

Space Manager
with LiveReorg® 8.3

User Guide



Copyright 2017 Quest Software Inc.
ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (www.quest.com) for regional and international office information.




Patents

Space Manager with LiveReorg is protected by U.S. Patents No. 7,805,423 and 7,970,748. Additional patents pending. For more information, go to www.quest.com/legal.

Trademarks

Quest, Quest Software, LiveReorg, and the Quest logo are trademarks of Quest Software Inc. in the U.S.A. and other countries. For a complete list of Quest Software trademarks, please visit our website at www.quest.com/legal. Microsoft, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Oracle is a trademark or registered trademark of Oracle and/or its affiliates in the United States and other countries. UNIX is a registered trademark of The Open Group. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other trademarks are property of their respective owners.

Legend

-  **IMPORTANT, NOTE, TIP, MOBILE, VIDEO:** An information icon indicates supporting information.
-  **CAUTION:** A CAUTION indicates potential damage to hardware or loss of data if instructions are not followed.
-  **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

Space Manager with LiveReorg
User Guide
January 2017

Contents

Contents	3
Smart Start for Space Manager	10
Welcome to Space Manager	10
Use Space Manager to Detect and Resolve Space Usage Problems	10
Smart Start for Space Manager	11
Connect to Databases	12
Working with Multiple Connections	14
Reorganization Method Selection Chart	14
Oracle License Compliance	15
License Key	15
Space Manager User Requirements	16
Server Objects Installation	16
Server Agent Installation	17
Running Space Manager	17
Usage Feedback	18
Collect Statistics	19
The Space Manager Repository	19
Repository Update Job	19
Reorganization Wizards	19
About the Space Manager Repository	20
Repository Update Job	20
Repository-Related Tasks	20
What Is Supported for Statistics Collection	20
Profile DML Activity	22
Smart Start for DML Profiler	22
Generate DML Profile Reports	23
Manage Storage	24
Monitor Storage from the Explorer	24
What You Can Do From the Explorer	24
Monitor Database Storage	25
Monitor Tablespace and Object Storage	26
Advanced Search	28
Monitor Datafile Storage	29
View Recycle Bin and Free Space	30
Move Objects to Tablespace	31
Reclaim Space	32
View Reorg Need	32

How Reorg Need Is Calculated	33
Reorg Need Factors	34
Reorg Need and Statistics Collection Methods	35
Reorg Need After Reorganization	35
Manage Storage with Object Properties	35
What Is Supported for Object Properties	36
Open Object Properties	37
View Object Properties	37
View Partition Information	37
View Size and Statistics	38
View Extents	38
View Column Definitions	39
View Constraints	39
View Dependencies	39
View Triggers	40
View DDL	40
Manage Storage with Tablespace Properties	40
Open Tablespace Properties	40
Create Tablespaces	40
Alter Tablespaces	43
Tablespace Properties Descriptions	45
Resize Datafile	48
Reorganize Objects with Reorg Manager	50
About Reorganization Wizards	50
Reorg Manager	50
Partitioning Wizard	51
Move to Tablespace Dialog	51
Reclaim Space Dialog	51
Keys Options in Reorganization Wizards	51
Identify Objects That Need to Be Reorganized	52
Move Objects to Tablespace	53
Reclaim Space	53
Resize Datafile	54
About Standard Reorg with Reorg Manager	55
Advantages of Standard Reorganization	55
What is Not Supported for Standard Reorganization	55
About Live Reorg with Reorg Manager	56
What is Not Supported for Live Reorganizations	56
Concepts and Terms	58
Best Practices for Live Reorganizations	59
Switch Mode	62
Change Approval Method for Online Scripts	63
Approve the Switch	64
Getting Started with Reorg Manager	64
Standard Reorganization	65
Live Reorganization	65
Smart Start for Standard Reorganizations	65
Smart Start for Live Reorganizations	67

Launch Reorg Manager	69
Specify Options - Reorg Manager and Partitioning Wizard	70
Select Global Reorganization Options	70
Select Scripting Options	72
Customize Object Allocations	75
Choose a Scheduling Method (LiveReorg)	78
Schedule Script Execution in Wizards	79
Schedule Script Generation	80
Save Reorg Manager Settings	81
Run Scripts from the Editor	82
More Options for Reorg Manager	82
About Selecting Target Tablespace	82
Target Tablespace Considerations	84
About Data Movement Methods	86
More About Scripting Options (LiveReorg)	86
Select a Partition Reorganization Method	89
Set Storage Strategies	90
About Statistics Collection Options	91
Review Space Usage	91
Respond to Checks for a Posting Index	93
Partition Tables with the Partitioning Wizard	95
Requirements for the Partitioning Wizard	95
How Live Reorganizations Are Performed	95
Smart Start for the Partitioning Wizard	96
Launch Partitioning Wizard	98
How to Launch the Partitioning Wizard	98
Objects Not Supported by the Partitioning Wizard	98
How the Wizard Streamlines Partitioning	99
Graphic Features in the Wizard	99
Automated Features in the Wizard	99
What is Not Supported for the Partitioning Wizard	100
Key Concepts for the Partitioning Wizard	101
Partitioning Wizard Basics	101
Partition and Subpartition	101
Range Partitioning	102
Hash Partitioning	102
List Partitioning	102
Composite Range-Hash Partitioning	103
Composite Range-List Partitioning	103
Partitioning Key	104
Local Indexes	105
Global Indexes	105
View Partitioning Schemes	105
Specify Partitioning Method	105
Specify Partitioning Key Columns	107
Specify Partitioning Parameters	108
Best Practices for the Partitioning Wizard	110
Specify Partitioning Key Values	111

About the Partition Key Values Page	111
About Partitioning Key Values	111
Manually Specify Key Values	111
Best Practices for Partitioning Key Values	112
Key Value Grids for Range-Partitioned Objects	113
Key Value Lists for List Partitioned Objects	114
Respond to Checks for a Posting Index	114
Specify Partition Names	115
Run and Monitor Scripts	117
Run and Schedule Scripts	117
How to Execute a Script	117
Script Execution Choices in Wizards	117
Before Running a Script	117
Schedule Scripts	119
Schedule Script Execution in Wizards	120
Run Scripts from the Editor	121
About Scheduling a Recurring Reorganization	122
Manage Scheduled Scripts	122
Restart Scheduled Scripts	124
Restart for Live and Partitioning Scripts	125
Restart for Standard Scripts	126
Restart Commands	127
Monitor Scripts	127
Job Monitor vs Properties Dialog	128
Monitor Interactive Execution in Editor	129
Use Email to Monitor Script Status and Results	129
Use Live Reorg Shell	129
Using Job Monitor to Monitor Scripts	129
Job Monitor Display Options	131
Using Group and Script Properties to Monitor Scripts	132
Monitor Live Reorganizations	133
Change Approval Method for Online Scripts	135
Approve the Switch	136
Enable Email (SMTP) Notification	136
Enable SNMP Traps Notification	138
Configure Additional Notification Events	138
Monitor Scripts Using Live Reorg Shell	140
Manage and Monitor QSA	143
Monitor Statistics Collection Scripts	144
Manage Scheduled Tasks	145
Run Reports	146
Generate Reorganization Reports	147
Generate Reorganization Candidates Report	150
Generate DML Profile Reports	150
Generate Largest Segments Report	151
Generate Fastest Growing Segments Report	152
Generate Tablespace Growth Report	152
Preview and Export Reports	153

Manage Scheduled Tasks	154
Options	155
Reorg Need Options	155
Repository Schedule Options	156
Explorer Object Lists	156
Reorg Manager Options	157
Partitioning Wizard	157
Partitioning Wizard Defaults	157
Partitioning Key Options	159
Partitioning Estimates	159
Server Agent Options	160
Miscellaneous Options	160
Troubleshooting	162
Troubleshoot Script Execution Errors	162
General Guidelines	162
Common Oracle Errors Found in Script Execution	163
Where to Find Script Logs	164
Status Codes for Script Execution	165
Codes for Script Groups	165
Codes for Unscheduled Scripts	165
Codes for Completed Scripts	165
Codes for Scripts that Are No Longer Running	166
Codes for Halted and Scheduled Scripts	166
Codes for Executing Scripts	167
Codes for LiveReorg Scripts	167
QSA Error Messages	167
QSA Server Agent Errors (20300 – 20359)	168
QSA Server Agent Installation Errors (20360 – 20389)	169
Miscellaneous Errors (20390 – 20399)	172
Script Syntax Errors (20400 – 20499)	174
Script Execution Errors (20500 – 20599)	175
FastCopy Errors (20600 – 20699)	177
Administration Functions	181
License Key	181
Install or Upgrade Server Objects	181
Reinstall Server Objects	183
Uninstall (Remove) Server Objects	183
About QSA Server Agent	184
QSA Server Agent Installer	184
Start or Stop QSA Server Agent	185
Start QSA Using the Admin Function	185
Restart QSA Using the Admin Function	186
Stop QSA Using the Admin Function	186
Start QSA Under Special Circumstances	186
Stop QSA Under Special Circumstances	187
Specify Parameters for the QSA Server Agent	187

QSA Server Agent Parameters	189
SENDMAIL Parameters	191
SNMP Trap Parameters	191
Additional Notification Parameters	191
SEND_LR_COMPLETE Options	192
Locks Reporting Parameters	193
QSA Server Agent Locks Reporting Parameters	194
QSA Server Agent Locking Strategies	195
Wait Mode	195
Wait for No Locks	195
No-Wait Mode	195
Install or Upgrade Server Agent on UNIX or Linux	196
Install Server Agent on Windows	197
Install the Agent on a Local Server	197
Install the Agent on a Remote Server	198
Uninstall the Server Agent	199
Uninstall QSA from UNIX or Linux	199
Uninstall QSA from Windows	199
Oracle Upgrade	200
Change Repository Update Schedule	201
Install or Uninstall Demonstration Data	201
Install Demonstration Data	202
Uninstall Demonstration Data	202
Install Analysis Packages	202
Appendix	204
Use the Find Long LONGs Package	204
Populate the Find Long LONGs table	204
Query the Find Long LONGs table	206
Defragment Tablespaces and Coalesce Free Space	207
Defragment Tablespaces	207
Defragment with an Interim Tablespace	207
Empty a Tablespace Before You Defragment	208
How Live Reorganizations Work	209
Stages of a Live Reorganization	209
Estimate the Time Needed for a Live Reorganization	213
Posting Indexes	214
Collection Triggers and the Collection Table	219
The Posting Procedure and OCI Calls	221
LiveReorg Objects and Their Tablespace	221
LiveReorg Commands	225
Global Operations for Space Manager	227
How Space Manager Uses Locale Settings	228
Requirements and Recommendations for Global Operation	229
How Space Manager Reports Locale Settings	231
Use the SQL Editor for a Certain Locale	232
About Us	234
We are more than just a name	234

Our brand, our vision. Together.	234
Contact Quest	234
Technical Support Resources	234

Smart Start for Space Manager

Welcome to Space Manager

Space Manager with LiveReorg® is a comprehensive solution for achieving optimal capacity and performance in Oracle® databases. It provides the DBA with powerful functionality for making the most of storage resources. Space Manager can be used to collect statistics, visualize space use, detect and resolve space-use problems, identify the benefits of reorganizing tables and indexes, reorganize and resize objects for efficient space use, partition tables and indexes, plan for database growth, and document reorganization activities.

Use Space Manager to Detect and Resolve Space Usage Problems

Here are some of the ways Space Manager streamlines the strategic tasks involved in managing database storage:

Detect Problems

- The **Explorer** window allows you to quickly detect the space-use problems that impact database capacity and performance. Space-usage pie-charts provide a graphic display of space use in your databases and tablespaces, allowing you to visualize how space is allocated and how much is free.

In the Explorer window, you can drill down to information on each tablespace. A tablespace view displays the tablespace segments and includes statistics on space usage, including segment counts and sizes. The **Reorg Need column** reveals which tablespace objects need to be reorganized.
- **Predicted Reorganization Benefit** reports calculate the amount of space you can reclaim by reorganizing specific tables or indexes.
- **Reorganization History** reports document your reorganization activities for a specific time period, which allows you to effectively plan for future reorganizations and extrapolate space savings based on data from past reorganizations.
- **Reorganization Candidates** reports help you identify objects with the highest reorg need.
- **Size and Growth** reports provide information to help you plan future capacity.

Resolve Problems

- **Reorg Manager** automates the reorganization and repair processes that resolve space-use problems.
- The **Partitioning Wizard** lets you easily visualize and specify partitioning schemes that provide better performance and easier maintenance.
- All reorganization, repair, and partitioning modules are designed to deliver maximum benefit with minimum impact. Online reorganization methods allow you to reorganize with little or no downtime. During a live reorganization, applications can remain online and users can continue activity against

tables as they are reorganized. Because partitioning scripts use the live reorganization method, they allow you to perform critical database maintenance without interrupting access to data.

Smart Start for Space Manager

This Smart Start outlines the workflow involved in using Space Manager to manage database space. It identifies key features and displays the order in which these features are typically used.

1. **Run the Find Long LONGs procedure** in order to populate the Find Long LONGs table.

Reorg Manager checks this table for the size of LONG and LONG RAW columns. The size of these columns determines which data movement methods can be used to reorganize their tables. See [Appendix](#) on page 204 for more information.

2. **Run the Repository Update job.**

Make sure the Repository Update job runs at least one time before running a reorganization or attempting to identify objects in need of reorganization. This job updates the reorg need and wasted space information in the Space Manager Repository. See [About the Space Manager Repository](#) on page 20 for more information.

3. **Check space usage** in a database and detect space-use problems using Space Manager storage monitoring tools.

- **Explorer**—The Explorer displays when you open Space Manager allowing you to view space usage and space issues for a database. In addition, you can view storage information for a tablespace, datafile, object, or segment. See [What You Can Do From the Explorer](#) on page 24 for more information.

Right-click a tablespace and select **Tablespace Properties** to monitor and manage tablespace storage properties.

- **Datafile Maps**—Datafile maps help you visualize space use in a datafile. Each datafile map includes a list of the segments with blocking extents. See [Monitor Datafile Storage](#).
- **Predicted Reorganization Benefit report**—The Predicted Reorganization Benefit report shows you the amount of space you can reclaim if you reorganize a selected table or index. See [Generate Reorganization Reports](#) on page 147 for more information.
- **Reorganization Candidates Reports**—These reports produce a list of tables or indexes with the highest reorg need. See [Generate Reorganization Candidates Report](#).

4. **Identify reorg need.** Use the Explorer and Reports to identify reorg need. See [Identify Objects That Need to Be Reorganized](#).

5. **Manage space** in a database and resolve space-use problems with properties dialogs and reorganization wizards. You can redefine extent allocation and resize objects for more efficient space use. You can also relocate or partition objects.

- **Reorg Manager**—Use Reorg Manager to resolve space-use problems for objects from anywhere in a database. Reorg Manager provides two reorganization methods: standard and live (available when the LiveReorg option is licensed). During reorganizations with both methods, you can relocate objects to different tablespaces.

For help determining which reorganization method to use, see [Standard Reorganization](#).

Note: The Live option is only available when the LiveReorg option is licensed.

- **Partitioning Wizard**—Use the Partitioning Wizard to partition, repartition, or unpartition objects and resolve space-use problems during a live reorganization. See [Requirements for the Partitioning Wizard](#) on page 95 for more information.

Note: The Partitioning Wizard is only available when the LiveReorg option is licensed.

- **Object Properties**—Use Object Properties to view storage values for an object. See [Manage Storage with Object Properties](#) on page 35 for more information.
- **Tablespace Properties**—Use Tablespace Properties to create tablespaces, add datafiles to existing tablespaces, and size new and existing datafiles. You can also work from here to alter certain properties for existing tablespaces and datafiles. See [Open Tablespace Properties](#) on page 40 for more information.
- **Move to Tablespace**—Use the Move to Tablespace dialog to relocate an object to another tablespace without launching the Reorg Manager. See [Move Objects to Tablespace](#) on page 53 for more information.
- **Reclaim Space**—Use the Reclaim Space dialog to reclaim space in an object without launching the Reorg Manager Wizard. This method performs a live reorganization using the default settings. See [Reclaim Space](#) on page 53 for more information.

6. Monitor and manage script execution.

Use the Script/Job Monitor to check the status of all scripts that are stored in the Space Manager repository.

Stored scripts are run from the server on a scheduled basis by the QSA Server Agent. They can include reorganization and partitioning scripts.

