or more copies of the Software, all such versions and copies of the Software are subject to this EEULA. You may make one back-up copy, in accordance with the terms of this Agreement, for each version of the Software you use. You may not rent, lease, sublicense, lend or transfer versions or copies of the Software you do not use, or Software contained on any unused media, except as part of the permanent transfer of all Software and Documentation as described above. If you acquire an upgrade or update for Software, you may use the previous version for ninety (90) days after you receive the new version in order to assist you in the transition to the new version, after which time you no longer have a license to use the previous version.

5. Support Services. Quadralay may provide you with support services related to the Software ("Support Services"). The provision and use of any such Support Services will be pursuant to Quadralay policies as described in the Documentation, in "online documentation", or in other Quadralay materials. There may be an additional fee for Support Services. You acknowledge and agree that Quadralay may use technical information you provide related to the Software and/or the Documentation for its business purposes, including for product support and development.

6. Audit. The fee schedule under this EEULA is scaleable, based upon the number of physical installations of the Software. Quadralay shall have the right to audit use of the Software upon its request, no more than once annually, with reasonable notice and during normal business hours. Quadralay shall keep compliance information confidential, except as may be necessary to enforce the license. Such an audit shall be at Quadralay's expense, unless the audit reveals a more than 10% discrepancy between the license fees paid and the license fees due, in which case Licensee shall pay the reasonable expense of the audit.

7. Limited Warranty. Quadralay warrants to you that the Software will perform substantially in accordance with the Documentation for the thirty (30) day period following your receipt of the Software. To make a warranty claim, you must return the Software to the location where you obtained it, or if you downloaded it, provide affirmation in writing that you have destroyed all copies of the Software and Documentation, along with a copy of your sales receipt within such thirty (30) day period. If the Software does not perform substantially in accordance with the Documentation, the entire and exclusive liability and remedy shall be limited to either, at Quadralay's option, the replacement of the Software or the refund of the license fee you paid for the Software. QUADRALAY AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. THE FOREGOING STATES THE SOLE AND EXCLUSIVE REMEDIES FOR QUADRALAY'S OR SUPPLIERS' BREACH OF WARRANTY. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, QUADRALAY AND ITS SUPPLIERS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, AS TO NONINFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. QUADRALAY PROVIDES THE SOFTWARE, DOCUMENTATION AND SUPPORT SERVICES, IF ANY, AS IS AND WITH ALL FAULTS. Some states or jurisdictions do not allow the exclusion of implied warranties or limitations on how

long an implied warranty may last, so the above limitations may not apply to you. To the extent permissible, any implied warranties that cannot be excluded are limited to DEFECTS DISCOVERED DURING THE PERIOD OF THE LIMITED WARRANTY (THIRTY (30)] DAYS); AS TO ANY DEFECTS DISCOVERED AFTER THE THIRTY (30) DAY PERIOD, THERE IS NO WARRANTY OR CONDITION OF ANY KIND. This warranty gives you specific legal rights. You may have other rights which vary from state to state or jurisdiction to jurisdiction.

8. Limitation of Liability. IN NO EVENT WILL QUADRALAY OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS, LOSS OF DATA, LOSS OF USE, BUSINESS INTERRUPTION, OR LOST SAVINGS, EVEN IF A QUADRALAY REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY THIRD PARTY. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, so the above limitations may not apply to you.

9. Governing Law and General Provisions.

A. Governing Law. This Agreement will be governed by the laws in force in the State of Texas excluding the application of its conflicts of law rules. This Agreement will not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. If any part of this Agreement is found void and unenforceable, it will not affect the validity of the balance of the Agreement, which shall remain valid and enforceable according to its terms. You agree that the Software will not be shipped, transferred or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions or regulations.

B. Term of Agreement. This EULA becomes effective when accepted by you or when you act to install, copy, download, access or otherwise use the Software, whichever occurs first; in any such event you agree to be bound by the terms of this EULA. Unless terminated by Quadralay pursuant to its terms, the EULA will remain in effect pursuant to its terms as long as you possess, access or otherwise use the Software, the Documentation or any components of either the Software or the Documentation.

C. Termination. Without prejudice to any other rights, Quadralay may terminate this EULA if you fail to comply with the terms and conditions of the EULA. In event of such termination, upon receipt of written notice of termination, you must immediately destroy all copies of the Software and Documentation and any component parts of such copies.

D. Entire Agreement. This Agreement may only be modified in writing signed by an authorized officer of Quadralay. This is the entire agreement between Quadralay and you relating to the Software and the Documentation, and it supersedes any prior representations, discussions, undertakings, end user license agreements, communications or advertising relating to the Software and the Documentation. To the extent the terms of any Quadralay policies or programs for support services may conflict with the terms of this EULA, the terms of this EULA shall control.

E. Reservation of Rights. All rights related to the Software and Documentation not expressly granted are reserved by Quadralay.

10. Notice to United States Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. ??101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. ?12.212 or 48 C.F.R. ??227.7202, as applicable. Consistent with 48 C.F.R. ?12.212 or 48 C.F.R. ?? 227.7202-1 through 227.7202-4, as applicable, the "Commercial Computer Software" and "Commercial Computer Software Documentation" are being licensed to U.S. Government end users (i) only as "Commercial Items" and (ii) with only those rights as are granted to all other end users pursuant to the terms and conditions herein.

WebWorks is a trademark of Quadralay Corporation.

EULA/01.15.10/CAM

[Products](#) | [Services](#) | [Solutions](#) | [Partners](#) | [Resources](#) | [Support](#) | [Company](#) | [News](#) | [Contact](#)

| Site Map

? Copyright 1992-2010 Quadralay Corporation.  
All rights reserved. Various trademarks held by their respective owners.

XPP3

Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana Univeristy" and "Indiana Univeristy Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <http://www.extreme.indiana.edu/>.