You can also use the Script/Job Monitor to manage stored scripts and the QSA Server Agent. Script functions allow you to do the following:

- Run, cancel, restart, open, and remove scripts. See [How to Execute a Script and Manage Scheduled Scripts](#).
- Schedule and unschedule scripts. See [Schedule Scripts](#).
- Retrieve script execution logs. See [Manage Scheduled Scripts](#).
- Approve the switch for a live reorganization. See [Approve the Switch](#).

QSA Server Agent functions allow you to start or stop the agent and check its status. See [Manage and Monitor QSA](#).

Connect to Databases

Space Manager allows you to create connection profiles to store database connection information. Each profile is comprised of the service name, user name, and password for a database connection. When you select a connection from the tree-list in the Explorer, these values are entered automatically.

The profiles created from a Space Manager client are for that client only. You can create as many profiles as needed for different databases or the same database. A database can have different profiles for different users.

You can have multiple database connections open concurrently during a Space Manager session.

i Note: Connection information for the current connection is displayed in the Space Manager title bar. Information consists of service name for the current connection and user name of the connecting user.

Before creating a new connection

- **Oracle client must match architecture (bitness) of Space Manager.** Install an Oracle client that matches the bitness of the Space Manager application. For example, 64-bit Space Manager requires a 64-bit Oracle client and 32-bit Space Manager requires a 32-bit Oracle client.
- **Tnsnames file.** Make sure the tnsnames.ora file on the Space Manager client computer contains service names for all databases to be managed with Space Manager.

To create a new database connection


1. Select **File | Add New Connection**.
2. Enter the connection information in the Connection dialog. Review the following for additional information:

TNS / Direct	Select the type of connection to create. TNS —Select if using a TNSNames.ora file. Direct —Select to enter database address information directly, without using a TNSNames.ora file.
Server	(TNS only) Select the TNS name (net service name) for the target database. The field lists all names in the TNSNames.ora file on the client computer. i Note If you are running Space Manager in an active-active OPS or RAC environment, select the service name for the instance where the Quest Server Agent (QSA) is installed.
SID	(Direct only) Enter the SID or service name for the target database.
Host	(Direct only) Enter the host name for the target database.
Port	(Direct only) Enter port number, if other than the default.
User Name	Enter the user name of an Oracle user. The user must have the DBA role or the QUEST_SPC_APP_PRIV role. See Space Manager User Requirements on page 16 for more information.
Password	Enter the password for the Oracle user. i Note: Saved passwords are encrypted.
Profile Name	Enter a name for the profile. This name is used to identify the connection in the Explorer tree-list (left pane of Explorer window).
Oracle Home	Allows you to select the Oracle home to use, if you have multiple Oracle homes specified. Space Manager reads the TNSNames.ora file associated with the selected Oracle home.

3. Click **Add Database** to save the profile. Space Manager adds the connection to the tree view in the Explorer window.

To connect



Use one of the following methods to open a connection:

- Select a database connection in the tree view (left pane) in the Explorer.
- Right-click a database connection and select **Connect**.
- Select a connection and click .

i Note: Space Manager server components must be installed in the database to which you want to connect. See [Install or Upgrade Server Objects](#) on page 181 for more information.

To disconnect

Do one of the following:

- Right-click the connection name in the tree-list and select **Disconnect**.
- Select a connection name in the tree-list and click .
- Select a window that is connected to the database from which you want to disconnect. With this window as the active window, click .

i Note: If you disconnect from a database, all Space Manager windows associated to that database connection will close.

To modify connection properties

- Right-click a connection in the tree-list and select **Edit Connection Parameters**.

Working with Multiple Connections

When multiple connections are open for a Space Manager session, you can select one as the current connection. All functions performed and all windows opened will apply to the database that is set as the current database connection.

To set the current connection (or switch connections)

Do one of the following:

- In the Explorer window, select the database connection in the Explorer tree-list.
- Select a window associated to the database connection.

Reorganization Method Selection Chart

When should you select a certain reorganization method? This depends on the functions you want to use, the objects you want to reorganize, and the version of your Oracle database.

You can use the Predicted Reorganization Benefit report to identify the best reorganization options for an object. The report identifies the benefits of reorganizing selected objects and shows the amount of space you can reclaim by reorganizing these objects. From the report, you can right-click an object to reorganize it. See [Generate Reorganization Reports](#) on page 147 for more information.

When you want to...	Select...		
	Standard Reorg	Live Reorg Online Switch	Live Reorg T-Lock Switch
Resolve space-use problems and adjust storage	•	•	•
Reorganize with minimal downtime		•	•
Reorganize with zero downtime		•	
Defragment a tablespace	•		
Pause a reorg for datafile maintenance	•		

When you want to...	Select...		
	Standard Reorg	Live Reorg Online Switch	Live Reorg T-Lock Switch
Sort data by a certain index	•	•	•
Reorganize partitioned objects	•	•	•
Reorganize individual partitions or subpartitions	•	•	•
Reorganize tables with LOBs	•	•	•
Reorganize tables with LONGs (including LONGs >32KB)	•		•
Reorganize tables with both a LOB and a LONG	•		•
Reorganize IOTs and their IOT overflows	•	•	Partitioned only
Reorganize indexes independently of their tables	•		

Oracle License Compliance

Each new release of Space Manager with LiveReorg is designed to optimize your abilities to administer the Oracle database. If Oracle introduces a new data item, feature, or parameter, and if the Space Manager user community desires that new element, you will likely find new functionality within this product to help you with it.

i Important! By default, Oracle installs and enables some features in the database which may not be covered by your existing license, such as Partitioning, RAC, Advanced Workload Repository (AWR) and Compression to name a few. Space Manager takes advantage of these features if found installed in your database, and use of these features (including use by Space Manager) may increase your Oracle licensing fees. Being properly licensed by Oracle is your (organization's) responsibility.

License Key

After you first install the Space Manager client and try to connect to a database, you are prompted to enter a license key. Once you enter a permanent key, you do not need to enter it again. It is preserved during upgrades. You only need to change or re-enter a license key when:

- Your trial key expires.
- You switch from a trial key to a permanent key.
- You change your licensing agreement with Quest Software.
- You uninstall and then reinstall the Space Manager client.

To update a license key

1. Select **Tools | Admin | License Key**.
2. Enter your permanent or trial license key.

Space Manager User Requirements

This section describes the privileges required to install Space Manager server components, install the Server Agent, and run Space Manager.

Server Objects Installation

To install Space Manager server components, you must have the DBA role and full privileges for the target database. (Do not use SYS or SYSTEM.) Your user name must not require double quotes.

i Note: In Oracle 12c, the server objects and Server Agent must be installed by a common user with DBA privileges on all containers. See [Installation DBA in Oracle 12c](#).

The DBA account that installs the Space Manager server objects (components) owns the objects. This same DBA account must be used to upgrade and uninstall the server components (which include Space Manager objects).

The DBA account used to install the server components should also be used to install the Server Agent.

The privileges needed to run FND_STATS as a non-APPs user are automatically granted during installation of Space Manager server components. Any user with the DBA or QUEST_SPC_APP_PRIV role can run FND_STATS from Space Manager.

Installation DBA in Oracle 12c

In an Oracle 12c database, the server objects and Server Agent must be installed by a common user with DBA privileges on all containers. This account must be created prior to installation of the server objects and Server Agent.

1. In the Oracle 12c root container, using a common user account such as SYS or SYSTEM, create another common user account to be used for installing server objects (and Server Agent).
2. Grant the new common user account DBA privileges across all containers. For example, use a grant statement such as the following:

```
grant dba to c##<user> container = all;
```

Privileges and the O7_DICTIONARY_ACCESSIBILITY Parameter

When Oracle's O7_DICTIONARY_ACCESSIBILITY initialization parameter is set to FALSE for a target database, the following requirements must be met:

- The installation DBA must have the EXECUTE on SYS.DBMS_SYS_SQL privilege. During installation, the installation DBA is prompted to log in using a SYSDBA account and then run scripts to grant this privilege.
- If the target database does not allow remote login by users with the SYSDBA privilege, the installation DBA must have remote database administration capability. This capability must be granted prior to installation.
 1. First, create an authentication password file for the target database.
 2. Second, set the REMOTE_LOGIN_PASSWORDFILE parameter to EXCLUSIVE for the DBA account you plan to use as the user for Space Manager server components.

Server Agent Installation

The Server Agent must be installed by user accounts with sufficient privileges. The agent must be upgraded, configured, and uninstalled by its installation account. On UNIX and Linux servers, the agent is installed by an operating system account and an Oracle DBA. On Windows servers, the agent is installed by an Oracle DBA.

i Note: In Oracle 12c, the server objects and Server Agent must be installed by a common user with DBA privileges on all containers. See [Installation DBA in Oracle 12c](#).

UNIX and Linux Privileges

To install the agent on UNIX or Linux, you must have READ, WRITE, and DELETE privileges for the installation directory. During installation, the user connects to the database server host using SSH. The user needs privileges similar to the operating system user that installed Oracle.

Oracle Privileges

To install the agent on Oracle, you must have the UNLIMITED TABLESPACE system privilege and full privileges for the target database. (Do not use SYS or SYSTEM.) To install the agent on Windows, you must have all permissions for the agent installation folder.

On all supported platforms, the installation DBA must have the EXECUTE privilege on:

- DBMS_PIPE
- DBMS_SQL
- DBMS_SYS_SQL
- DBMS_LOB

Normally, EXECUTE privileges are granted as part of the DBA role or during Server Agent installation. If the privileges cannot be granted during installation or if they are revoked, the agent does not run correctly. To fix the problem, grant the missing privileges to the installation DBA.

The installation DBA connects to the target database when QSA is installed. It also automatically connects the agent to the database each time the system is started.

i Note: Use the same DBA account to install Space Manager server components and the Server Agent. The same account is automatically used when you run the Server Agent Installer from Space Manager immediately after server components are installed.

Running Space Manager

Space Manager can be run by DBA users and by non-DBA users that have the QUEST_SPC_APP_PRIV role. This section describes how to grant the QUEST_SPC_APP_PRIV role from Space Manager.

To grant privileges from Space Manager

To grant privileges from Space Manager, the installation DBA runs the following procedure in the SQL Editor (in the procedure, 'USER NAME' is the user name of the user being granted privileges):

```
Begin
    quest_spc_user_manager.grant_user ('USER NAME');
End;
```

The procedure should be run for non-DBA users that will run Space Manager. The procedure grants the QUEST_SPC_APP_PRIV role and additional privileges to non-DBAs. It grants only the QUEST_SPC_APP_PRIV role to non-installation DBAs.

i Note: Privileges must be granted after each upgrade, as well as after the initial installation.

Usage Feedback

Enabling usage feedback allows Quest Software to gather general application use statistics for Space Manager with LiveReorg. Usage feedback can be used by the Space Manager development team to better address the needs of the Space Manager user community.

To enable usage feedback

1. Select **Help | About**.
2. Select the Usage Feedback tab.
3. Select the **Enable usage feedback** checkbox.

To disable usage feedback

1. Select **Help | About**.
2. Select the Usage Feedback tab.
3. Clear the **Enable usage feedback** check box.

Collect Statistics

Storage statistics are essential to tracking space use in a database. Space Manager's primary method for collecting storage statistics is the daily job that updates the Space Manager Repository. In addition, Space Manager also provides options in the Reorg Wizards for automatically collecting statistics after objects are reorganized, repaired, or partitioned.

Statistics are used by Space Manager's problem-detection, problem-resolution, and capacity-planning modules. All provide the best results when statistics are current. Most reports require at least one statistics collection for database objects.

To see the date the Repository was last updated, select a database in the Explorer tree-list. The *Last Update* date is displayed in the database information in the Explorer window.

The Space Manager Repository

The Space Manager Repository collects and stores statistics that are used by Space Manager to calculate reorg need and wasted space and to display space usage information in the Explorer, Object Properties, the Reorg Wizards, and the reports.

See [About the Space Manager Repository](#) on page 20 for more information.

Repository Update Job

The Space Manager Repository is updated by the Repository Update job. By default, this job is scheduled to run daily. You can modify the Repository Update job's schedule (**Tools | Admin | Change Repository Update Schedule**).

This job collects statistics from the Oracle data dictionary and from running Oracle procedures (like those in the DBMS_SPACE package). These statistics are then used by Space Manager to display the list of objects in the Explorer, to estimate table sizes, to generate trend reports, to calculate reorg need and wasted space, and to provide a variety of storage and space use information throughout the Space Manager interface.

See [Change Repository Update Schedule](#) on page 201 for more information.

Reorganization Wizards

The Reorg Manager and the Partitioning Wizard provide options that let you automatically update statistics to reflect changes made during reorganization, repair, or partitioning.

These Reorganization Wizards also provide options that let you preserve existing statistics in the Oracle data dictionary. You might want to preserve statistics if you are using pre-set statistics for Oracle's cost based optimizer.

For more information about the **Analysis** options provided in the Reorganization Wizards, see [Select Scripting Options](#) and [About Statistics Collection Options](#).

About the Space Manager Repository

The Space Manager Repository collects and stores statistics. The statistics stored in the Repository are used by Space Manager to calculate reorg need and wasted space. Reorg need and wasted space are displayed in the Explorer, used in generated reports, and used by the Reorganization Wizards.

In addition, statistics from the Repository are displayed in the Explorer and in Object Properties.

There is one repository for each database where Space Manager is installed. An unlimited number of collections can be accumulated in the repository for each object. When multiple collections are made in one day, only the last collection is kept.

Repository Update Job

Statistics are collected by the Repository Update job that is scheduled to run daily, by default. You can modify the frequency at which this job runs, if necessary. See [Change Repository Update Schedule](#) on page 201 for more information.

This job collects statistics from the Oracle data dictionary and from running Oracle procedures (like those in the DBMS_SPACE package). These statistics are then used by Space Manager to display the list of objects in the Explorer, to estimate table sizes, to generate trend reports, to calculate reorg need and wasted space, and to provide a variety of storage and space use information throughout the Space Manager interface.

Accumulated statistics provide the historic information needed to track growth rates over time and project future space needs. If statistics have never been collected for an object, zero growth is assumed for that object.

Considerations

- If the statistics copied from the data dictionary are pre-set statistics for the cost-based optimizer, be aware that they may affect calculations by Space Manager modules. Space Manager's reorganization modules and reports provide the best results when current storage statistics are available in the repository.
- If an object does not have statistics in the Oracle data dictionary, statistics are not copied to the Space Manager repository for that object.

Repository-Related Tasks

See the following topics for more information about Repository-related tasks:

- [Change Repository Update Schedule](#)—Learn how to change the frequency at which the Repository is updated.
- [Explorer Object Lists](#)—You can select the method Space Manager uses to gather and display the list of objects in the Explorer.
- [Repository Schedule Options](#)—You can specify the number of segments updated during each Repository Job run.

What Is Supported for Statistics Collection

Supported

- Non-partitioned tables and indexes
- Partitioned tables and indexes

- Table partitions and subpartitions
- Index partitions and subpartitions
- Clusters, clustered tables, cluster indexes, indexes on clustered tables
- Most user-defined datatypes

Not Supported

- IOT overflows
- IOT overflow partitions
- LOBs
- LOB partitions and subpartitions
- LOB indexes
- LOB index partitions and subpartitions
- Nested tables
- Object tables
- Rollback segments
- Cache segments
- Temporary segments
- Unknown segment types

Profile DML Activity

Smart Start for DML Profiler

You can use the DML Profiler to profile SQL going against a table you want to reorganize so that you can plan when to start and switch a table reorg. This Smart Start provides an overview of all the major steps involved in profiling DML activity.

To create a DML Profile

1. In the Explorer, right-click a table and select **DML Profile | Create/Modify Collection**.
2. The Create DML Activity Collection dialog opens to the **Parameters** tab.
If you want to add objects, select the **Select Objects** tab. Select additional objects to include and click **Add to List**.
3. In the Parameters tab, specify collection parameters. Review the following for additional information:

Duration	Enter then number of days, hours, minutes, and seconds that you want to collect activity.
Cycle Length	(Advanced) Enter the number of seconds that elapse between two queries on the collection table.
Cycles per Collection	(Advanced) Enter the number of cycles between two updates on the collection statistics table.
Long Running Transaction Cycles	(Advanced) Enter the number of cycles that can elapse before a transaction is considered long running.
Schedule	Select whether to run activity collection instantly, save the script to run at a later time, or schedule the collection script to run on a specific date and time.

4. Click **Save**.


To stop a currently-running DML Profile script

- In the Job Monitor window, right-click the script and select **Cancel**.