5. Products derived from this software may not use "Indiana Univeristy" name nor may "Indiana Univeristy" appear in their name, without prior written permission of the Indiana University.

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

XStream

(BSD Style License)

Copyright (c) 2003-2006, Joe Walnes  
 Copyright (c) 2006-2007, XStream Committers  
 All rights reserved.

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

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

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

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

## Zlib

```
/* zlib.h -- interface of the 'zlib' general purpose compression library
   version 1.2.3, July 18th, 2005
```

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

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

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org	Mark Adler madler@alumni.caltech.edu
------------------------------------	-----------------------------------------

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

\*/

Apache License v2

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licenser shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

#### APACHE HTTP SERVER SUBCOMPONENTS:

The Apache HTTP Server includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For the mod\_mime\_magic component:

```
/*
 * mod_mime_magic: MIME type lookup via file magic numbers
 * Copyright (c) 1996-1997 Cisco Systems, Inc.
 *
 * This software was submitted by Cisco Systems to the Apache Group in July
 * 1997. Future revisions and derivatives of this source code must
 * acknowledge Cisco Systems as the original contributor of this module.
 * All other licensing and usage conditions are those of the Apache Group.
 *
 * Some of this code is derived from the free version of the file command
 * originally posted to comp.sources.unix. Copyright info for that program
 * is included below as required.
 * -----
 * - Copyright (c) Ian F. Darwin, 1987. Written by Ian F. Darwin.
 *
 * This software is not subject to any license of the American Telephone and
 * Telegraph Company or of the Regents of the University of California.
 *
```

## Text of Third-Party Software Licenses

```
* Permission is granted to anyone to use this software for any purpose on any
* computer system, and to alter it and redistribute it freely, subject to
* the following restrictions:
*
* 1. The author is not responsible for the consequences of use of this
* software, no matter how awful, even if they arise from flaws in it.
*
* 2. The origin of this software must not be misrepresented, either by
* explicit claim or by omission. Since few users ever read sources, credits
* must appear in the documentation.
*
* 3. Altered versions must be plainly marked as such, and must not be
* misrepresented as being the original software. Since few users ever read
* sources, credits must appear in the documentation.
*
* 4. This notice may not be removed or altered.
* -----
*/

```

For the modules\mappers\mod\_imap.c component:

```
"macmartinized" polygon code copyright 1992 by Eric Haines, erich@eye.com
```

For the server\util\_md5.c component:

```
*****
* NCSA HTTPd Server
* Software Development Group
* National Center for Supercomputing Applications
* University of Illinois at Urbana-Champaign
* 605 E. Springfield, Champaign, IL 61820
* httpd@ncsa.uiuc.edu
*
* Copyright (C) 1995, Board of Trustees of the University of Illinois
*
*****
* md5.c: NCSA HTTPd code which uses the md5c.c RSA Code
*
* Original Code Copyright (C) 1994, Jeff Hostetler, Spyglass, Inc.
* Portions of Content-MD5 code Copyright (C) 1993, 1994 by Carnegie Mellon
* University (see Copyright below).
* Portions of Content-MD5 code Copyright (C) 1991 Bell Communications
* Research, Inc. (Bellcore) (see Copyright below).
* Portions extracted from mpack, John G. Myers - jgm+@cmu.edu
* Content-MD5 Code contributed by Martin Hamilton (martin@net.lut.ac.uk)
*
*/

```

```
/* these portions extracted from mpack, John G. Myers - jgm+@cmu.edu */
/* (C) Copyright 1993,1994 by Carnegie Mellon University
* All Rights Reserved.
*
* Permission to use, copy, modify, distribute, and sell this software
* and its documentation for any purpose is hereby granted without
* fee, provided that the above copyright notice appear in all copies
```

```

* and that both that copyright notice and this permission notice
* appear in supporting documentation, and that the name of Carnegie
* Mellon University not be used in advertising or publicity
* pertaining to distribution of the software without specific,
* written prior permission. Carnegie Mellon University makes no
* representations about the suitability of this software for any
* purpose. It is provided "as is" without express or implied
* warranty.
*
* CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO
* THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
* AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE
* FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN
* AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
* OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
* SOFTWARE.
*/

```

```

/*
* Copyright (c) 1991 Bell Communications Research, Inc. (Bellcore)
*
* Permission to use, copy, modify, and distribute this material
* for any purpose and without fee is hereby granted, provided
* that the above copyright notice and this permission notice
* appear in all copies, and that the name of Bellcore not be
* used in advertising or publicity pertaining to this
* material without the specific, prior written permission
* of an authorized representative of Bellcore. BELLCORE
* MAKES NO REPRESENTATIONS ABOUT THE ACCURACY OR SUITABILITY
* OF THIS MATERIAL FOR ANY PURPOSE. IT IS PROVIDED "AS IS",
* WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES.
*/

```

For the srclib\apr\include\apr\_md5.h component:

```

/*
* This work is derived from material Copyright RSA Data Security, Inc.
*
* The RSA copyright statement and Licence for that original material is
* included below. This is followed by the Apache copyright statement and
* licence for the modifications made to that material.
*/
/* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
 rights reserved.

License to copy and use this software is granted provided that it
is identified as the "RSA Data Security, Inc. MD5 Message-Digest
Algorithm" in all material mentioning or referencing this software
or this function.

License is also granted to make and use derivative works provided
that such works are identified as "derived from the RSA Data
Security, Inc. MD5 Message-Digest Algorithm" in all material
mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either
the merchantability of this software or the suitability of this
software for any particular purpose. It is provided "as is"