To modify the script's collection duration

1. In the Explorer tree-list, select the database that contains the object.
2. Expand the **DML Collection Profile** node for that database.
3. Select the owner of the object. The DML Profile Collections window opens and lists all the profiles for that owner.
4. Right-click the profile and select **Modify Collection**.

To reschedule a script

- In the Job Monitor window, right-click the script and select **Schedule** or click .

To unschedule a scheduled script

- In the Job Monitor window, right-click the scheduled script and select **Unschedule** or click .

To generate a DML Profile report

- In the Explorer window, right-click an object and select **DML Profile | DML Profile Report**.

i Tips:

- You can modify or delete a collection if the collection script is *not* running.
- An object can have only one active DML collection at a time. An active collection is a running or scheduled collection. An object can have multiple completed collections.

Generate DML Profile Reports

To generate a DML Profile report

1. In the Explorer, right-click an object or segment and select **DML Profile | DML Profile Report**.
2. Select a **From** and **To** date at the bottom of the report, if needed.
3. Select each of the tabs to view report details.

i Tips:

- To create a DML Profile, right-click an object or segment and select **DML Profile | Create/Modify Collection**. See [Smart Start for DML Profiler](#) on page 22 for more information.
- To open the DML Profile Collections window for an owner in the selected database, expand the DML Profile Collection node in the database tree-list and select the owner.

Export a report

You can export a report to one of a number of different file formats, such as PDF or Excel. See [Preview and Export Reports](#) on page 153 for more information.

Manage Storage

Monitor Storage from the Explorer

The Explorer provides a comprehensive overview of space use throughout a database—at the database, tablespace, datafile, object, and segment levels. You can drill down from the database level to the tablespace and object levels.

Pie-chart-style graphs let you see at-a-glance how much space is allocated and how much space is free at the database and tablespace levels. The Properties and the Statistics panels, as well as columns in the Segments (objects) grid, provide numeric information on space usage, various counts, and object sizes.

i Note: The Explorer displays automatically when you launch Space Manager. If you have multiple connection profiles, each connection is displayed in the Explorer tree-list (left pane).

In databases that are Oracle 10g or higher, Recycle Bin information is displayed. At the database and tablespace levels, the Explorer reveals how much space is being used by dropped objects. This lets you see how much can be reclaimed for use by other objects.

What You Can Do From the Explorer

Use the Explorer to view storage information. In addition, the Explorer serves as the central launch point for Space Manager functionality. Many functions are available from the right-click menu when you select a database, tablespace, or segment in the Explorer.

From the Explorer window you can perform the following tasks:

- **Display information**—Display storage and space use information for a database, a tablespace, a datafile, or a segment.
 - See [Monitor Database Storage](#) on page 25 for more information about displaying database information.
 - See [Monitor Tablespace and Object Storage](#) on page 26 for more information about displaying tablespace and object/segment information.
 - See [Monitor Datafile Storage](#) to learn how to view the Datafile Map, segments and reorg need information.
 - See [View Recycle Bin and Free Space](#) on page 30 for more information about managing the Recycle Bin and free space.
- **Search for segments/objects**—Use simple or advanced search to find objects.
 - Use the **Search** box to search for objects, segments, and partitions by name in the Explorer. See [To use the Search field](#).
 - Use the **Advanced** search feature to use multiple search criteria to find objects. See [Advanced Search](#).
- **View and Identify reorg need**—Use the Reorg Need column in the Segments/Objects grid and the Datafile Map to identify objects in need of reorganization. See [View Reorg Need](#).

- **Launch Reorganization Wizards**—Launch the Reorg Manager or the Partitioning Wizard by right-clicking one or more selected objects in the grid. See [Launch Reorg Manager](#) and [Launch Partitioning Wizard](#).

In addition, you can launch other Space Manager tools from the Main menu. Actions you perform are applied to the currently-selected database in the Explorer.

- **Generate a report**—Use the Main menu to open the Reports window and generate a report for the selected database.
- **Open the Script/Job Monitor**—Use the Main menu to open the Script/Job Monitor for the selected database.
- **Open the Options dialog**—Use the Main menu to open the Options dialog and specify options for the selected database.

i Tip: You can drag and drop databases in the Explorer tree-list to rearrange the list.

Monitor Database Storage

You can use the Explorer to monitor database storage. Two views are available to display database storage information for each database.

- **Database view**—In the Explorer, select a database in the left pane to display database information in the right pane. A summary of the database properties and statistics displays, as well as a pie chart showing total database space usage.

i Note: Clicking a database name in the left pane also opens the connection.

- **Tablespaces view**—In the Explorer, click the **Tablespaces** node under a database to display a list of tablespaces. A tablespaces grid displays in the right pane listing the tablespaces included in the database, as well as the total sizes and counts for various segment types for each tablespace.

To view a summary of database properties and statistics

Select a database in the Explorer tree-list. The Properties, Statistics, and Graph panels display.

- **Properties**—Lists the database properties.
- **Statistics**—Lists database total size and free space, and total counts and total sizes for tables, indexes, LOBs.
- **Graph**—Displays a pie chart that lets you visualize space-use percentages by segment type for the entire database.

i Note: Status information for the Repository and for the Quest Server Agent is also displayed above the Properties panel.

To view the Tablespaces list and summary information

Select the **Tablespaces** node in the Explorer tree-list. A summary of tablespace information, as well as a list of tablespaces, displays in the right pane.

- **Summary**—Lists total number of tablespaces, grouped by tablespace category.
- **Tablespaces**—Displays a list of tablespaces and a grid which displays storage information for each tablespace in the database. The grid lists the total size and total count for tables, indexes, and LOBs for each tablespace.

Allocation Chart—The Allocation Chart column is a visual presentation of the amount of allocated and free space for each tablespace relative to the largest tablespace. Use the charts to easily compare tablespace sizes. Also use the charts to compare free to allocated space for each tablespace.

i Note: For Oracle 10g (or later) databases, recycle bin space allocation information is included at the database and tablespace level.

i Tip: You can search for objects/segments by name using the **Search** text box in the upper-right corner of the Explorer window.

Top Issues




When you select a database in the Explorer tree-list, Space Manager displays a list of the top issues for that database. The Issues list displays in a pane below the database summary information.

For each issue, the list provides an icon for easy identification of issue severity, the object type, and a description of the issue.

You can use the Issues list to identify objects in need of reorganization and launch the Reorg Manager directly from this list.

To view the Top Issues for a database

1. Select a database in the Explorer tree-list. The Issues list displays in the right-pane (below the summary information).
2. Review the list of top issues. Click a column heading to sort by the column. Icons identify the most urgent issues. Review the following icon definitions:

Icon	Description
	Severe issue. Indicates a severe issue, such as a tablespace that is very low on free space.
	Warning. Indicates an issue that might need attention, such as a tablespace that is moderately low on free space or a table that has wasted space.
	Free space. Indicates the tablespace has a large amount of free space. Consider releasing this free space back to the file system.

To launch the Reorg Manager from the Top Issues list

- Right-click an object in the list and select **Reorg Manager**.

Monitor Tablespace and Object Storage

In the Explorer, from the database level, you can drill down to storage information for individual tablespaces and their segments or objects. To select a tablespace, click the tablespace name in the Explorer tree-list.

i Note: Read-only tablespaces are displayed with the international “no” symbol. Offline tablespaces have a red center.

To view a summary of tablespace information

1. Select a tablespace in the Explorer tree-list. The Properties, Statistics, and Graph panels display above the Segments grid.

Properties	Statistics	Graph
Tablespace QUEST_SPC_DEMO_TS_01	Total Size: 20.00 Mb	
Status: Online	Free Space: 2.75 Mb	
Contents: Permanent	Table Segments: 35 12.00 Mb	
Extent Management: Local	Index Segments: 33 2.00 Mb	
Allocation Type: Autoallocate	LOB Segments: 7 0 Kb	
Segment Management: Auto	Recycled: 0 0 Kb	
Datafile Type: Normal	Other: 0 0 Kb	
Block Size: 8 Kb	Datafiles: 1	
	Autoextensible Datafiles: 0	

- **Properties**—Lists the tablespace properties.
- **Statistics**—Lists tablespace total size and free space, and total counts and total sizes for tables, indexes, LOBs.
- **Graph**—Use the graph to quickly visualize how space is allocated and how much is free.

i Note: For Oracle 10g (or later) databases, recycle bin space allocation information is included in the tablespace summary.

To list tablespace segments

1. Select a tablespace in the Explorer tree-list. The Segments list displays below the Properties, Statistics, and Graph panels.
2. The Segments list displays the following information for each segment in the tablespace.
 - Segment owner
 - Segment name
 - Partition name
 - Segment type
 - Sub-type
 - Wasted space and space allocated
 - Reorg need

i Note: Objects that do not use storage are not displayed in the Segments grid. Instead, their components that do use storage are displayed. For example, partitions and subpartitions are displayed instead of their parent partitioned object. IOT indexes are displayed instead of IOTs.

To view Reorg Need

The Reorg Need column provides the reorg need value calculated by Space Manager for each object in a grid. See [View Reorg Need](#) on page 32 for more information.

- Click the **Reorg Need** column header to sort objects by reorg need.
- Select *Reorg Need* in the **Group By** field to group objects by reorg need.
- To modify Reorg Need default values, go to **Tools | Options | Reorg Need**. See [Reorg Need Options](#) on page 155 for more information.

To view objects by Owner

1. In the Explorer tree-list, for a selected database, expand the **Owners** node.
2. Select an owner and expand the node.

3. Click **Tables** or **Indexes** to display the objects owned by that user.

i Tip: To sort objects by reorg need, click the **Reorg Need** column header in the object grid.

To use the Search field

You can search for an object, segment, or partition by name. Enter a partial or whole word to find a match. The field uses incremental search. All names that include the string entered are returned.

Space Manager also provides an advanced search feature. See [Advanced Search](#).

1. In the Explorer tree-list, select a database, a tablespace, or an owner to narrow your search to that group of objects.
2. In the **Search** text box in the upper-right corner of the Explorer window, enter a whole or partial name.
3. Space Manager uses incremental search to find matching segment, object, and partition names. Non-whole-word matching is used to return names that contain a match in any segment of the name.

To use the Group By field

1. In the Explorer tree-list, select a tablespace (or select an owner and then tables/indexes).
2. In the **Group by** field, select the column to group by.
3. After grouping objects/segments, view the total allocation for each group in the **Allocation** column.
4. Review these tips:
 - If grouping by Reorg Need, you can modify the threshold values for High, Medium, and Low Reorg Need in **Tools | Options | Reorg Need**. See [Reorg Need Options](#).
 - Click a column header to sort by that field before grouping. After grouping, the sort order is preserved within each group.

To launch the Reorg Manager or Partitioning Wizard

1. Select one or more objects in a segments or objects grid.
2. Then right-click and select **Reorg Manager** or **Partitioning Wizard**.




i Tips:

- To display the Object Properties window, right-click an object in the grid and select **Object Properties**. See [Manage Storage with Object Properties](#) on page 35 for more information.
- To display the Tablespace Properties window, right-click a tablespace name in the Explorer tree-list and select **Tablespace Properties**.
- To select multiple objects to reorganize simultaneously, use a combination of selected node in the tree-list and the **Search** and **Group By** fields to find and select the objects.
- To view recycle bin information, click on the **Recycle Bin** icon under a tablespace node. See [View Recycle Bin and Free Space](#) on page 30 for more information.
- To view free space information, click on the **FreeSpace** icon under a tablespace node. See [View Recycle Bin and Free Space](#) on page 30 for more information.

Advanced Search

The Space Manager Advanced Search feature allows you to use multiple search criteria to find objects. This is useful when you want to find a diverse set of objects to include in one reorganization script group. Search for objects within a database or a tablespace. Access the Advanced Search feature from the Explorer window.

To use Advanced Search

1. In the Explorer tree-list, select a database, tablespace, or owner to search. You can also select the Tables or Indexes node under an owner.
2. Click the **Advanced** button beside the Search box in the Explorer window to open the Advanced Search window.
3. Use the fields provided to create one or more conditions to use as your search criteria. Select an operator and then select or enter a value in each applicable field.
4. After specifying search criteria, click **Start New Search**. The search results are displayed in the lower pane.
5. Click the **Show search text** button to display the text of your search condition/conditions.
6. To refine your search, you can modify criteria or add additional criteria and click **Start New Search**.
7. To add to your current search results, you can add one or more new conditions (or change values in existing conditions) and then click  and select **Add to results**. The new or modified conditions are used to select objects to add.
8. To remove objects from the current results, you can add one or more new conditions (or change values in existing conditions) and then click  and select **Remove from results**. The new or modified conditions are used to select objects to remove.
9. To search within the current results, add or modify conditions and then click  and select **Search within results**.
10. Review the following additional instructions:
 - When using the *in* or *not in* operators, enter search values as a comma-separated list.
 - To perform a case-sensitive search, surround the value in double quotes. Unquoted text values are converted to uppercase before performing a search.
 - Empty fields are not included in the search criteria.
 - In the **Group by** field, select a column to group results by column values.

To launch the Reorg Manager or Partitioning Wizard from search results

1. Select one or more objects in the search results grid.
2. Then right-click the selection and select **Reorg Manager** or **Partitioning Wizard**.

Monitor Datafile Storage

In the Explorer, you can drill down from the tablespace level to view storage information for each datafile. The datafile map helps you to visualize space usage for a datafile.

To display datafile storage and datafile map

1. Select a tablespace from the Explorer tree-list.
2. Expand the tablespace node. Then expand the Datafiles node to display a list of datafiles.
3. Select a datafile from the list. The following information is displayed:
 - **Summary information**—Size, free space, maximum size (a size is displayed only if the datafile is auto-extensible), and reclaimable space.
 - **Segments with blocking extents**—A list of segments with segment type, name, and owner.

- **Datafile map**—A color-coded graphic representation of datafile space use. Segments/extents are color-coded to easily view space usage by segment type. Hover over each color-coded segment/extent to display the segment name. Click on an extent to highlight all extents in the segment.

i Note: Datafiles are listed by their Oracle-assigned datafile number.

4. Use the datafile map to view storage information:

- Hover over each cell to display the names of all segments in that cell.
- Click on a cell in the map to highlight all extents allocated to that segment.

5. Toggle between two views: *Segment Type* and *Reorg Need*.

- **Segment Type**—Select this view to color-code segments by segment type. Use the legend to identify segment type.
- **Reorg Need**—Select this view to color-code segments by reorg need. Use the legend to identify reorg need severity.

i Note: To modify threshold settings for reorg need severity, select **Tools | Options | Reorg Need**.

6. Identify segments with blocking extents.

- The **Segments with blocking extents** list contains extents that are blocking you from resizing the datafile.
- Click on a segment name in the list to highlight all extents within that segment in the datafile map.

i Note: Press **F5** to refresh the datafile map.

To launch the Reorg Manager from Segments list

- In the **Segments with blocking extents** list, right-click a segment and select **Reorg Manager**.

To view a list of datafiles for a tablespace

1. In the Explorer tree-list, select the **Datafiles** node for a tablespace. A list of datafiles for that tablespace displays.
2. Review the list to see storage information details for each datafile, including size, how much space is allocated and how much is free space, and whether the datafile is auto-extensible.

i Tip: You can add a datafile using the Tablespace Properties window (right-click a tablespace name in the tree-list and select **Tablespace Properties**).

View Recycle Bin and Free Space

Available in databases that are Oracle 10g or later, the Recycle Bin is a logical container that is used for dropped tables and their dependent objects. Oracle automatically preserves dropped objects in the Recycle Bin in case you need to recover them. Oracle then treats the space used by these objects as free space. The space can be reclaimed for use by other objects.

In the Explorer, you can view Recycle Bin space and objects for each database and for each tablespace. You can also view information about free space for each tablespace.

To view Recycle Bin space for a database or tablespace

1. In the Explorer tree-list, expand the node for the database or tablespace you are interested in.
2. Click on the **Recycle Bin** icon under the database or tablespace node. The list of segments in the Recycle Bin displays on the right side of the Explorer window.

3. Check the **Allocation** column to see how much Recycle Bin space can be reclaimed for use by other objects.

i **Tip:** Click on a database or tablespace name in the Explorer tree-list to display a pie chart which shows the amount of Recycled space as a percentage of total space.

To view free space for a tablespace

1. In the Explorer tree-list, expand a tablespace node.
2. Click on the **Free Space** icon under the tablespace node. The list of free extents displays on the right side of the Explorer window.

i **Tip:** Click on a database or tablespace name in the Explorer tree-list to display a pie chart which shows the amount of free space as a percentage of total space.

To purge an object from the Recycle Bin

1. Select the **Recycle Bin** node for a database or tablespace.
2. In the Segments pane, right-click a segment (object) and select **Purge** or **Purge object**.

To restore a table from the Recycle Bin

1. Select the **Recycle Bin** node for a database or tablespace.
2. In the Segments pane, right-click a table or table partition and select **Restore table**.

To purge the Recycle Bin

Do one of the following:

- Right-click a **Recycle Bin** node in the Explorer tree-list and select **Purge Recycle Bin**.
- Select the **Recycle Bin** node for a database or tablespace. In the Segments pane, click **Purge Recycle Bin**.

Move Objects to Tablespace

You can easily move an object or segment to another tablespace from within the Explorer without launching the Reorg Manager. You can execute the task immediately or schedule it.

To move an object to a tablespace

1. In the Explorer, do one of the following:
 - Right-click the object or segment and select **Move to Tablespace**.
 - Drag the object from the Segments grid (or Tables or Indexes grid) to a new tablespace in the Explorer tree-list.
2. In the Move Objects to Tablespace dialog, select a **Target Tablespace** from the list.
3. Then select one of the following:
 - **Execute Immediately** to move the object immediately.
 - **Schedule Date** to move the object at a scheduled date and time.
4. Click **Next**.
5. Review the summary and click **Finish** to execute immediately or when scheduled.

i **Tip:** You can also perform this action from the search results in the Advanced Search window. Right-click an object in the search results and select **Move to Tablespace**.

Reclaim Space

You can easily reclaim space in a segment/object from within the Explorer without opening the Reorg Manager Wizard. The Reclaim Space method performs a live reorganization using the default settings.

You can execute the task immediately or schedule the task in the Reclaim Space dialog.

To reclaim space

1. In the Explorer, right-click an object or segment and select **Reclaim Space**.
2. In the Reclaim Space dialog, select one of the following:
 - **Execute Immediately** to reclaim space in the object immediately.
 - **Schedule Date** to reclaim space in the object at a scheduled date and time.
3. Click **Next**.
4. Review the summary and click **Finish** to execute immediately or when scheduled.

i Tip: You can also perform this action from the search results in the Advanced Search window. Right-click an object in the search results and select **Reclaim Space**.

View Reorg Need

Use the Reorg Need column in the Segments/Objects grid and the Datafile Map to identify objects in need of reorganization.

Reorg need is a measure of the benefit of reorganizing an object to resolve space-use problems. Benefit is the improvement in performance to be gained and wasted space to be recovered. The higher the value, the greater the benefit.

i Note: Space Manager provides the best Reorg Need and Wasted space information when statistics are current in the Repository. Be sure to run the Repository Update job on regular or nightly basis (see [Change Repository Update Schedule](#)).

Use the following Explorer features to identify objects with the greatest need for reorganization.

To learn more about the reorg need value, review the following topics:

- [How Reorg Need Is Calculated](#)
- [Reorg Need Factors](#)
- [Reorg Need and Statistics Collection Methods](#)
- [Reorg Need After Reorganization](#)

Explorer - Segments Grid

Each segment/object grid in the Explorer includes a **Reorg Need** column. This column provides the calculated reorg need value for each object in the grid.

To view Reorg Need column

1. In the Explorer, select a tablespace or owner to view reorg need.
 - **Tablespace:** Select a tablespace in the Explorer tree-list (left pane). The grid on the right lists the tablespace segments and includes a **Reorg Need** column.
 - **Owner:** Select an owner in the Explorer tree-list (left pane). Expand the owner node and then select **Tables** or **Indexes**. The grid on the right lists the owner's objects and includes a **Reorg Need** column.

2. Review the Reorg Need column to determine those segments that would benefit from reorganization. Click the column header twice to sort from highest to lowest need.

See [Monitor Tablespace and Object Storage](#) on page 26 for more information.

Explorer - Advanced Search

Use the Advanced Search window to search for objects based on their reorg need value. See [Advanced Search](#) on page 28 for more information.

Explorer - Datafile Map

Use the Datafile Map to view segments with extents that are blocking you from resizing the datafile.

To view segments with blocking extents

1. Select the tablespace for which you want to view a datafile map.
2. Expand the tablespace datafile node and select a datafile. The datafile map displays along with a list of segments with blocking extents.
3. Use the list of segments with blocking extents to determine which segments you want to reorganize.
4. Right-click a segment in the list to launch the Reorg Manager.

See [Monitor Datafile Storage](#) on page 29 for more information.

How Reorg Need Is Calculated

Reorg need is calculated based on factors that have a negative impact on database performance. These factors include wasted blocks and excessive chaining (which force Oracle to perform additional reads).

Which factors are used for an object depends on its type and whether it resides in a tablespace that is data dictionary managed or locally managed. The following factors are used for different object types:

- Wasted data blocks—Used for all objects.
- Chained rows—Used for tables only.

Reorg Need Calculations

To determine reorg need for an object, Space Manager performs the following calculations:

1. Each reorg need factor is given a normalized value between zero and one. For example, if a table has no chained rows, the Chained Rows factor is zero. If all of a table's rows are chained, the Chained Rows factor is one. Factors that are not used in calculating reorg need for an object are given a value of zero.
2. Each reorg need factor is multiplied by a weight. Default weights are 70% for wasted blocks and 30% for chained rows. Default weights add up to 100%.

i Note: You can specify the default weight values in **Tools | Options | Reorg Need**. See [Reorg Need Options](#).

3. All weighted values are added together to arrive at a reorg need value in the range of 1 to 100.

i Note: Statistics for reorg need factors are taken from the Oracle data dictionary and DBMS_SPACE.

Reorg Need Factors

This section describes the factors that are used by Space Manager to calculate reorg need.

Wasted Blocks Factor

The Wasted Blocks factor indicates what percentage of an object's data blocks are wasted. Wasted blocks are the number used beyond the number currently needed for data. These blocks once contained data but are now empty because data has been deleted. The number of data blocks used is the high water mark for an object. This is the point up to which Oracle reads during full object scans. The high water mark rises when data is inserted. It does not go down when data is deleted. (Wasted Blocks is a factor used for all objects.)

Why it contributes to reorg need:

Wasted data blocks degrade database performance because they require Oracle to read more blocks than necessary during full object scans. Reorganizing objects with wasted data blocks resets (drops) the high water mark to the number of data blocks actually needed for data. Oracle then needs to scan fewer blocks for those objects.

Values used:

The following fields display the values used in calculating the Wasted Blocks factor:

- **Blocks Used**—This field displays the highest number of data blocks ever used by an object (its high water mark).
- **Blocks Needed**—This field displays the number of data blocks currently needed by an object for data.
- **Wasted Blocks**—This field displays the number wasted data blocks for an object. It is derived by subtracting blocks needed from blocks used.

Chained Rows Factor

The Chained Rows factor indicates what percentage of a table's rows are chained. Chained rows are rows that have been migrated from their original data block due to data growth. They can also be rows that span multiple data blocks because they are too large to fit into a single data block. The Chained Rows factor takes both types of chaining into account. However, only migrated rows are repaired by Space Manager. (Chained Rows is a factor used for tables only.)

Why it contributes to reorg need:

Excessive chaining degrades performance during queries. This is because chaining forces Oracle to read multiple data blocks to retrieve a single row. First, Oracle retrieves the row's original data block. Then, it uses the pointer in the original block to retrieve the row's new data block. Reorganizing (or repairing) tables with migrated rows reduces the number of data blocks used by these rows. This in turn reduces the data blocks Oracle needs to scan.

Values used:

The following fields display the values used in calculating the Chained Rows factor:

- **Chained Rows**—This field displays the number of chained rows in a table.
- **Total Rows**—This field displays the total number of rows in a table.

Reorg Need and Statistics Collection Methods

Reorg Need Calculations

Reorg Need calculations include two factors, wasted space and chained rows. Wasted Space in a segment is calculated using `DBMS_SPACE.SPACE_USAGE` and is calculated for segments in a locally managed tablespace only.

How Users Can Compensate for Missing Chained Row Statistics

When `DBMS_STATS` or `FND_STATS` is used to collect statistics, you can compensate for missing chained-row statistics by decreasing the weight of the **Chained Rows** factor in reorg need calculations. The default weight is 30%.

To decrease the weight for the **Chained Rows** factor, update the value for **Chained Row Weight** in **Tools | Options | ReorgNeed**. Also, increase the weight for the **Wasted Space Weight** so that the total of all weights is 100.

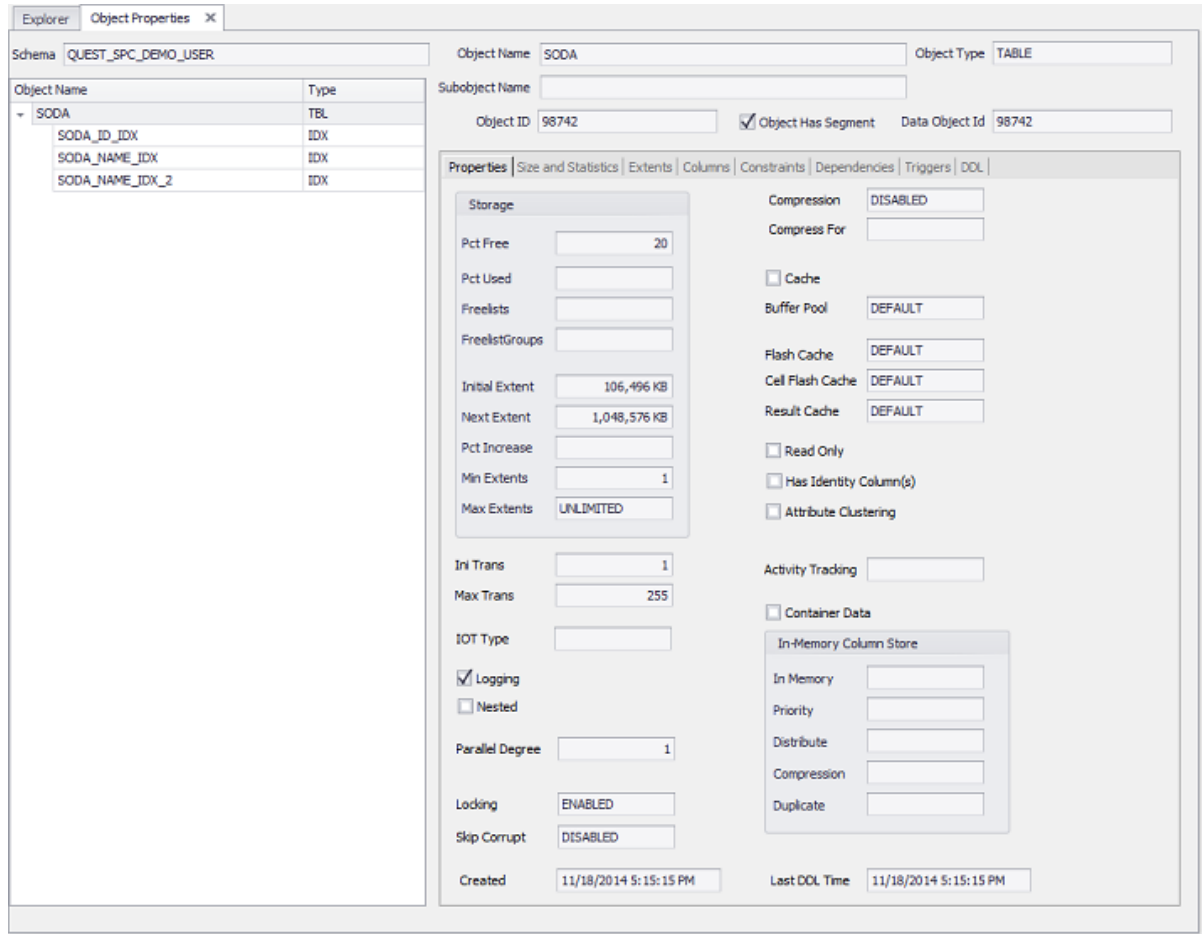
Reorg Need After Reorganization

Normally, values for reorg need and need ratio drop to zero after an object is reorganized and its statistics are updated.

Manage Storage with Object Properties

Use Object Properties to view comprehensive information for an object and its related objects.

To open Object Properties, right-click an object in the Explorer and select **Object Properties**. See [Open Object Properties](#) on page 37 for more information.



What Is Supported for Object Properties

Supported

- Non-partitioned tables and indexes
- Partitioned tables and indexes (range, hash, list)
- Tables with LOBs, LOB partitions, and LOB subpartitions
- IOTs and partitioned IOTs
- IOT overflows
- Clusters, clustered tables, cluster indexes, indexes on clustered tables
- Tables with user-defined datatypes
- Tables with ROWID and UROWID datatypes

Not Supported

- Nested tables
- Object tables
- VARRAYS
- Temporary tables
- Materialized views

- Replication tables; objects that are part of a replication group

i Note: In most cases, Object Properties cannot be opened for objects it does not support.

Open Object Properties

To open Object Properties

1. Right-click an object in the Explorer or in a Predicted Reorganization Benefit report and select **Object Properties**.
2. The object and its related objects are listed in a hierarchical tree view in the left pane. Related objects descend from the top-level object. The objects can include partitions, subpartitions, indexes, LOBs, and IOT overflows. (Object types are represented with various symbols.)

i Note: In most cases, a table is shown at the top of the object tree. However, a cluster is shown at the top when the launching object is a cluster, a clustered table, a cluster index, or an index on a clustered table.
3. View properties for the selected object by selecting the various Object Properties tabs. The tabs available change according to the type of object selected.
4. View properties for a related object by selecting it in the object tree.

i Note: Information at the top of the Object Properties window identifies the object selected in the object tree. The information includes object name, owner, object type, and object ID. Identifying information is displayed for all tabs.

View Object Properties

View the Properties tab in Object Properties to see various attributes for the selected object.

i Note: The panels and fields displayed in the Properties tab change according to the type of object selected. If an attribute is not used for an object, **N/A** is displayed in the attribute field.

Property	Description
Storage Attributes	<p>The Storage panel displays the attributes set with Oracle's storage clause. These include INITIAL, NEXT, PCTINCREASE, MINEXTENTS, and MAXEXTENTS. They determine how extents are allocated for objects.</p> <p>If an object resides in a locally managed tablespace, the values displayed may be different from the values specified when the object was created or reorganized:</p> <ul style="list-style-type: none"> • MINEXTENTS—The value shown for MINEXTENTS is always 1, no matter which allocation type is used for a locally managed tablespace. <p>The Storage panel also displays attributes set with Oracle's physical attributes clause, such as PCTFREE and PCTUSED. INITRANS and MAXTRANS are displayed below the Storage panel. These attributes determine how space is used inside object data blocks.</p>
Additional Attributes	<p>Additional attributes are displayed when they are available for an object. These attributes include logging and caching information.</p>

View Partition Information

The Partitioning tab of Object Properties is displayed when a partitioned object, partition, or subpartition is selected in the object tree. The fields in this tab change according to which of these objects is selected.

View Size and Statistics

View the Size and Statistics tab in Object Properties to see storage statistics for a selected object. This tab is displayed for objects that are supported for statistics collection. It is not displayed for objects such as LOBs.

Along with the Reorg Need value, the Size and Statistics tab can help you determine whether the object would benefit from being reorganized.

The Source for Statistics

The Statistics tab displays statistics from the Space Manager Repository.

Tablespace Statistics

- **Block Size**—This field displays the data block size for an object's database or tablespace (in databases that allow block sizes to vary by tablespace).

Allocated Space

- **Segment Size**—This field displays total space allocated for an object.

Space Manager Statistics

- **Actually Used**—This field displays the amount of space used by an object that currently contains data.

Data Dictionary Statistics

- **Row Count**—This field displays the number of rows in a table.
- **Chained Rows Count**—This field displays the number of chained rows in a table.
- **Average Row Length**—This field displays the average length of rows in a table. The length value includes row overhead.

View Extents

View the Extents tab in Object Properties to see a list of the extents allocated to an object.

Extent Statistics

- **File Name**—File name of the datafile for an extent.
- **Status**—Whether a datafile is available or not.
- **Autoextensible**—Whether a datafile is autoextensible.
- **Extent ID**—Sequence number reflecting the order in which an extent was allocated.
- **Block ID**—Starting block number of an extent.
- **Blocks**—Number of data blocks.
- **Extent Size**—Size in kilobytes.
- **Number of Extents**—This field displays the total number of extents used by an object.
- **Average Size**—This field displays the average size of the extents used by an object.

- **Minimum Size**—This field displays the size of the smallest extent used by an object.
- **Maximum Size**—This field displays the size of the largest extent used by an object.