```

## Text of Third-Party Software Licenses

without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

\*/

For the srclib\apr\passwd\apr\_md5.c component:

```
/*
 * This work is derived from material Copyright RSA Data Security, Inc.
 *
 * The RSA copyright statement and Licence for that original material is
 * included below. This is followed by the Apache copyright statement and
 * licence for the modifications made to that material.
 */
/* MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm
 */
/* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
 rights reserved.
```

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

\*/

```
/*
 * The apr_md5_encode() routine uses much code obtained from the FreeBSD 3.0
 * MD5 crypt() function, which is licenced as follows:
 * -----
 * "THE BEER-WARE LICENSE" (Revision 42):
 * <phk@login.dknet.dk> wrote this file. As long as you retain this notice you
 * can do whatever you want with this stuff. If we meet some day, and you think
 * this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp
 * -----
 */

```

For the srclib\apr-util\crypto\apr\_md4.c component:

```
* This is derived from material copyright RSA Data Security, Inc.
* Their notice is reproduced below in its entirety.
*
* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
* rights reserved.
```

```

*
* License to copy and use this software is granted provided that it
* is identified as the "RSA Data Security, Inc. MD4 Message-Digest
* Algorithm" in all material mentioning or referencing this software
* or this function.
*
* License is also granted to make and use derivative works provided
* that such works are identified as "derived from the RSA Data
* Security, Inc. MD4 Message-Digest Algorithm" in all material
* mentioning or referencing the derived work.
*
* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.
*
* These notices must be retained in any copies of any part of this
* documentation and/or software.
*/

```

For the srclib\apr-util\include\apr\_md4.h component:

```

*
* This is derived from material copyright RSA Data Security, Inc.
* Their notice is reproduced below in its entirety.
*
* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
* rights reserved.
*
* License to copy and use this software is granted provided that it
* is identified as the "RSA Data Security, Inc. MD4 Message-Digest
* Algorithm" in all material mentioning or referencing this software
* or this function.
*
* License is also granted to make and use derivative works provided
* that such works are identified as "derived from the RSA Data
* Security, Inc. MD4 Message-Digest Algorithm" in all material
* mentioning or referencing the derived work.
*
* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.
*
* These notices must be retained in any copies of any part of this
* documentation and/or software.
*/

```

For the srclib\apr-util\test\testdbm.c component:

```

/* =====
* The Apache Software License, Version 1.1
*
* Copyright (c) 2000-2002 The Apache Software Foundation. All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions

```

## Text of Third-Party Software Licenses

```
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. The end-user documentation included with the redistribution,
*    if any, must include the following acknowledgment:
*        "This product includes software developed by the
*         Apache Software Foundation (http://www.apache.org/)."
* Alternately, this acknowledgment may appear in the software itself,
* if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Apache" and "Apache Software Foundation" must
*    not be used to endorse or promote products derived from this
*    software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*    nor may "Apache" appear in their name, without prior written
*    permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* =====
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
* This file came from the SDBM package (written by oz@nexus.yorku.ca).
* That package was under public domain. This file has been ported to
* APR, updated to ANSI C and other, newer idioms, and added to the Apache
* codebase under the above copyright and license.
*/

```

For the srclib\apr-util\test\testmd4.c component:

```

*
* This is derived from material copyright RSA Data Security, Inc.
* Their notice is reproduced below in its entirety.
*
* Copyright (C) 1990-2, RSA Data Security, Inc. Created 1990. All
```

```

* rights reserved.
*
* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.
*
* These notices must be retained in any copies of any part of this
* documentation and/or software.
*/

```

For the srclib\apr-util\xml\expat\conftools\install-sh component:

```

#
# install - install a program, script, or datafile
# This comes from X11R5 (mit/util/scripts/install.sh) .
#
# Copyright 1991 by the Massachusetts Institute of Technology
#
# Permission to use, copy, modify, distribute, and sell this software and its
# documentation for any purpose is hereby granted without fee, provided that
# the above copyright notice appear in all copies and that both that
# copyright notice and this permission notice appear in supporting
# documentation, and that the name of M.I.T. not be used in advertising or
# publicity pertaining to distribution of the software without specific,
# written prior permission. M.I.T. makes no representations about the
# suitability of this software for any purpose. It is provided "as is"
# without express or implied warranty.
#

```

For the srclib\pcre\install-sh component:

```

#
# Copyright 1991 by the Massachusetts Institute of Technology
#
# Permission to use, copy, modify, distribute, and sell this software and its
# documentation for any purpose is hereby granted without fee, provided that
# the above copyright notice appear in all copies and that both that
# copyright notice and this permission notice appear in supporting
# documentation, and that the name of M.I.T. not be used in advertising or
# publicity pertaining to distribution of the software without specific,
# written prior permission. M.I.T. makes no representations about the
# suitability of this software for any purpose. It is provided "as is"
# without express or implied warranty.

```

For the pcre component:

PCRE LICENCE

-----

PCRE is a library of functions to support regular expressions whose syntax  
and semantics are as close as possible to those of the Perl 5 language.

Written by: Philip Hazel <ph10@cam.ac.uk>

University of Cambridge Computing Service,  
Cambridge, England. Phone: +44 1223 334714.

Copyright (c) 1997-2001 University of Cambridge

## Text of Third-Party Software Licenses

Permission is granted to anyone to use this software for any purpose on any computer system, and to redistribute it freely, subject to the following restrictions:

1. This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. In practice, this means that if you use PCRE in software which you distribute to others, commercially or otherwise, you must put a sentence like this

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England.

somewhere reasonably visible in your documentation and in any relevant files or online help data or similar. A reference to the ftp site for the source, that is, to

<ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/>

should also be given in the documentation.

3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. If PCRE is embedded in any software that is released under the GNU General Purpose Licence (GPL), or Lesser General Purpose Licence (LGPL), then the terms of that licence shall supersede any condition above with which it is incompatible.

The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself.