The maximum number of extents listed is 500.

View Column Definitions

View the Columns tab in Object Properties to see column definitions for a table, index, cluster, clustered table, or IOT. A yellow key symbol identifies columns that are included in a table's PRIMARY KEY.

The Columns tab is not displayed for partitions and subpartitions.

i Note: If a table contains a user-defined datatype or XMLType datatype, the Columns tab displays all hidden columns associated with that datatype.

View Constraints

View the Constraints tab in Object Properties to see a list of all integrity constraints defined for a table. This tab displays Primary Key, Unique, Foreign Key, Referencing Foreign Key, Not Null, and Check constraints.

The information displayed for each constraint includes deferrable state, status (enabled or disabled), and whether or not it is validated.

Constraints are grouped by type under type nodes. To collapse or expand a single type group, click - / + beside its node.

i Note: The Constraints tab is not displayed for table partitions and subpartitions. To see the constraints used for these objects, select their parent table in the object tree. The constraints defined for a parent table are used for all of its partitions or subpartitions.

Constraint Fields

- **Column Name**—This field or table displays a list of all columns that define a constraint. If the column names are not visible, click + to expand the constraint name node.
- **Foreign Key Constraint Fields**—Information includes the name of the constraint referenced by a FOREIGN KEY, the table the constraint is on, the owner of the table, and the action defined for the referenced constraint.
- **Referencing Foreign Key Constraint Fields**—Information includes the name of the referencing table, the owner of the referencing table, and the action defined for the referenced constraint.
- **Check Constraint Fields**—Information consists of the check condition defined for the selected constraint.

View Dependencies

View the Dependencies tab in Object Properties to see a list of dependencies that reference a table. This tab displays triggers, procedures, views, materialized views, and packages.

The information displayed for each dependency includes dependency name, owner, type, and status. To see more information on triggers, view the Triggers tab.

View Dependency Trees

If a dependency is referenced by other dependencies, the dependencies are displayed in a hierarchical tree. This lets you see dependency relationships at a glance, saving the time it would take to trace dependencies that are many levels deep.

Dependencies that reference the selected table are displayed at the top level of the dependency tree. Dependencies that reference a table dependency descend from that dependency in the second level of the tree. Dependencies that reference second-level dependencies descend from them in the third level, and so on.

View Triggers

View the Triggers tab in Object Properties to see a list of the triggers defined for a table. The information displayed for each trigger includes trigger name, owner, trigger type, trigger firing event, and status.

To view the trigger description and the trigger body, select the trigger in the list. The description and body are displayed in the lower pane.

View DDL

Use the DDL tab of Object Properties to view the DDL statement for the selected object.

To view Object DDL statement

1. Right-click an object in the Explorer window and select **Object Properties**.
2. In the Object Properties window, select the **DDL** tab.
3. In the DDL tab, click **Get DDL**.

Manage Storage with Tablespace Properties

Open Tablespace Properties

The Tablespace Properties window allows you to create a new tablespace or modify an existing tablespace.

- See [Create Tablespaces](#) to learn how to create a new tablespace.
- See [Alter Tablespaces](#) to learn how to modify an existing tablespace.
- See [Tablespace Properties Descriptions](#) for detailed descriptions of tablespace property options.

To open Tablespace Properties for a new tablespace

1. Right-click any tablespace in the Space Manager Explorer.
2. Select **New Tablespace**.

To open Tablespace Properties for an existing tablespace

1. Right-click a tablespace in the Explorer or a report.
2. Select **Tablespace Properties**.

i **Tip:** To increase or decrease the size of a datafile, see [Resize Datafile](#).

Create Tablespaces

Before you create a new tablespace, estimate the average size and growth rate of the objects it will contain. Datafiles in the tablespace should be sized to provide space for growth in the near future.

See [Tablespace Properties Descriptions](#) for detailed descriptions of the tablespace property options.

To create a tablespace with *Tablespace Properties*

1. Right-click any tablespace in the Space Manager Explorer and select **New Tablespace** to open *Tablespace Properties*.
2. Enter a name for the tablespace in the **Tablespace** field. The default name is **NEW_TABLESPACE_0**. Using a standard name is recommended. The maximum number of characters allowed is 30.
3. In the **Status** panel, select the **Online** checkbox if to create an online tablespace. Its objects can be accessed by users and applications. Clear the **Online** checkbox to create an offline tablespace.
4. In the **Contents** panel, specify the content type for the tablespace by selecting one of the following options:
 - Permanent—Select this option to use a tablespace for storing segments for permanent objects.
 - Temporary—Select this option to use a tablespace for allocating temporary segments for sort operations. Objects cannot be created or stored in a temporary tablespace.
 - Undo—Select this option to use a tablespace for storing undo records.
5. In the **Extent Management** panel, specify the extent management type for the tablespace by selecting one of the following options:
 - Dictionary—Select this option to create a data dictionary managed tablespace. (Dictionary-managed tablespaces cannot be created in a database where the SYSTEM tablespace is locally managed.)
 - Local—Select this option to create locally-managed tablespace.
6. Select a data block size for the tablespace from the **Block Size** field. This field lists all block sizes for which the buffer cache in SGA memory is configured. The size initially displayed is the default for the database. (The **Block Size** field is disabled for databases that are not configured to use multiple block sizes.)
7. If you are creating a permanent locally-managed tablespace, select one of the following **Segment Space Management** options:
 - Auto—Select this option if segment space should be managed automatically with bitmaps. The bitmaps track how much space is available for row inserts in segment data blocks. (**Auto** must be selected for a bigfile tablespace. If **Manual** is selected, the ORA-32772 error occurs when the script to create the tablespace is run.)
 - Manual—Select this option if segment space should be managed with free lists. These track which segment data blocks have space for row inserts. Oracle uses space in data blocks based on object values for the FREELISTS, FREELISTS GROUPS, and PCTUSED storage parameters.
8. If you are creating a locally-managed tablespace in an Oracle 10g database, specify datafile type for the tablespace. Select the **Bigfile** checkbox if type should be bigfile. The tablespace can contain one very large datafile. Clear the **Bigfile** checkbox if type should be smallfile. The tablespace can contain multiple smaller datafiles.

i Note: The **Bigfile** checkbox is automatically selected when bigfile is the default tablespace type for a database.
9. If you are creating a dictionary-managed tablespace, set default tablespace storage values in the **Attributes** panel (these values are used for objects without their own storage values):
 - Initial—Select the size of the first extent to allocate for tablespace objects.
 - Next—Specify the size of additional extents to allocate for tablespace objects. The second extent is the size specified for **Next**. Subsequent extents are the size of the last extent allocated multiplied by the value for PCTINCREASE.

- **Min Extents**—Select the minimum number of extents to allocate initially for tablespace objects.
 - **Max Extents**—Select the maximum number of extents that can be allocated for each tablespace object. The maximum includes the INITIAL extent. To allow an unlimited number of extents to be allocated, select **unlimited**.
 - **Pct Increase**—Enter the percentage by which each additional extent can grow over the last extent allocated. If you enter zero, all additional extents are the size specified for NEXT. If PCTINCREASE is greater than zero, the second extent allocated is the size specified for NEXT. Each extent after that is the size of the last extent multiplied by PCTINCREASE.
 - **Logging**—Specify whether logging to Oracle redo logs should be turned on or off for tablespace objects. Select **Logging** to turn on logging to Oracle redo logs. Clear the check box to turn off logging.
 - **Minimum Extent**—Specify a minimum size for extents created in the tablespace. Extent sizes will be equal to or a multiple of the value specified.
10. If you are creating a locally-managed tablespace, specify allocation type and the logging property in the **Attributes** panel (extents for all objects created or reorganized in the tablespace are sized based on allocation type):
- **Autoallocate**—Select this option if extents should be sized automatically by Oracle. The minimum extent size is 64K. Autoallocate is recommended for tablespaces that will contain tables requiring different extent sizes. (Autoallocate cannot be used for temporary local tablespaces.)
 - **Uniform**—Select this option if extents should be uniform in size. Also specify a size in the **Size** field. The default is 1M (1024K). Using a uniform size helps minimize fragmentation. The size should be large enough to prevent an excessive number of extents from being allocated.
 - **Logging**—Specify whether logging to Oracle redo logs should be turned on or off for tablespace objects. Select **Logging** to turn on logging to Oracle redo logs. Select **No Logging** to turn off logging.
11. Add datafiles to the tablespace using options in the **Datafiles** pane. One datafile is allocated by default. Click **Add Datafile** for each additional datafile you want to allocate. The number of datafiles you can add depends on the maximum allowed by your operating system.
- i** **Tip:** To delete a datafile, click its row and click **Delete Datafile**. If a tablespace has only one datafile, the datafile cannot be deleted. (Options for adding and deleting datafiles are disabled for bigfile tablespaces, which use only one datafile.)
12. Specify the following properties for each datafile in a tablespace:
- **Name**—Specify a datafile name in this field or use the default name. Default names include the name of the tablespace. The name of the first datafile ends in **_01.dbf**. The number increases by one for each datafile added. For example, the name of the second datafile ends in **_02.dbf**.
 - **Reuse**—Select this checkbox to reuse an existing datafile instead of creating one. Also enter the name of the datafile in the **Name** field.
 - **Specify Size**—Select this checkbox if you are reusing an existing datafile and want to change the size of the datafile. Clear this checkbox to retain the datafile's current size.
 - **Size**—Enter the amount of space to allocate for a datafile in this field. Also select **M** or **K** for megabytes or kilobytes. If you are changing the size of an existing datafile, the new size can be equal to or less than the physical size of the datafile. If you are sizing a bigfile datafile, the size can be up to 128 terabytes for a tablespace with a 32K block size. The size can be up to 32 terabytes for a tablespace with an 8K block size.
 - **Include Header Block**—Select this option to add extra space to a datafile for use by the file header. If a tablespace is locally managed, the space is also used by the extent bitmap. The amount of space added is equivalent to one data block for a dictionary managed tablespace and 64K for a locally managed tablespace.

- **Actual Size**—If you selected the **Include Header Block** option, view this field to see the total amount of space to be allocated for a datafile. The total includes space for data plus space for the header. (When **Include Header Block** is not selected, values for **Actual Size** and **Size** are the same.)
 - **Autoextend**—Select **ON** in this field if a datafile should be autoextendible. Select **OFF** if a datafile should not be autoextendible. When a datafile is autoextendible, it can allocate extents automatically as objects grow. This reduces the need for immediate intervention when objects require more space. It also reduces the risk of tablespace failure.
 - **Next auto extend size**—If a datafile is autoextendible, specify the size of the extents it can allocate. Also select **M** or **K** for megabytes or kilobytes.
 - **Max Size**—If a datafile is autoextendible, specify the maximum size to which it can grow. Also select **M** or **K** for megabytes or kilobytes.
13. Click **Generate Script** to generate the script for creating the tablespace. The script is displayed in the SQL Editor. Press **F9** to run the script.
 14. Close the SQL Editor and the Tablespace Properties window.
 15. Press **F5** from the Explorer to refresh the Tablespaces tab.

Alter Tablespaces

You can use the Tablespace Properties window to make changes to tablespaces, such as adding or deleting datafiles. After you modify properties, Tablespace Properties generates the script for altering a tablespace and opens the script in the SQL Editor.

The options available in the Tablespace Properties window are determined by the Oracle database version and the extent management type for a tablespace. Options that are unavailable for the selected tablespace are disabled in the window.

i Note: The following options are not supported: extent management (dictionary managed or locally managed), segment space management (auto or manual), block size, and bigfile or smallfile tablespace type.

See [Tablespace Properties Descriptions](#) for detailed descriptions of the tablespace property options.

To alter a tablespace with Tablespace Properties

1. Open Tablespace Properties by right-clicking a tablespace in the Explorer or a report, and selecting **Tablespace Properties**.
2. If your database is Oracle 10g or later, you can rename the tablespace by entering a new name in the **Tablespace** field. The name is displayed in bold. Using a standard name is recommended. A maximum of 30 characters is allowed.
3. To alter the online/offline property for a tablespace, change the setting for the **Online** checkbox. Select this checkbox to put a tablespace online. Its objects can be accessed by users and applications. Clear the checkbox to take a tablespace offline.
4. To alter the read-write property for a tablespace, change the setting for the **Read Only** checkbox. Select this checkbox to make the tablespace read-only. Write operations cannot be performed on datafiles the tablespace. Clear the checkbox to make the tablespace read-write.
5. To alter content type for a dictionary managed tablespace, select one of the following options in the **Contents** panel:
 - **Permanent**—Select this option to use a tablespace for creating and storing segments for permanent objects.
 - **Temporary**—Select this option to use a tablespace for allocating temporary segments for sort operations. Objects cannot be created or stored in a temporary tablespace.

6. To alter default storage values for a dictionary-managed tablespace, modify any of the following parameters in the **Attributes** panel:
 - **Initial**—Select the size of the first extent to allocate for tablespace objects.
 - **Next**—Specify the size of the additional extents to allocate for tablespace objects. The second extent is the size specified for **Next**. Subsequent extents are the size of the last extent allocated multiplied by the value for **PCTINCREASE**.
 - **Min Extents**—Select the minimum number of extents to allocate initially for tablespace objects.
 - **Max Extents**—Select the maximum number of extents that can be allocated for each tablespace object. The maximum includes the **INITIAL** extent. To allow an unlimited number of extents to be allocated, select **unlimited**.
 - **Pct Increase**—Enter the percentage by which each additional extent can grow over the last extent allocated. If you enter zero, all additional extents are the size specified for **NEXT**. If **PCTINCREASE** is greater than zero, the second extent allocated is the size specified for **NEXT**. Each extent after that is the size of the last extent multiplied by **PCTINCREASE**.
 - **Logging**—Specify whether logging to Oracle redo logs should be turned on or off for tablespace objects. Select **Logging** to turn on logging to Oracle redo logs. Select **No Logging** to turn off logging.
 - **Minimum Extent**—Specify a minimum size for extents created in the tablespace. Extent sizes will be equal to or a multiple of the value specified.
7. If the tablespace is locally managed, select **Logging** in the **Attributes** panel to turn on logging to Oracle redo logs for tablespace objects. Select **No Logging** to turn off logging.
8. To add a datafile to a smallfile tablespace, click **Add Datafile** in the **Datafiles** pane. Then specify the following properties in the new datafile row:
 - **Name**—Specify a datafile name in this field or use the default name. Default names include the name of the tablespace. The name of the first datafile ends in **_01.dbf**. The number increases by one for each datafile added. For example, the name of the second datafile ends in **_02.dbf**.
 - **Reuse**—Select this checkbox to reuse an existing datafile instead of creating one. Also enter the name of the datafile in the **Name** field.
 - **Specify Size**—Select this checkbox if you are reusing an existing datafile and want to change the size of the datafile. Clear this checkbox to retain the datafile's current size.
 - **Size**—Enter the amount of space to allocate for a datafile in this field. Also select **M** or **K** for megabytes or kilobytes. If you are changing the size of an existing datafile, the new size can be equal to or less than the physical size of the datafile. If you are sizing a bigfile datafile, the size can be up to 128 terabytes for tablespace with a 32K block size. The size can be up to 32 terabytes for a tablespace with an 8K block size.
 - **Include Header Block**—Select this option to add extra space to a datafile for use by the file header. If a tablespace is locally managed, the space is also used by the extent bitmap. The amount of space added is equivalent to one data block for a dictionary managed tablespace and 64K for a locally managed tablespace.
 - **Actual Size**—If you selected the **Include Header Block** option, view this field to see the total amount of space to be allocated for a datafile. The total includes space for data plus space for the header. (When **Include Header Block** is not selected, the values for **Actual Size** and **Size** are the same.)
 - **Autoextend**—Select **ON** in this field if a datafile should be autoextensible. Select **OFF** if a datafile should not be autoextensible. When a datafile is autoextensible, it can allocate extents automatically as objects grow. This reduces the need for immediate intervention when objects require more space. It also reduces the risk of tablespace failure.
 - **Next auto extend size**—If a datafile is autoextensible, specify the size of the extents it can allocate. Also select **M** or **K** for megabytes or kilobytes.

- **Max Size**—If a datafile is autoextensible, specify the maximum size to which it can grow. Also select **M** or **K** for megabytes or kilobytes.

i Tip: To alter an existing datafile for a smallfile or bigfile tablespace, click the row for that datafile and modify values for **Size**, **Include Header Block**, **Autoextend**, **Next auto extend size**, or **Max Size**. (Names of existing datafiles cannot be altered.) To delete the row for a datafile, click its row and click **Delete Datafile**.