End PCRE LICENCE

For the test\zb.c component:

```
/*
=====
ZeusBench V1.01
```

This program is Copyright (C) Zeus Technology Limited 1996.

This program may be used and copied freely providing this copyright notice is not removed.

This software is provided "as is" and any express or implied warranties, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall Zeus Technology Ltd. be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute good or services; loss of use, data, or profits; or business interruption) however caused and on theory of liability. Whether in contract, strict liability or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the

possibility of such damage.

Written by Adam Twiss (adam@zeus.co.uk). March 1996

Thanks to the following people for their input:

Mike Belshe (mbelshe@netscape.com)

Michael Campanella (campanella@stevms.enet.dec.com)

\*/

For the expat xml parser component:

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd  
and Clark Cooper

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

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

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

=====

#### Axltistic License

No longer required by any Aster Database component.

=====

#### Create Commons 2.5

Creative Commons Attribution License v2.5  
License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

#### 1. Definitions

Collective Work means a work, such as a periodical issue, anthology or

## Text of Third-Party Software Licenses

encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

Derivative Work means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

Licensor means the individual or entity that offers the Work under the terms of this License.

Original Author means the individual or entity who created the Work.

Work means the copyrightable work of authorship offered under the terms of this License.

You means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;  
to create and reproduce Derivative Works;  
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;  
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.  
For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act

(or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

**4. Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

#### 5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

**6. Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL,

## Text of Third-Party Software Licenses

INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

### 8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, neither party will use the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time.

Creative Commons may be contacted at <http://creativecommons.org/>.

## Common Dev and Distrib 1.0

Common Development and Distribution License (CDDL)  
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

## 1. Definitions.

1.1. ?Contributor? means each individual or entity that creates or contributes to the creation of Modifications.

1.2. ?Contributor Version? means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. ?Covered Software? means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. ?Executable? means the Covered Software in any form other than Source Code.

1.5. ?Initial Developer? means the individual or entity that first makes Original Software available under this License.

1.6. ?Larger Work? means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. ?License? means this document.

1.8. ?Licensable? means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. ?Modifications? means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. ?Original Software? means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. ?Patent Claims? means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. ?Source Code? means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. ?You? (or ?Your?) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, ?You? includes any entity which controls, is controlled by, or is under common

## Text of Third-Party Software Licenses

control with You. For purposes of this definition, ?control? means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

### 2. License Grants.

#### 2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

- (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
- (b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
- (c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
- (d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

#### 2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- (a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
- (b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
- (c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
- (d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other

software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

### 3. Distribution Obligations.

#### 3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

#### 3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

#### 3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

#### 3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipient's rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

#### 3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You

## Text of Third-Party Software Licenses

offer.

### 3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

## 4. Versions of the License.

### 4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

### 4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

### 4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

## 5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN ?AS IS? BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

## 6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as ?Participant?) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

#### 7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY?S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

#### 8. U.S. GOVERNMENT END USERS.

The Covered Software is a ?commercial item,? as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of ?commercial computer software? (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and ?commercial computer software documentation? as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

#### 9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction?s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a

## Text of Third-Party Software Licenses

notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

### 10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

### NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Common Public License 1.0

Common Public License v1.0

Common Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

### 1. DEFINITIONS

Contribution means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
  - i) changes to the Program, and
  - ii) additions to the Program;where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

Contributor means any person or entity that distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement, including all Contributors.

## 2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

## 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
  - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
  - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

## Text of Third-Party Software Licenses

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

## 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

## 5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of

rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

#### 6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the

## Text of Third-Party Software Licenses

intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

### GNU Public License v2

#### GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

#### GNU GENERAL PUBLIC LICENSE

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively

when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary

form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

## Text of Third-Party Software Licenses

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE

POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

## Text of Third-Party Software Licenses

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

### GNU Public License v3

#### GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run

modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

#### TERMS AND CONDITIONS

##### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a

menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with

the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not

invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this

## Text of Third-Party Software Licenses

License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions;

the above requirements apply either way.

#### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the

rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is

conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you

to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

#### Lesser GNU Public License v2

##### GNU LIBRARY GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.  
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

##### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for

your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software

sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

#### GNU LIBRARY GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's

## Text of Third-Party Software Licenses

complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under

the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may

distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

## Text of Third-Party Software Licenses

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN

WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

## Text of Third-Party Software Licenses

### Lesser GNU Public License v2.1

#### GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who decide to use it. You  
can use it too, but we suggest you first think carefully about whether  
this license or the ordinary General Public License is the better  
strategy to use in any particular case, based on the explanations  
below.

When we speak of free software, we are referring to freedom of use,  
not price. Our General Public Licenses are designed to make sure that  
you have the freedom to distribute copies of free software (and charge  
for this service if you wish); that you receive source code or can get  
it if you want it; that you can change the software and use pieces of  
it in new free programs; and that you are informed that you can do  
these things.

To protect your rights, we need to make restrictions that forbid  
distributors to deny you these rights or to ask you to surrender these  
rights. These restrictions translate to certain responsibilities for  
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis  
or for a fee, you must give the recipients all the rights that we gave  
you. You must make sure that they, too, receive or can get the source  
code. If you link other code with the library, you must provide  
complete object files to the recipients, so that they can relink them  
with the library after making changes to the library and recompiling  
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the  
library, and (2) we offer you this license, which gives you legal  
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that  
there is no warranty for the free library. Also, if the library is  
modified by someone else and passed on, the recipients should know  
that what they have is not the original version, so that the original  
author's reputation will not be affected by problems that might be

introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany

## Text of Third-Party Software Licenses

it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever

changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact

that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to

be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

Lesser GNU Public License v3

GNU LESSER GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

#### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object

## Text of Third-Party Software Licenses

code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
  - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
  - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL

for conveying Corresponding Source.)

#### 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

#### 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

#### MIT License

Copyright (C) <year> by <copyright holders>

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

## Text of Third-Party Software Licenses

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### PostgreSQL License

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

Portions Copyright (c) 1996-2008, PostgreSQL Global Development Group

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

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

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

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

# Index

## Symbols

\_bee\_internal user 265  
\_bee\_sys user 265  
\_bee\_sysman user 265

## A

about this book 17  
access control 246  
access permissions  
    set SQL-MapReduce execution rights 138  
account 246  
    default database users 246  
account creation 149  
account, default admin account 149, 247  
account, default beehive account 149, 247  
ACT  
    check version of ACT 64  
    command history 77  
    command reference 75  
    command-line flag reference 63  
    file management commands 77  
    launching 61  
    launching on Linux or Solaris 62  
    launching on Mac 62  
    launching on Windows 61  
    list all databases 79  
    list all tables in Aster Database 78  
    parameter, setting with \set 76  
    parameters 63  
    parameters at SQL prompt 75  
    tab completion 74  
    transpose columns and rows in output 80  
    using ACT 68  
ACT commands 75  
    command-line options 63  
dF, install, remove, and download 140  
install 111  
    SQL-prompt options 75  
activate  
    invalidates analytic tables 26  
Active Directory authentication 257  
ADD CHECK 158  
ADD COLUMN 157

administrative role 247  
administrator 246  
    creating SQL account for 149  
    default admin 246  
    default administrator account 149, 247  
    role 247  
    role for 247  
    superuser 247  
administrator role 247  
Admission Control 268  
admission control 268  
Admission Limits 268  
    setting up 271  
Admission Limits tab of AMC 271  
ALTER COLUMN 158  
ALTER TABLE  
    examples 157  
    renaming partitions 46  
AMC  
    Admission Limits tab 271  
    Workload Management tab 276  
analytic tables  
    invalid 26  
analytics  
    SQL-MapReduce 82  
API 82  
    C and C++ 104  
    Java 97  
    SQL-MapReduce 82  
    stream API 117  
application code  
    install SQL-MapReduce application code 140  
archiving data 367  
argument 87  
    SQL-MapReduce function argument 87  
    to SQL-MapReduce Java API function 100  
Aster Data, about 17  
Aster Database  
    checking version of ACT 64  
    table types 20  
Aster Database architecture  
    planning 148  
Aster Database Backup 342  
Aster support portal 16  
authentication 252  
    Active Directory 257  
    LDAP 253

password 263  
types 252  
authorization 246  
available space 168  
  reclaim dead space 168

## B

backup 342  
  CLI 347  
  save to tape 367  
  scheduling a logical backup 355  
  scheduling a physical backup 354  
  scheduling a physical backup w/o incrementals 355  
  user interface for 347  
  version numbers and 365  
backup cluster 342  
backup node 342  
balance data  
  defined 369  
  invalidates analytic tables 26  
balance process  
  invalidates analytic tables 26  
beehive user 149, 247  
best practices 208  
  data modeling 208  
  disk space 168  
BIT, the application 369  
bulk export  
  Teradata-Aster connector 292  
bulk load  
  SQL-H 320  
  Teradata-Aster connector 292

## C

C and C++ API 104  
C SDK for SQL-MapReduce 104  
C# functions in Aster Database 117  
capacity  
  reclaim dead space 168  
case-sensitivity in identifiers  
  SQL-MapReduce function names 143  
casting in stream 119  
catalog\_admin 247  
celling data 29  
character set encoding  
  database setting 151  
CHECK 161  
  introduction to logical partitioning 36, 390  
child partition 369  
child table 34  
  detailed explanation 34  
  example 36, 390  
  example monthly child tables 48, 393

  INHERITS and 389  
class, packaging for SQL-MapReduce 101  
cleanup  
  VACUUM usage notes 168  
client software versions 64  
column  
  transpose columns and rows 80  
column alignment of query output 79  
command buffer 76  
command history 76, 77  
  send to file 76  
command reference  
  ACT command-line flags 63  
  ACT SQL prompt 75  
command-line (in SQL) commands for ACT 75  
command-line options for ACT 63  
commands  
  ACT command-line flags 63  
  ACT commands at SQL prompt 75  
compress data 153  
compression 153  
concurrency 270  
connect  
  SQL-H 320  
  Teradata-Aster Connector 292  
CONSTRAINT  
  examples of 393  
  notes on adding 158  
conventions 16  
coordinator 369  
copy SQL-MapReduce function to file 140  
copyright 17  
cost to read all the rows 225  
CREATE DATABASE  
  overview of 150  
CREATE DIMENSION TABLE 21  
CREATE FACT TABLE 21  
create mr\_driver table 325  
CREATE TABLE  
  overview for users 20  
CREATE TABLE AS SELECT  
  example 155  
  using to free up space 171  
CREATE USER  
  notes on 248  
credentials 149, 247  
  default for admin SQL user 149, 247  
  default for beehive SQL user 149, 247  
CSV 369  
CTAS 370  
  CREATE TABLE...AS SELECT  
        48  
cursor  
  ACT, cursors in 71