9. Click **Generate Script** to generate the script for altering the tablespace. The script is displayed in the SQL Editor. Press **F9** to run the script.
10. Close the SQL Editor and the Tablespace Properties window.
11. Press **F5** from the Explorer to refresh the Tablespaces tab with any changes to datafiles in the tablespace.

i Tip: To increase or decrease the size of a datafile, see [Resize Datafile](#).

Tablespace Properties Descriptions

This section describes options in Tablespace Properties. Descriptions are given in alphabetic order by option name.

Block Size Options

Use the **Block Size** option to specify the data block size to use for a new tablespace. It is available for databases that are configured to allow use of different block sizes for different tablespaces.

When you create a tablespace, the block size initially displayed is the default for the database. You can select a different block size from a list. This includes all the block sizes for which the buffer cache in SGA memory is configured.

The Block Size option is disabled for existing databases and databases that do not allow multiple block sizes. You cannot specify a nonstandard block size for a temporary tablespace.

i Note When you select a block size for a dictionary managed tablespace, other values are adjusted automatically. If **Minimum Extent** is greater than zero, the value is adjusted to be equal to or a multiple of the new block size. If the **Include Header** option is selected for a datafile, the value for **Actual Size** is adjusted to reflect the change in header size caused by the change in block size.

Content Options

Content type options determine what type of segments a tablespace can contain.

Option	Description
Permanent	When content type is permanent, a tablespace can be used for creating and storing segments for permanent objects such as tables, indexes, and rollback segments.
Temporary	When content type is temporary, a tablespace can be used for allocating temporary segments for sort operations. Normally, many calls are required to allocate temporary segments for sort operations. In a temporary tablespace, one sort segment is cached and used for each instance requesting a sort. This improves performance for concurrent sorts and reduces their overhead. It also improves performance for medium-sized sorts that cannot be done completely in memory.
Undo	When content type is undo, a tablespace can be used for storing the undo records that are generated by Oracle. These records are used to undo (roll back) changes to the database

Option	Description
	<p>during recovery or to achieve read consistency. They are also used as requested by ROLLBACK statements.</p> <p>The undo content type cannot be used for dictionary-managed tablespaces. Tablespace Properties prevents use of undo in the following ways:</p> <ul style="list-style-type: none"> • When the Undo option is selected after the Dictionary option is selected, extent management is automatically changed to Local. • When the Dictionary option is selected after the Undo option is selected, a message alerts you that undo can only be used for locally managed tablespaces.

Datafile Type Option

Use the **Datafile type** option to specify tablespace type—bigfile or smallfile. The option is available for databases that are Oracle 10g or later. It is automatically selected when bigfile is the default tablespace type for a database. Bigfile and smallfile tablespaces can exist in the same database.

Tablespace	Description
Bigfile	<p>A bigfile tablespace contains one very large datafile or tempfile. The file can be up to 128 terabytes in size (for tablespaces using the 32K block size) or up to 32 terabytes in size (for tablespaces using the 8K block size). It provides much greater storage capacity than a smallfile tablespace and reduces the number of datafiles needed for a database.</p> <p>Introduced in Oracle 10g, bigfile tablespaces can be used with Oracle's Automatic Storage Management. They work best in systems that support large file sizes and have logical volume managers that support striping or RAID.</p> <p>Bigfile tablespaces support local extent management and automatic segment space management. They do not support dictionary extent management. They can be used with manual segment space management if tablespace content is temporary or undo.</p>
Smallfile	<p>A smallfile tablespace can contain multiple datafiles or tempfiles. All are smaller than the sizes allowed for bigfile tablespaces. The maximum number of datafiles a smallfile tablespace can contain depends on your operating system, but is typically 1022. The maximum file size allowed also depends on your operating system, but is typically 4MB blocks.</p> <p>Smallfile tablespaces are available for all versions of Oracle. The tablespace type traditionally used in Oracle databases, they work with both data dictionary and local extent management.</p>

Extent Management options

Extent management options determine how extents are managed in a tablespace—at the database level, by the Oracle data dictionary, or locally, by the tablespace itself.

Option	Description
Dictionary	<p>In a data dictionary managed tablespace, space use is tracked in the Oracle data dictionary. Extents are allocated based on object storage values. If objects do not have their own storage values, extents are allocated based on default tablespace storage values. Storage can be specified using fields that are displayed in the Attributes panel when the Dictionary option is selected.</p> <p>i Note Dictionary managed tablespaces cannot be created in a database where the SYSTEM tablespace is locally managed.</p>

Option	Description
Local	<p>In a locally managed tablespace, space use is tracked in bitmaps in tablespace datafiles. Extents are allocated based on allocation type—autoallocate or uniform. Allocation type can be specified using fields that are displayed in the Attributes panel when the Local option is selected:</p> <ul style="list-style-type: none"> • Autoallocate—When allocation type is autoallocate, extents for tablespace objects are sized automatically by Oracle (the minimum extent size is 64K). Additional extents can increase in size as an object grows. Autoallocate is a good choice for tablespaces that will contain tables requiring different extent sizes or tables for which you cannot predict growth rate. (Autoallocate cannot be used for temporary local tablespaces.) • Uniform—When allocation type is uniform, extents for tablespace objects are the same size. You specify the size to use in the Size field. The default size is 1M (1024K). Uniform is a good choice for tablespaces that will contain tables for which you can accurately predict growth rate. Ideally, the uniform size should be large enough to prevent an excessive number of extents from being allocated for the tables.

Name Options

Tablespace names can be specified when you create a tablespace. They can be changed whenever needed for tablespaces in Oracle 10g databases. To change a name, enter a new name in the **Tablespace** field of Tablespace Properties and run the alter script generated by Tablespace Properties. The field displays the new name in bold to indicate that a change has been made. A maximum of 30 characters is allowed for tablespace names.

Rename functionality can be used for both permanent and temporary tablespaces that are online. All of the datafiles in a tablespace must also be online.

i Note Renaming a tablespace does not change the names of its datafiles.

Note the following best practices for tablespace names:

- Use standard tablespace names.
- Do not attempt to rename the SYSTEM and SYSAUX tablespaces.
- Do not rename tablespaces that are included in reorganization scripts until script execution is complete. If you change a tablespace name before or during script execution, the script aborts with one of two errors. An “ORA-00959: tablespace does not exist” error occurs when the **Coalesce and check free space** option is not selected. An “ORA-20101: wait_for_freespace” error occurs when the Coalesce option is selected.

Segment Space Management Options

Segment space management options determine whether free space in tablespace segments is managed automatically or manually.

These options are set in the Segment Space Management panel of Tablespace Properties. They are disabled for Oracle 9i tablespaces that are defined as temporary. Manual segment space management is automatically used for temporary locally managed tablespaces. However, the SEGMENT SPACE MANAGEMENT clause is not included in Tablespace Properties scripts.

Option	Description
Auto	Automatic segment space management uses bitmaps to manage free space in tablespace segments. The bitmaps track how much space is available for row inserts in segment data

Option	Description
	blocks. Oracle uses space in data blocks based on information in these bitmaps. It does not check values for the FREELISTS, FREELIST GROUPS, and PCTUSED storage parameters. As a result, you do not need to specify these values when you reorganize objects in tablespaces that use automatic segment space management. If you specify the values in a reorganization wizard, they will be scripted and displayed in the interface, but they will not be used by Oracle.
Manual	Manual segment space management uses free lists to manage free space in tablespace segments. Free lists are lists of data blocks that have space available for row inserts. Oracle allocates free space based on object values for the FREELISTS, FREELIST GROUPS, and PCTUSED storage parameters.

Status Options

Status options determine whether a tablespace is online or offline. For existing tablespaces, status options also determine whether a tablespace is read-write or read-only. Status options are set in the Status panel of Tablespace Properties.

Online / Offline

Online/offline status is set with the **Online** checkbox.:

- **Online**—Select this checkbox to put a tablespace online. (It is selected by default.) When a tablespace is online, its objects are available to users and applications. Tablespaces are normally online.
- **Offline**—Clear this checkbox to take a tablespace offline. When a tablespace is offline, its objects are not available to users and applications.

Read-write / Read-only

Read-write status is set with the **Read Only** checkbox. This checkbox is displayed for existing tablespaces. It is not displayed when you create a tablespace. By default, new tablespaces are read-write.

Resize Datafile

Space Manager allows you to easily resize a datafile using the Datafile Resize wizard. You can launch the wizard from the Explorer after selecting a datafile to resize. Use this feature to resize a datafile prior to creating or running a reorganization script.

i Note: Previous releases of Space Manager (8.0.2 or earlier) allowed you to add a "script-pause" option to a reorganization script and use the pause to resize a datafile. Beginning with Space Manager 8.1, the script-pause option is no longer available. This is because its main functionality is superseded by the Datafile Resize wizard. Datafile Resize uses LiveReorg, which is preferable to standard reorganization, the method used with the script-pause function.

To resize a datafile

1. In the Databases tree-list in the Explorer, expand the tablespace node for the selected datafile.
2. Select the Datafile node to display the list of datafiles in right pane, and then select a datafile.
3. Right-click the datafile and select **Resize Datafile**. The Datafile Resize wizard opens.

4. To increase size, select **Increase datafile size** and click **Next**.
 - a. Specify a new size in the **Resize to** field and click **Next**.
 - b. On the Execute SQL page, click **SQL Execute** to execute the SQL command to resize the datafile.
5. To decrease size, select **Decrease datafile size** and click **Next**. The next page of the wizard displays the available options for resize method and resulting datafile size.
 - If you do not want to reorganize segments, select **No Data Movement** and then click **Next**. On the Execute SQL page, click **SQL Execute** to execute the SQL command to resize the datafile.
 - To use the recommended size and to reorganize segments, select **Recommended** and click **Next**. This option uses a live reorganization. The next page of the wizard displays the list of segments to be reorganized and several options. Review the following for additional information:

In Same Tablespace	Select to leave tables in their current tablespace.
Move to New Tablespace	Select to move tables to a different tablespace. Then select the new tablespace from the list. i Note: All tables in the list are moved to new tablespace.
Turn Off Autoextent for Datafile	Select this option if you do not want this datafile to be autoextensible.
Drop Datafile if Empty	Select to drop the datafile if it is empty after a successful reorganization.

After specifying reorganization options, click **Next**. Schedule the reorganization and resize script.

Execute Immediately	Select to schedule the script to run immediately
Start Job At	Select to schedule the job to run at a later date and time.

Click **Submit Job**. The scheduled script displays in the Script/Job Monitor.

Reorganize Objects with Reorg Manager

About Reorganization Wizards

Reorganization is the main solution for space-use problems such as wasted space, fragmentation, chained rows, and table/index contention.

Space Manager provides two reorganization wizards that allow you to reorganize objects and tablespaces in a variety of ways.

- [Reorg Manager](#)
- [Partitioning Wizard](#)

Space Manager also provides two dialogs for performing simple reorganization tasks.

- [Move to Tablespace Dialog](#)
- [Reclaim Space Dialog](#)

Reorg Manager

Reorg Manager can be used to reorganize and relocate objects from anywhere in a database.

The Reorg Manager provides two methods for reorganization—**standard** and **live**. The Standard method is offline; the live method is online.

- **Standard Reorg**—Standard reorganization is an offline method. During a standard reorganization, tables are not available for DML activity. See [Advantages of Standard Reorganization](#) on page 55 for more information.
- **Live Reorg**—Live reorganization is an online method. During a live reorganization, tables are available for DML activity by applications and users. When the online switch style is used, tables remain available throughout a live reorganization, including the switch from original table to reorganized copy table. Live reorganizations can be performed when the LiveReorg option is licensed. See [About Live Reorg with Reorg Manager](#) on page 56 for more information.

For help selecting a reorganization method to use, see [Standard Reorganization](#).

To get started using the Reorg Manager, see [Launch Reorg Manager](#).

i Tip: You can save Reorg Manager settings as a Reorg Plan to edit and reuse later. See [Save Reorg Manager Settings](#) on page 81 for more information. You can also use saved settings to automate reorganization script execution. See [About Scheduling a Recurring Reorganization](#) on page 122 for more information.

Partitioning Wizard

The Partitioning Wizard can be used to partition, unpartition, and repartition tables and indexes during live reorganizations. This wizard provides a graphic interface that lets you easily visualize a partitioning scheme and specify partitioning options. The wizard also automates tasks such as setting partitioning key values, sizing partitions, and selecting tablespaces for I/O balancing. See [Requirements for the Partitioning Wizard](#) on page 95 for more information.

To get started using the Partitioning Wizard, see [Launch Partitioning Wizard](#).

Move to Tablespace Dialog

Use the Move to Tablespace dialog to move an object to another tablespace without opening the Reorg Manager Wizard. See [Move Objects to Tablespace](#) on page 53 for more information.

Reclaim Space Dialog

Use the Reclaim Space dialog to reclaim space without launching the Reorg Manager Wizard. This method performs a live reorganization using the default settings. See [Reclaim Space](#) on page 53 for more information.

Keys Options in Reorganization Wizards

Review the following key options available in reorganization wizards:

- Target tablespace options—Use target tablespace options to reorganize objects in their current tablespace or relocate them for better I/O balancing. If you choose to leave objects in their current tablespace during a standard reorganization, you can reorganize them using free space in another tablespace on an interim basis.
- Sort options—Use sort options to sort data and improve its proximity for faster query access. These options are available for standard and live reorganizations.
- NOLOGGING option—Use Oracle's NOLOGGING option to speed up reorganization of indexes.
- Parallel processing by Oracle—Use Oracle's parallel processing to speed up reorganization of tables and indexes.
- Parallel processing by QSA— Use parallel processing by the Quest Server Agent (QSA) to speed up reorganization of indexes and recompile of dependencies.
- Scheduling options—Use scheduling options to schedule a reorganization script to run at times when database activity is light and the script will have the least impact.

Identify Objects That Need to Be Reorganized

You can quickly identify objects and tablespaces that need to be reorganized by viewing them in the Explorer or by generating a Reorganization report.

Method	Description
<p>Use the Explorer to identify reorg need</p>	<p>Use the Explorer to identify objects with the greatest need for reorganization.</p> <p>Explorer - Segments Grid</p> <p>Each segment/object grid in the Explorer includes a Reorg Need column. This column provides the calculated reorg need value for each object in the grid.</p> <p>To view Reorg Need column</p> <ol style="list-style-type: none"> In the Explorer, select a tablespace or owner to view reorg need. <ul style="list-style-type: none"> Tablespace: Select a tablespace in the Explorer tree-list (left pane). The grid on the right lists the tablespace segments and includes a Reorg Need column. Owner: Select an owner in the Explorer tree-list (left pane). Expand the owner node and then select Tables or Indexes. The grid on the right lists the owner's objects and includes a Reorg Need column. Review the Reorg Need column to determine those segments that would benefit from reorganization. Click the column header twice to sort from highest to lowest need. <p>See Monitor Tablespace and Object Storage on page 26 for more information.</p> <p>Explorer - Advanced Search</p> <p>Use the Advanced Search window to search for objects based on their reorg need value. See Advanced Search on page 28 for more information.</p> <p>Explorer - Datafile Map</p> <p>Use the Datafile Map to view segments with extents that are blocking you from resizing the datafile.</p> <p>To view segments with blocking extents</p> <ol style="list-style-type: none"> Select the tablespace for which you want to view a datafile map. Expand the tablespace datafile node and select a datafile. The datafile map displays along with a list of segments with blocking extents. Use the list of segments with blocking extents to determine which segments you want to reorganize. Right-click a segment in the list to launch the Reorg Manager. <p>See Monitor Datafile Storage on page 29 for more information.</p>
<p>Use Reorganization Reports to identify reorg need</p>	<p>Reorganization Reports</p> <p>There are two types of Reorganization reports you can generate.</p> <ul style="list-style-type: none"> The <i>Predicted Reorganization Benefit</i> report shows you the amount of space you can reclaim if you reorganize a selected table or index. The <i>Reorganization History</i> report shows which tables or indexes were reorganized during the selected time period, which allows you to extrapolate space savings due to reorganization activity. For planning reorganizations in

Method	Description
	<p>the future, this report provides you with historical information on how much time it took to run reorganizations on specific objects.</p> <p>See Generate Reorganization Reports on page 147 for more information.</p> <p>Reorganization Candidates Report</p> <ul style="list-style-type: none"> Identify objects with the highest reorg need by generating a <i>Reorganization Candidates</i> report. See Generate Reorganization Candidates Report on page 150 for more information. <p>See Run Reports on page 146 for more information about all types of reports you can generate in Space Manager.</p>

Move Objects to Tablespace

You can easily move an object or segment to another tablespace from within the Explorer without launching the Reorg Manager. You can execute the task immediately or schedule it.