custom code  
  SQL-MapReduce 82  
  stream API 117  
customer support 16

## D

data 167  
  backing up data 342  
  compress 153  
  Hadoop and HCatalog 320  
  offsite copy 367  
  reclaim dead space 168  
  Teradata-Aster connector 292  
data export 290  
  Teradata-Aster connector 292  
data loading  
  SQL-H 320  
  Teradata-Aster connector 292  
data locality 370  
data modeling 176  
data modeling best practices 208  
data storage 167  
  reclaim dead space 168  
database 150  
  create 150  
  default users and roles 246  
  delete 151  
  drop 151  
  list all tables 78  
  list the databases in a cluster 79  
  name limitations 150  
database administration, introduction 150  
datatype  
  changing 158  
  datatypes supported for use as distribution keys 32  
date of publication 17  
db\_admin 247  
db\_superuser 149, 247  
dba 148  
  introduction to dba tasks 148  
DDL 370  
dead space, reclaiming 168  
default 246  
  default users and roles 246  
default database 246  
default login 149, 247  
default schema 263  
default user 246  
delete rows  
  with CREATE TABLE AS SELECT 171  
  with DELETE, VACUUM, and ANALYZE 170  
  with DROP TABLE 171  
  with TRUNCATE 170  
delete SQL-MapReduce function 144  
DELIMITER clause in stream function 119  
delimiters  
  setting in ACT 79  
deployment planning 148  
design guidelines 208  
dF command in ACT 140  
dimension table 22  
  compared with fact table 21  
  creating 21  
  distribution key 33  
  example CREATE 24  
  overview 22  
  partitioning options 31  
  replication 190  
dimensional model 21  
  example 23  
disaster recovery 367  
  overview 342  
disk 167  
  Aster Database disk usage, explained 167  
  free space 167  
  reclaim dead space on disk 168  
  required amount of free disk space 168  
disk full 168  
disk space 167  
  free space, required amount of 168  
  reclaim dead space 168  
disk usage  
  explained 167  
distributing data 29  
distribution key 31  
  datatypes supported for use as distribution keys 32  
  defined 371  
  inheritance and 47  
  primary key and 32  
  rules for declaring and using 32  
  vs. INHERITS 31  
  vs. PARTITION BY 29  
documentation conventions 16  
documentation version and updates 17  
documentation, about 15  
download a file from the cluster 143  
download command in ACT 140  
DOWNLOAD FILE 143  
drivers  
  Hadoop 320  
  Teradata-Aster Connector 292  
DROP COLUMN 158  
DROP CONSTRAINT 158  
DROP DATABASE  
  overview 151

## E

edition 17  
ELT 371  
error codes 380  
  list of error codes in Group Membership Data Dictionary  
    Views 380  
  SQL-MapReduce 115  
ETL 371  
expanded output mode 80  
EXPLAIN  
  localCost 225  
  network cost 225  
  reading an Aster Database EXPLAIN plan 224  
  width (of row) 225  
export 290  
  Teradata-Aster Connector 292  
  tools for 290  
export data 290  
  Teradata-Aster connector 292

## F

fact table 21  
  compared with dimension table 21  
  creating 21  
  distribution key 32  
  example CREATE 24  
  example partitioning with INHERITS 393  
  example partitioning with PARTITION KEY 23  
  overview 21  
  partitioning options 31  
FastExport  
  TD\_SPOOLMODE 314  
FETCH\_COUNT 71  
FETCH\_LIMIT 72  
fetch-count 71  
fetch-limit 72  
field separator, setting in ACT 79  
file  
  downloading from the cluster 143  
  install in Aster Database 111  
  installing in the cluster 142  
  redirect command history to file 76  
  redirect query history to file 76  
  redirect query results to file 77  
  removing from the cluster 144  
file input to ACT 77  
  with \i 77  
  with -f 64  
file upload in Aster Database 140  
flat file storage on Aster Database 140  
foreign key: declaration not supported in Aster Database 371  
fractioning data 29  
fragmenting data 29

free space 168

  how to free up disk space 168  
  reclaim dead space 168

fully replicated 31

functions

  downloading from the cluster 143  
  installation example 111  
  installing 140  
  installing in the cluster 142  
  list all functions 141  
  naming, case-sensitivity and 143  
  permissions 138  
  removing from the cluster 144  
  SQL-MapReduce 82  
  uppercase letters in a function name 143

## G

garbage collection  
  overview 168  
get latest documentation 17  
getting started 148  
global concurrency limit 270  
glossary 368  
GRANT  
  read-only access 252  
group  
  list all groups in Aster Database 78  
guidelines 208

## H

Hadoop 320, 372  
  configuring SQL-H 321  
  datatype support 332  
Hadoop connector 320  
hard restart  
  invalidates analytic tables 26  
hash partitioning 372  
HCatalog 320, 372  
  datatypes 332  
help 16  
history in ACT 77  
HortonWorks 320

## I

imbalanced 372  
incorporate 372  
incremental backups, turning off 355  
in-database applications 372  
index  
  example of adding indexes 48, 393  
  list all indexes in Aster Database 78  
inheritance 34

CHECK constraints and 161  
detailed explanation 34  
example 36, 390  
using INHERITS 389  
vs. partitioning 29  
inherited attributes of a partitioned (child) table 47  
INHERITS  
    detailed explanation 34, 389  
    example 393  
install  
    SQL-MapReduce function 111, 140  
    stream function 140  
install (command in ACT) 111  
install a file in SQL-MapReduce 142  
install command in ACT 140  
INSTALL FILE 142  
installed files 140  
    testing SQL-MapReduce C functions and 111  
    using in an SQL-MapReduce C function 108  
IWT 372

**J**  
jar file, loading in SQL-MapReduce 101  
Java API 97  
JDK for SQL-MapReduce 98  
job priority 268

**L**  
LDAP auth 252  
LDAP authentication 253  
licenses 396  
LIMIT  
    FETCH\_LIMIT as alternative to 72  
limit rows returned at a time 71  
limit total rows returned per query 72  
list all functions 141  
list all tables 78  
list available SQL-MapReduce functions 140  
list databases command 79  
list partitioning 372, 390  
load  
    install SQL-MapReduce application code 140  
load data  
    other tools 290  
    SQL-H (Hadoop) 320  
    Teradata-Aster connector 292  
load\_from\_hcatalog function 325  
loading  
    install SQL-MapReduce application code 140  
    SQL-H (Hadoop) 320  
    Teradata-Aster Connector 292  
localCost in EXPLAIN plan 225  
logical partitioning 34