To move an object to a tablespace

- In the Explorer, do one of the following:
 - Right-click the object or segment and select **Move to Tablespace**.
 - Drag the object from the Segments grid (or Tables or Indexes grid) to a new tablespace in the Explorer tree-list.
- In the Move Objects to Tablespace dialog, select a **Target Tablespace** from the list.
- Then select one of the following:
 - Execute Immediately** to move the object immediately.
 - Schedule Date** to move the object at a scheduled date and time.
- Click **Next**.
- Review the summary and click **Finish** to execute immediately or when scheduled.

i Tip: You can also perform this action from the search results in the Advanced Search window. Right-click an object in the search results and select **Move to Tablespace**.

Reclaim Space

You can easily reclaim space in a segment/object from within the Explorer without opening the Reorg Manager Wizard. The Reclaim Space method performs a live reorganization using the default settings.

You can execute the task immediately or schedule the task in the Reclaim Space dialog.

To reclaim space

1. In the Explorer, right-click an object or segment and select **Reclaim Space**.
2. In the Reclaim Space dialog, select one of the following:
 - **Execute Immediately** to reclaim space in the object immediately.
 - **Schedule Date** to reclaim space in the object at a scheduled date and time.
3. Click **Next**.
4. Review the summary and click **Finish** to execute immediately or when scheduled.

i Tip: You can also perform this action from the search results in the Advanced Search window. Right-click an object in the search results and select **Reclaim Space**.

Resize Datafile

Space Manager allows you to easily resize a datafile using the Datafile Resize wizard. You can launch the wizard from the Explorer after selecting a datafile to resize. Use this feature to resize a datafile prior to creating or running a reorganization script.

i Note: Previous releases of Space Manager (8.0.2 or earlier) allowed you to add a "script-pause" option to a reorganization script and use the pause to resize a datafile. Beginning with Space Manager 8.1, the script-pause option is no longer available. This is because its main functionality is superseded by the Datafile Resize wizard. Datafile Resize uses LiveReorg, which is preferable to standard reorganization, the method used with the script-pause function.

To resize a datafile

1. In the Databases tree-list in the Explorer, expand the tablespace node for the selected datafile.
2. Select the Datafile node to display the list of datafiles in right pane, and then select a datafile.
3. Right-click the datafile and select **Resize Datafile**. The Datafile Resize wizard opens.
4. To increase size, select **Increase datafile size** and click **Next**.
 - a. Specify a new size in the **Resize to** field and click **Next**.
 - b. On the Execute SQL page, click **SQL Execute** to execute the SQL command to resize the datafile.
5. To decrease size, select **Decrease datafile size** and click **Next**. The next page of the wizard displays the available options for resize method and resulting datafile size.
 - If you do not want to reorganize segments, select **No Data Movement** and then click **Next**. On the Execute SQL page, click **SQL Execute** to execute the SQL command to resize the datafile.
 - To use the recommended size and to reorganize segments, select **Recommended** and click **Next**. This option uses a live reorganization. The next page of the wizard displays the list of segments to be reorganized and several options. Review the following for additional information:

In Same Tablespace	Select to leave tables in their current tablespace.
Move to New Tablespace	Select to move tables to a different tablespace. Then select the new tablespace from the list. i Note: All tables in the list are moved to new tablespace.

Turn Off Autoextent for Datafile	Select this option if you do not want this datafile to be autoextensible.
Drop Datafile if Empty	Select to drop the datafile if it is empty after a successful reorganization.

After specifying reorganization options, click **Next**. Schedule the reorganization and resize script.

Execute Immediately	Select to schedule the script to run immediately
Start Job At	Select to schedule the job to run at a later date and time.

Click **Submit Job**. The scheduled script displays in the Script/Job Monitor.

About Standard Reorg with Reorg Manager

Standard reorganization is an offline reorganization method. This means that the objects being reorganized are not available and should not be accessed during a standard reorganization. DML or DDL activity should not be performed against these objects by users or applications. If either type of activity is performed, data loss can occur.

i Note Standard reorganizations for objects from anywhere in a database can be performed with Reorg Manager.

Advantages of Standard Reorganization

One of the main advantages of standard reorganization with Reorg Manager is that it supports more object types, datatypes, and options than any other method.

What is Not Supported for Standard Reorganization

This topic lists which Oracle features are not supported for standard reorganizations. Use the information in this topic when selecting which reorganization method to use and which objects to include in a reorganization.

Category	Oracle Feature Not Supported
Tables and Indexes	Temporary tables
	Tables with dimensions
	Advanced Queuing tables
	Replication tables; objects that are part of a replication group
	External tables
IOTs and Clusters	Domain indexes (including interMedia and ConText indexes)
	Bitmap indexes on IOTs
	Clusters that contain tables with LONG columns
User Defined Data Types	Object tables (data types for tables)

Category	Oracle Feature Not Supported
	Nested tables
Various Data Types	XMLType tables
Miscellaneous	Oracle Label Security

About Live Reorg with Reorg Manager

Live reorganization with the Reorg Manager is designed to resolve space-use problems with minimum impact on data availability. When tables are reorganized using this method, applications can remain online and users can continue activity against the tables. Data remains available for Data Manipulation Language (DML) statements, including INSERTS, UPDATES, and DELETES.

When the online switch is used for a live reorganization, tables remain available throughout the entire reorganization, including the switch from original table to reorganized copy table. When the T-Lock switch is used, original and copy tables are locked very briefly for the switch.

i Note: Before you can perform live reorganizations with Space Manager, the LiveReorg option must be licensed. In addition, the QSA Server Agent must be installed, current, and running.

Live reorganization supports the following. For a detailed list of what is not supported by live reorganizations, see [What is Not Supported for Live Reorganizations](#).

- Non-partitioned and partitioned tables
- Individual partitions and subpartitions
- Tables with LOBs
- Tables with LONGs that are up to two gigabytes in size
- Tables with both a LOB and a LONG

What is Not Supported for Live Reorganizations

This topic lists which Oracle features are not supported for live reorganizations. It also lists which features are not supported for the T-Lock switch or the Online switch.

Use the information in this topic when selecting which reorganization method to use, which objects to include in a reorganization, and which type of table switch to use in a live reorganization.

Not Supported for Live Reorganization

The following Oracle features are not supported for live reorganization.

Category	Oracle Feature Not Supported
Tables and Indexes	Entire system-partitioned tables i Note: Individual partitions of system-partitioned tables are supported for live reorganization.
	Entire reference-partitioned tables

Category	Oracle Feature Not Supported
	i Note: Individual partitions of reference-partitioned tables are supported for live reorganization using the T-Lock switch.
	Parent tables of reference-partitioned table
	Indexes without their tables To reorganize an index online independently of its table, perform a standard reorganization with Rebuild ONLINE
	Temporary tables
	Tables with dimensions
	Advanced Queuing tables
	Replication tables; objects that are part of a replication group
	External tables
	Tables owned by SYS or SYSTEM or with names starting with QUEST_%
	Individual partitions of a partitioned table are not supported if Space Manager needs to create a ROWID posting index and the UseRowidForPartitions parameter is not set to true. In this special case, the entire partitioned table is reorganized (if supported). See ROWID Posting Indexes on page 218 for more information.
IOTs	Entire partitioned IOTs i Note: Individual partitions of IOTs are supported for live reorganization.
Clusters	Clusters, clustered tables, cluster indexes, indexes on clustered tables
Materialized Views	Materialized views
User Defined Data Types	Object tables (data types for tables)
	Nested tables
Various Data Types	XMLType tables
Miscellaneous	Oracle Label Security
	Tables with security policies enabled

Not Supported for T-Lock Switch

The following features are supported for live reorganization, but not the T-Lock switch. When a live reorganization is performed on an object containing one of these features, the Online switch is automatically used.

Category	Oracle Feature Not Supported
IOTs	Non-partitioned IOTs
Materialized Views	Materialized view logs Do not refresh views or access associated tables during reorganization; T-Lock switch can be used for partitions and subpartitions when they are reorganized individually (EXCHANGE partition reorganization method is used)

Not Supported for Online Switch

The following features are supported for live reorganization, but not when using the Online switch. When performing a live reorganization on objects containing these features, the T-Lock switch is required. To create a live reorganization script for objects containing one of these features, do not select the **Exclude tables requiring the T-Lock switch (online switch only)** option in the Reorg Wizard. See [Select Global Reorganization Options](#).

i Note: Oracle's Partitioning Option must be installed and licensed to use the Online switch.

Note that for the supported partitioned tables and indexes, when the online switch is used, partitions are reorganized individually.

Category	Oracle Feature Not Supported
Tables and Indexes	Non-partitioned tables with partitioned indexes
	Partitioned tables with global indexes
	Individual partitions of reference-partitioned tables
LONG Data Types	LONGs and LONG RAWs

Concepts and Terms

This section describes concepts and terms that apply to live reorganizations.

See also, [How Live Reorganizations Work](#).

Concept	Description
LiveReorg Objects and Tablespace	During a live reorganization, QSA uses LiveReorg objects and the LiveReorg tablespace. LiveReorg objects are created at the beginning of a live reorganization and stored in the LiveReorg tablespace. This is specified with the agent's LW_TABLESPACE parameter. LiveReorg objects include a collection table that is used for storing information on live transactions. One set of LiveReorg objects is created for each table in a live reorganization script. The objects are dropped after the switch from original table to reorganized copy table.
Original Table	This is the table being reorganized. The term "original table" is used for non-partitioned tables, partitioned tables, table partitions, and table subpartitions.
Copy Table	This is a copy of the original table. It is created at the beginning of a live reorganization and all original data is copied into it. Data is reorganized as it is inserted into the copy table. The copy table uses the storage attributes specified in Reorg Manager. If attributes are not specified, the original table's attributes are used.

Concept	Description
Live Transactions	As the copy table and its indexes are created, live DML activity can continue against the original table. Information on live transactions is stored in a collection table. The transactions are then posted (copied) to the reorganized copy table by QSA.
Posting	Posting is the process where live transactions against the original table are copied to its reorganized copy table. Posting is performed by QSA using information on live transactions that is stored in the agent's collection table.
Posting Index	<p>A posting index is used when live DML transactions are posted to the reorganized copy table. It can serve one or two purposes, depending on whether it is a selective or non-selective index:</p> <ul style="list-style-type: none"> • Selective posting indexes are used to uniquely identify rows in the reorganized copy table. They ensure that live transactions are posted to the correct rows. A selective index is one where every index entry points to a single location in table data. The PRIMARY KEY and unique indexes are selective indexes. A nonselective index is one where each index entry points to multiple locations in table data. • All posting indexes are used to speed up the posting process by increasing the efficiency of searches on the copy table. <p>What type of index is used as the posting index depends on what is available for a table. Space Manager tries to use the most selective index. See How Space Manager Chooses a Posting Index on page 215 for more information.</p>
Switch	<p>The switch is the point in a live reorganization when the original table is replaced with its reorganized copy table. At this point, all live transactions have been posted to the copy table so that the original table and copy table are identical in terms of the data they contain. There are two switch styles (or modes) available:</p> <ul style="list-style-type: none"> • Online switch—When the online switch is used, a table remains available for DML activity throughout a live reorganization, including the switch. • T-Lock switch—When the T-Lock switch is used, the table is unavailable during the switch. Any application trying to access the table during the switch will get a database error. Be sure that your application will handle these errors appropriately. See Switch Mode on page 62 for more information.

Best Practices for Live Reorganizations

Read this topic before creating and running live reorganization scripts.

This section provides best practices to follow before and during a live reorganization. It also provides best practices to follow if a live reorganization script is interrupted.

i Note: Space Manager provides the best Reorg Need and Wasted space information when statistics are current in the Repository. Be sure to run the Repository Update job on regular or nightly basis (see [Change Repository Update Schedule](#)).

! CAUTION: Reorganization commands must not be modified or removed from a script. If they are changed in any way, a reorganization can fail or unpredictable results can occur. Most commands begin with QUEST_EXEC.

Before Running a Live Reorganization

Review the following best practices prior to running a live reorganization

- **Reserve the LiveReorg tablespace**—Only use the LiveReorg tablespace for LiveReorg objects. If you locate a non-LiveReorg object in that tablespace, do not select the object for a live reorganization. Also, do not select a LiveReorg object for a live reorganization. The LiveReorg tablespace is specified with QSA's LW_TABLESPACE parameter.
- **Check free space in the LiveReorg tablespace**—Make sure sufficient free space is available in the tablespace being used for LiveReorg objects. If sufficient space is not available, a live reorganization will fail.
- **Do not select a table and its log together**—If a table has a materialized view log, do not select both the table and the log for the same reorganization. Also, if the table is included in one live reorganization script and the log is included in another live reorganization script, do not run the scripts at the same time.
- **Check for constraints that prevent NULLs in LONG columns**—Make sure tables with LONG columns do not have constraints that prevent insertion of NULLs into LONGs. If NULLs cannot be inserted into LONG columns, a live reorganization will fail with an error such as “ORA-01400: Cannot insert NULL” or “ORA-02290: Check constraint violated”. Examples of constraints that prevent insertion of NULLs include NOT NULL constraints and CHECK constraints with a NOT NULL condition.
- **Select the T-Lock switch for ROWID posting indexes**—If a ROWID index must serve as the posting index, select a combination of online and T-Lock switch (the default option). If you select online-only, tables that must use a ROWID index are excluded from reorganization.
- **Note time factors for the T-Lock switch and multiple partitions**—When multiple partitions or subpartitions from the same table are reorganized with the T-Lock switch, the switch is made individually for each partition or subpartition. If there are global indexes on the parent-partitioned table, these indexes are reorganized along with each partition or subpartition. Due to both factors, it may be faster to reorganize the entire table. Only one switch is made and global indexes are reorganized only once. (When the online switch is used, partitions are always reorganized individually even when the entire parent table is selected for reorganization.)
- **Note that SQL is used for tables with UDTs**—If a table contains a user-defined datatype (UDT), SQL is used as the data movement method for that table. Tables with UDTs are not supported for FastCopy. Tables with a UDT and a LONG column larger than 32,760 bytes cannot be reorganized on a live basis. This is because neither data movement method can be used for the table. FastCopy cannot be used because of the presence of a UDT. SQL cannot be used because of the size of the LONG column.
- **Schedule a live reorganization**—Schedule a live reorganization script so that it is run by QSA. LiveReorg commands can only be processed by the agent. Do not run a live reorganization script from the SQL Editor. If you try to do this and ignore error messages, data loss can occur.

Best Practices During a Live Reorganization

- **Do not run DDL statements**—Do not run Data Definition (DDL) statements against the objects being reorganized. (DML statements can be run.) DDL activity against a table should be stopped before you open Reorg Manager for that table. It should not be resumed until reorganization of the table is complete. Examples of DDL statements include ALTER TABLE, CREATE INDEX, and DROP TRIGGER.
- **Do not run multiple scripts for the same tables**—Do not run a reorganization script at the same time as another Space Manager script when both include the same objects or tables that share referential integrity.
- **Avoid operations that cannot be posted**—Avoid operations that cannot be posted to copy tables during a live reorganization. These operations are:
 - SQL*Loader direct path load.
 - SQL*Loader with the REPLACE option.
 - TRUNCATE.
- **Avoid certain transactions and operations**—Avoid long transactions during a live reorganization. Ideally, you should commit after each transaction. Also avoid operations that generate a lot of activity.

Examples of these operations include bulk loads and updates to every row in the table being reorganized.

- **Avoid large batch jobs**—Avoid running large batch jobs during an online switch. One way to do this is to schedule a reorganization script for a time when no large batch jobs are running.
- **Avoid canceling a reorganization**—Avoid canceling a reorganization script at any point, especially the switch. Although it is possible to cancel and restart reorganization scripts, you should only stop and start the script due to script failures and system shutdowns. Do not use the **Cancel and Clean up** command if you intend to restart a script.