CHECK constraints and 161  
defined 373  
detailed explanation 34  
example 36, 390  
inheritance of table attributes 47  
using INHERITS 389  
vs. physical partitioning 29  
login  
    default 246  
login, default 149, 247

**M**  
machine 373  
map function in MapReduce 98, 104  
MapReduce 82  
maximum number of concurrent queries 270  
memory usage in ACT: limit with FETCH\_COUNT paging of  
    query results 71  
mixed case function names 143  
mixed workload 268  
monitor  
    QoS 286  
    SQL-MapReduce execution 114  
mr\_driver 325

**N**  
naming  
    database names 150  
    SQL-MapReduce function names 143  
naming conventions  
    database names 150  
    SQL-MapReduce function names 143  
nc\_tables  
    defined 373  
nc\_qos\_concurrency 270  
ncluster\_backup 342  
network cost in EXPLAIN plan 225  
NIC bonding  
    defined 373  
node  
    backup node 342  
    defined 373  
node addition  
    invalidates analytic tables 26  
node failover  
    invalidates analytic tables 26  
node splitting 374  
NOT NULL 158  
null 158  
    constraint on column 158

## O

ODBC, defined 374  
offsite backup 367  
ON  
    ON to specify target of stream function 118  
operate function in SQL-MapReduce 99  
optimizer  
    example 393  
ORDER BY clause in stream function 118  
output  
    transpose columns and rows 80  
OUTPUTS clause in stream function 119

## P

paceling data 29  
parameter  
    settings for ACT 76  
    SQL-MapReduce arguments 100  
parent table  
    detailed explanation 34  
    example 36, 390  
parent-child table inheritance 374  
parent-child table relationships 34  
    using INHERITS 389  
partition 374  
    defined 374  
PARTITION BY  
    partition across all rows 144  
    SQL-MapReduce and 84  
    vs. DISTRIBUTE BY 30  
partition count  
    defined 374  
partition key  
    example 22  
    inheritance and 391  
    When is a partition key required? 152  
partition splitting  
    defined 374  
    invalidates analytic tables 26  
partitioned table 31  
PartitionFunction in SQL-MapReduce 98, 104  
partitioning 29  
    approaches to 29  
    best practices 187  
    constraints example 393  
    defined 374  
    example with INHERITS 393  
    example with PARTITION KEY 23  
    inheritance of table attributes 47  
    logical and physical partitioning compared 29  
passive coordinator 374  
password 252  
    default password for admin SQL user 149, 247

    default password for beehive SQL user 149, 247  
    password authentication for Aster Database users 252  
password authentication setup 263  
performance tuning  
    improve ACT performance with FETCH\_LIMIT 72  
    limit ACT memory use with FETCH\_COUNT 71  
    server-side cursors in ACT 71  
Perl functions in Aster Database 117  
permissions 246  
    default roles in Aster Database 246  
    set SQL-MapReduce execution rights 138  
    SQL-MapReduce 141  
physical partition 374  
physical partitioning 31  
    defined 374  
    vs. logical partitioning 29  
physically partitioned table 31  
pivot column and row output 80  
plan, reading the EXPLAIN output 224  
planning a deployment 148  
portal 16  
ports  
    QoS manager port 286  
predicate syntax for workload 282  
/primary 242  
primary data directories 242  
primary interface  
    defined 375  
primary key  
    defining a PRIMARY KEY constraint 164  
    distribution key and 32  
    inheritance and 391  
primary queen 375  
priority 268  
privileges  
    default roles in Aster Database 246  
prohibit user access to cluster 270  
Python functions in Aster Database 117

## Q

QAS 257  
QoS 268, 270  
QoS context attribute 282  
QoS Manager page 286  
quality of service 268  
quality-of-service context attribute 282  
queen  
    defined 375  
    port for QoS 286  
queen replacement  
    invalidates analytic tables 26  
query  
    concurrency 270

transpose results 80  
typing in ACT 68  
query buffer 76  
  send to file 76  
query history 76  
  send to file 76  
query plan 224  
query results 77  
  send to file 77  
Quest Authentication Services 257  
quiesce cluster 270  
quiet mode of ACT 65

## R

range partitioning 36, 375, 390  
rapala lure 119  
read-only access 252  
redirect command history to file 76  
redirect query history to file 76  
redirect query results to file 77  
redistribution  
  in EXPLAIN plan 225  
reduce function in MapReduce 98, 104  
remove a file from SQL-MapReduce 144  
remove command in ACT 140  
remove dead tuples 168  
RENAME a table 159  
RENAME COLUMN 159  
repartitioning 375  
  in EXPLAIN plan 225  
replicate  
  defined 375  
replicated dimension table 375  
replicated table 31  
replication 190  
  best practices 190  
  defined 368  
replication factor, current 376  
replication factor, goal 376  
reserved names 150  
Resource Allocation 271  
results 77  
  send to file 77  
role 246  
  administrator 247  
  catalog\_admin 247  
  default roles in Aster Database 246  
row  
  transpose columns and rows 80  
RowFunction in SQL-MapReduce 98, 104  
Ruby functions in Aster Database 117  
running SQL scripts 77  
  with \i 77

with -f 64

## S

sample code 97  
  SQL-MapReduce Java examples bundle 97  
  tokenize Java example 99  
schema  
  default 263  
  designing 176  
  list all schemas in Aster Database 79  
  search path 263  
schema search path 263  
schema, defined 376  
SCRIPT clause in stream function 118  
scripts 77  
  run SQL script with \i 77  
  run SQL script with -f 64  
SDK 82  
  C and C++ 104  
  Java 97  
  SQL-MapReduce 82  
SDK bundle, Java version 97  
search path 263  
search\_path 263  
security  
  SQL-MapReduce security 138  
segmenting data 29  
send command history to file 76  
send query history to file 76  
send query results to file 77  
server-side cursors  
  in ACT 71  
set command in ACT 76  
setConcurrency.py 270  
Setting Global Limits 268  
shut down  
  deny all users access prior to 270  
single-user mode 270  
SLA 268  
slash-primary directory 242  
slicing data 29  
SMC 376  
soft restart  
  invalidates analytic tables 26  
space 167  
space available  
  reclaim dead space 168  
SQL  
  default login 246  
  parameters that affect SQL command interaction 75  
  parameters that set ACT client behavior 63  
  redirect query results to file 77  
  scripts, running with \i 77

scripts, running with -f 64  
SQL admin user 149, 247  
SQL commands  
    DOWNLOAD FILE 143  
    INSTALL FILE 142  
    UNINSTALL FILE 144  
    utility commands in ACT 75  
SQL prompt commands 75  
SQL prompt, using in ACT 68  
SQL-H 320  
    argument clauses 331  
    benefits of using 321  
    configuring 321  
    datatypes 332  
    limitations 339  
    tips for working with 336  
    troubleshooting 337  
    Using 329  
SQL-MapReduce 82  
    API, C and C++ 104  
    API, Java 97  
    download function 143  
    error logging 115  
    install function 142  
    installing a function 140  
    installing a function, example 111  
    introduction 83  
    list all functions 141  
    listing available functions 140  
    monitoring execution 114  
    naming your functions 143  
    parameters, passing 100  
    remove function 144  
    security 138  
    syntax synopsis 85  
    transactions and 112  
    uppercase letters in function names 143  
    Which functions can I run? 141  
SQL-MapReduce C SDK 104  
    building and packaging a function 106  
    datatypes 109  
    error logging 108  
    example, building 105  
    function, how to write 107  
    getting the SDK 105  
    installing a function 111  
    introduction 104  
    listing available functions 140  
    memory management 109  
    monitoring 114  
    monitoring execution 114  
    naming conventions 108  
    operate function 108  
    passing arguments in the SQL query 108  
syntax synopsis 85  
testing functions locally 110  
transactions and 112  
SQL-MapReduce function names 143  
SQL-MapReduce Java SDK 97  
    building and packaging a function 101  
    constructor for an SQL-MapReduce function 99  
    example Sessionization 103  
    example SplitIntoWords and CountInput 102  
    function, how to write 98  
    operate function 99  
    passing arguments in the SQL query 100  
SQL-MR  
    load\_from\_hcatalog 325  
SQL-MR: See SQL-MapReduce.  
SSH 62  
    SSH client 62  
star schema 21  
stats db 377  
stop all queries 270  
storage utilization 167  
    reclaim dead space 168  
stream function 117  
    installing 140  
streaming API 117  
subdividing data 29  
superuser 246, 247  
    revoke rights 149  
support 16  
system internal users 265  
system tables, defined 377

## T

tab completion 74  
table  
    compress 153  
    create 20  
    creating a dimension table 21  
    creating a fact table 21  
    list all tables in Aster Database 78  
    logical partitioning 34  
    logical partitioning with INHERITS 389  
    replicated vs. partitioned 31  
    replication 190  
    types of 21  
table constraints example 393  
table inheritance  
    example 36, 390  
table partitioning  
    example with INHERITS 393  
    example with PARTITION KEY 23  
tables  
    analytic 25

creating 20  
persistence 25  
temporary 25  
tape backup 367  
technical support 16  
TEMP, example TEMP table 156  
temporary table, example 156  
Teradata  
    Spool mode 314  
Teradata connector  
    spool mode 314  
Teradata NoSpool mode 314  
Teradata-Aster connector 292  
TestRunner  
    testing SQL-MapReduce C API functions 110  
text file on Aster Database 140  
tokenize sample 99  
tools  
    SSH client 62  
tranching data 29  
transaction  
    SQL-MapReduce and transactions 112  
tuple 377  
type casting in stream 119  
typeface conventions 16

## U

UNINSTALL FILE 144  
uninstall SQL-MapReduce function 144  
updated documentation 17  
upload  
    install SQL-MapReduce application code 140  
    upload file to Aster Database 140  
    upload SQL-MapReduce function 140  
uppercase letters in a function name 143  
URL 16  
    Aster Support URL 16  
user 246  
    access control 246  
    add 248  
    authentication 246, 252  
    authentication, Active Directory 257  
    beehive 149, 247  
    creating SQL user 149  
    db\_superuser 149, 247  
    default Aster Database admin user 149, 247  
    default Aster Database beehive user 149, 247  
    default users in Aster Database 246  
    deny all users access 270  
    list all users in Aster Database 79  
    permissions for SQL-MapReduce functions 138  
    read-only access 252  
    roles 246

user-defined function  
    SQL-MapReduce 82  
users  
    system internal 265  
utilities  
    ncluster\_backup 342  
    SQL-H (Hadoop) 320  
    SQL-MapReduce 82  
    Teradata-Aster Connector 292

## V

VACUUM  
    usage guidelines 168  
VALUES 157  
    in SQL-MapReduce example 112  
VALUES clause 157  
VAS 257  
version  
    check ACT version 64  
    documentation version 17  
view  
    defined 377  
Vintela 257  
virtual worker  
    defined 377  
    partitioning and 29

## W

w\*z directories 242  
WAL file 377  
width in EXPLAIN plan 225  
WLM 268  
worker  
    defined 378  
worker node  
    defined 378  
workload 268  
    concurrency 270  
    manager page 286  
    predicate attributes, list of 282  
    predicate syntax 284  
    setting up 276  
    Workload Management tab of AMC 276  
Workload Management 271  
workload management 268

## X

x command 80

## Z

zip file in SQL-MapReduce 101