Best Practices During Interrupted Execution

- **Restart most interrupted live reorganization scripts**—If a live reorganization script fails, resolve the problem that caused the failure. Then restart the script using the Restart function in the Scripts/Job Monitor. See [Restart for Live and Partitioning Scripts](#) on page 125 for more information. It is recommended that you restart most interrupted live reorganization scripts, especially scripts that were interrupted during a T-Lock switch. If you decide not to restart a script, contact Quest Software Technical Support for assistance in cleanup and recovery.
- **Do not make changes to objects being reorganized**—While a live reorganization script is waiting to be restarted, do not run DDL statement against the objects being reorganized. Do not make changes to the structure of an original table or its copy table. Also, do not make changes to indexes, constraints, or triggers, including the collection trigger. DML activity can continue against an original table if the script was interrupted prior to the switch or during an online switch. It should be avoided if the script was interrupted during a T-Lock switch.
- **Do not make changes to LiveReorg objects**—While a live reorganization script is waiting to be restarted, do not drop or make changes to the temporary LiveReorg objects used by QSA. Leave these objects in place. They will be used when the script is restarted.
- **Troubleshoot collection-trigger failures**—If an error occurs during execution of the triggers used for collecting live transactions, a live reorganization fails during posting with this error: “Aborted QSA-20391 ORA-20777 during internal SQL”. To identify the cause of the error, check the script execution log. To recover from the error, resolve the problem that caused it. Then restart the script using the Restart function in the Scripts/Job Monitor.
- **Troubleshoot materialized view log errors**—If a row cannot be inserted in the materialized view log used for collecting direct UPDATES to LOBs, the following error is returned to the application that is accessing the table being reorganized: “ORA-12096: Error in materialized view log on ‘string’.‘string’.” In most cases, the error occurs when the LiveReorg tablespace does not have enough space for the materialized view log to grow. In this case, you can recover by increasing the amount of free space in the LiveReorg tablespace. Transactions can then continue against the table being reorganized. QSA automatically continues the live reorganization from the point at which it was interrupted.

Best Practices for RAC/OPS Environments

This section provides best practices to follow for live reorganizations in active-active Real Application Clusters (RAC) and Oracle Parallel Server (OPS) environments:

- **Instance and node**—Space Manager server components and QSA should be installed on the primary instance for a cluster database. The agent should be installed for only that instance. See “Install the Agent on OPS or RAC” in the Space Manager with LiveReorg Installation Guide.
- **Instance connection**—When you generate a live reorganization script, the Space Manager client must be connected to the instance where QSA is installed. You can make it possible to consistently connect to the agent instance by performing connection setup. See “Connection Setup for OPS or RAC” in the Space Manager with LiveReorg Installation Guide.
- **Changes to configuration**—Your RAC or OPS configuration should not be changed in terms of active and inactive instances while a live reorganization is in progress. Active instances should remain active. Inactive instances should remain inactive. If your configuration is changed during a live reorganization,

the reorganization script could fail. If this happens, restart the failed script after RAC or OPS configuration is complete.

- **Failover**—QSA does not fail over from one instance to another in an active-active environment. If the agent instance fails during a live reorganization, script execution aborts. To address this situation, bring up the instance where the agent is installed and restart script execution. If you cannot bring up the agent instance, contact Quest Software Technical Support for assistance in installing the agent on another instance. Then restart script execution using the Restart function in the Scripts/Job Monitor.

Switch Mode

Switch mode (style) determines whether a table remains available for DML activity during the switch from original table to reorganized copy table. A table remains available when the online switch mode is used. A table is briefly unavailable when the T-Lock switch is used.

In addition to switch mode, you can select a switch option for a live reorganization:

- **Switch mode** determines how the switch from original table to the reorganized copy table is made: online switch or T-Lock switch. The switch mode is selected in the first page of the Reorg Manager.
- **Switch option** determines how the switch begins once a copy table is ready to replace its original table. The switch can begin automatically, upon user approval, or in a time window you specify. A switch option is selected in the Scripting Options page of the Reorg Manager.

More About the Online Switch

When the online switch mode is used, the table being reorganized remains available throughout a live reorganization, including the switch. Applications can remain online and users can continue DML transactions against the original table as it is replaced with its reorganized copy.

The original table remains available even if execution fails. All live transactions made while a script is interrupted are posted to the copy table when script execution resumes. After you resolve the problem that caused a failure, you can restart script execution using the Restart function in the Scripts/Job Monitor. However, you should not restart a script if an error prevents an online switch. See [Restart for Live and Partitioning Scripts](#) on page 125 for more information.

i Note: Oracle's Partitioning Option must be installed and licensed to use the Online switch.

Special Considerations

- When the online switch is used for a live reorganization of a partitioned table, the table's partitions or subpartitions are reorganized individually. A message alerts you to this after you select the **Exclude tables requiring the T-Lock switch (online switch only)** mode.
- The online switch is not used with FastCopy or DBMS_DataPump. When the **Exclude tables requiring the T-Lock switch (online switch only)** mode is selected, all tables will be reorganized with SQL.
- The **Exclude tables requiring the T-Lock switch (online switch only)** mode should not be selected when a ROWID posting index will be used for any of the tables in a script. See [Posting Indexes](#) on page 214 for more information.
- The **Exclude tables requiring the T-Lock switch (online switch only)** mode cannot be used for certain types of objects. See [What is Not Supported for Live Reorganizations](#) on page 56 for more information.

More About the T-Lock Switch

When the T-Lock switch is used, the original table and its reorganized copy table are locked against changes during the switch. Trigger locks are used. Normally, the duration of the switch stage is very brief.

Although T-Locks prevent write access, they allow read-only access for most of the switch. When an entire table is being reorganized, read-only access is not available at the point when original and copy tables are renamed.

At that point, Oracle places an exclusive lock on the original and copy tables. The duration of the lock is a few seconds. To see when objects are being renamed, check the Scripts/Job Monitor. This displays an ALTER TABLE RENAME statement as the rename takes place.

What Is Locked

When a table partition or subpartition is reorganized individually, a T-Lock is placed on its parent-partitioned table. The entire table is locked. A T-Lock is also placed on the copy object. When an entire table is reorganized, T-Locks are placed on both the original table and its copy table. T-Locks on original and copy tables remain in place throughout the switch stage.

In some cases, T-Locks are also placed on tables that have FOREIGN KEY relationships with the table being reorganized. These include tables that reference the original table and tables that are referenced by the original table. Any T-Locks on tables involved in FOREIGN KEY relationships last only as long as it takes to recreate constraints.

Space Manager locks all tables related to the reorg table before starting the switch. If Space Manager cannot lock all the tables, it removes the locks and aborts the script while still capturing updates. This allows you to restart the reorg script when there is less or no activity going against the tables.

At the end of the switch stage, T-Locks are removed and the reorganized table can be accessed. T-Locks are removed as follows:

- Removal from tables—If a table was reorganized, the T-Lock is removed from the reorganized copy table. If the original table is dropped after the switch, the T-Lock is dropped along with it. If the original table is kept, the T-Lock remains in place. If you need to remove the T-Lock from the original table, contact Quest Software Technical Support (see [About Us](#)).
- Removal from partitions and subpartitions—If a partition or subpartition was reorganized, the T-Lock is removed from its parent-partitioned table. At this point, the reorganized partition or subpartition is part of the original parent table.

What Happens When Locked Tables Are Accessed

If users or applications attempt to access a T-Locked table during the switch, one of two messages is displayed in the applications. Messages are displayed in the following circumstances:

- If attempts are made to modify data in a table protected by a T-Lock, this QSA Server Agent message displays:

QSA-20509: Live reorganization in progress

The purpose of the message is to alert users that changes cannot be made at the current time. It does not indicate a problem with a reorganization.

- If attempts are made to query a table protected by an exclusive lock, this Oracle error might display:

ORA-00942: Table or view does not exist

If SAP is the application in use, a “table unknown error” is displayed. An exclusive lock is applied by Oracle when a table is renamed during a T-Lock switch.

A message is not raised if a T-Lock switch is made when there is no activity against the tables being reorganized. You can control when a switch is made using scripting options that determine how the switch begins. The user approval option allows you to stop activity and then approve the switch. The time window option allows the switch to be made automatically at a time when the database is quiet. See [More About Scripting Options \(LiveReorg\)](#) on page 86 for more information.

Change Approval Method for Online Scripts

You can change the approval method (switch option) for online scripts. Approval method determines how the switch proceeds for a live reorganization script or a partitioning script.

Approval method can be changed for online scripts that are waiting to run or waiting to be restarted. A change in approval method applies to all tables in a script for which the switch has not yet been made.

For more information about approval methods (also known as *Switch Options*), see [Select Scripting Options](#) and [More About Scripting Options \(LiveReorg\)](#).

To change approval method

1. In the Script/Job Monitor, do one of the following to change approval method:
 - Right-click a script group and select **Change Approval Type**.
 - Right-click a script group and select **Properties**. Then select the Script Attributes tab.
2. Select one of the approval types.
3. Click **OK** to save your changes and close the dialog.

Approve the Switch

When the switch option for a live reorganization script is **Upon user approval**, you must approve the switch for each table in the script. You can do this from the Scripts/Job Monitor or from its Live Reorg Detail pane using one of the following procedures.

To see if a table is waiting for the switch to be approved, check the **Status Message** column in Scripts/Job monitor. “Waiting for approval” is the status for a script when one of its tables is waiting for approval.

After you approve the switch for one table in a script, check to see when the next table is ready for the switch. How much time passes before approval is required depends on the size of the table being reorganized, the number and size of its indexes, and the level of activity against the table.

To approve the switch from the Scripts/Job Monitor

1. Open the Scripts/Job monitor by selecting **Manage | Script/Job Monitor**.
2. Check the **Status Message** column. “Waiting for approval” is displayed for each live reorganization script with a table that is ready for the switch.
3. Right-click a live reorganization script and select **Approve Switch**.

To approve the switch from Live Reorganization Detail

1. Open the Scripts/Job monitor by selecting **Manage | Script/Job Monitor**.
2. Select a live reorganization script. Information on the script is displayed in the Live Reorganization Detail pane.
3. When “Stage 3 (Waiting for approval)” is displayed for a table in the Live Reorg Detail pane, right-click anywhere in the pane and select **Approve Switch**.

Getting Started with Reorg Manager

The Reorg Manager provides two methods of reorganizing objects: Live Reorganization and Standard Reorganization. This topic provides information to help you select a reorganization method when using the Reorg Manager.

The method you select is used for all tables in a reorganization script.

i Note The Partitioning Wizard automatically uses the live method.

See the following topics for additional information about selecting a reorganization method:

- [Reorganization Method Selection Chart](#)
- [What is Not Supported for Standard Reorganization](#)
- [What is Not Supported for Live Reorganizations](#)
- [Concepts and Terms](#)

Standard Reorganization

Standard reorganization is an offline reorganization method. This means that the objects being reorganized are not available and should not be accessed during a standard reorganization. DML or DDL activity should not be performed against these objects by users or applications. If either type of activity is performed, data loss can occur.

Advantages of Standard Reorganization

One of the main advantages of standard reorganization with Reorg Manager is that it supports more object types, datatypes, and options than any other method.

Live Reorganization

The Live Reorganization option is enabled when LiveReorg is licensed and QSA is installed, current, and running for a database. Options for the online switch are enabled when Oracle's Partitioning Option is installed for a database.

i Note: Oracle's Partitioning Option must be installed and licensed to use the Online switch.

Live reorganization with the Reorg Manager is designed to resolve space-use problems with minimum impact on data availability. When tables are reorganized using this method, applications can remain online and users can continue activity against the tables. Data remains available for Data Manipulation Language (DML) statements, including INSERTS, UPDATES, and DELETES.

When the online switch is used for a live reorganization, tables remain available throughout the entire reorganization, including the switch from original table to reorganized copy table. When the T-Lock switch is used, original and copy tables are locked very briefly for the switch.

i Note: Before you can perform live reorganizations with Space Manager, the LiveReorg option must be licensed. In addition, the QSA Server Agent must be installed, current, and running.

Smart Start for Standard Reorganizations

This Smart Start outlines the workflow for standard reorganizations with Reorg Manager.

i Important: Before you run a standard reorganization script, stop DML and DDL activity against the objects being reorganized. DML activity should be suspended until the script completes successfully.

1. **Run the Find Long LONGs procedure** in order to populate the Find Long LONGs table.

Reorg Manager checks this table for the size of LONG and LONG RAW columns. The size of these columns determines which data movement methods can be used to reorganize the column's table. See [Appendix](#) on page 204 for more information.

2. **Identify Objects That Need to Be Reorganized.**

Use the Reorg Need column in the Explorer to identify which objects need to be reorganized for each tablespace or each owner. Predicted Reorganization Benefit reports identify the benefits of reorganizing tables or indexes. See [Identify Objects That Need to Be Reorganized](#) on page 52 for more information.

3. **Select the object or objects for reorganization** in the Explorer and launch the Reorg Manager. See [Launch Reorg Manager](#) on page 69 for more information. For help selecting which reorganization method to use, see [Standard Reorganization](#).
4. **Select Target Tablespaces.** See [Select Global Reorganization Options](#) on page 70 for more information.
 - i Tip:** You can select one tablespace for tables and one for indexes. This allows you to locate tables and their indexes in different tablespaces for better I/O balancing.
5. **Select script options** in the Scripting Options page.

These options let you increase reorganization speed with Oracle's NOLOGGING mode and two types of parallel processing. See [Select Scripting Options](#) on page 72 for more information.
6. **Select a Partition Reorganization Method.**

If the reorganization includes two or more partitions or subpartitions from the same parent object, the Partition Reorg Method dialog displays. The method selected determines whether partitions or subpartitions are reorganized individually or as part of an entire partitioned object. See [Select a Partition Reorganization Method](#) on page 89 for more information.
7. **Set Storage Strategies.**

Define a global storage strategy in the Storage Strategy window. Based on your extent strategy, values for INITIAL, NEXT, and PCTINCREASE are set automatically for all objects. You can override global values for individual objects in the Customize Object Allocations window. See [Set Storage Strategies](#) on page 90 for more information.

 - i Note:** The window is only displayed when the System tablespace is data dictionary managed. For objects in locally-managed tablespaces, tablespace extent sizes and storage values are used by default.
8. **Customize Object Allocations.**

Adjust storage for individual objects in the Customize Object Allocations window. If needed, you can override extent values set on a global basis. You can also override global settings for target tablespace and sorted reorganization. See [Customize Object Allocations](#) on page 75 for more information.
9. **Review Space Usage.**

Check the Space Usage Summary tab to see if target tablespaces and the LiveReorg tablespace have enough free space for a reorganization. This window identifies tablespaces that do not have enough free space overall or enough contiguous free space. To provide free space, select different target tablespaces or adjust storage values. To provide space in the LiveReorg tablespace, add datafiles or increase datafile size with Tablespace Properties. See [Review Space Usage](#) on page 91 for more information.
10. **Generate and Run the Reorganization Script.**

Generate a reorganization script and select an execution option in the Schedule Script Execution window. See [Schedule Script Execution in Wizards](#) on page 120 for more information. The script can be run immediately from the SQL Editor or on a scheduled basis from the database server. Scheduled execution is performed by the QSA Server Agent. Scripts that use FastCopy or parallel processing by QSA must be run on a scheduled basis. See [How to Execute a Script](#) on page 117 for more information.

 - i Tip:** You can save Reorg Manager settings as a Reorg Plan to edit and reuse later. See [Save Reorg Manager Settings](#) on page 81 for more information. You can also use saved settings to automate reorganization script execution. See [About Scheduling a Recurring Reorganization](#) on page 122 for more information.
11. **Monitor Script Execution.**

Monitor script execution from the Scripts/Job Monitor or the SQL Editor. Use the Scripts/Job Monitor for scripts that are run on a scheduled basis. See [Using Job Monitor to Monitor Scripts](#) on page 129 for more information.

Smart Start for Live Reorganizations

This Smart Start outlines the workflow for live reorganizations with Reorg Manager.

i Note: Space Manager provides the best Reorg Need and Wasted space information when statistics are current in the Repository. Be sure to run the Repository Update job on regular or nightly basis (see [Change Repository Update Schedule](#)).

1. **Run the Find Long LONGs procedure** in order to populate the Find Long LONGs table.

Reorg Manager checks this table for the size of LONG and LONG RAW columns. The size of these columns determines which data movement methods can be used to reorganize the column's table. See [Appendix](#) on page 204 for more information.

2. **Identify Objects That Need to Be Reorganized.**

Use the Reorg Need column in the Explorer to identify which objects need to be reorganized for each tablespace or each owner. Predicted Reorganization Benefit reports identify the benefits of reorganizing tables or indexes. See [Identify Objects That Need to Be Reorganized](#) on page 52 for more information.

3. **Select the object or objects for reorganization** in the Explorer and launch the Reorg Manager. See [Launch Reorg Manager](#) on page 69 for more information. For help selecting which reorganization method to use, see [Standard Reorganization](#).

4. **Select the switch mode** to use.

The switch mode determines how the switch from the original table to the reorganized copy table is made. You can select to use a combination of online switch and T-Lock switch. Or you can select to use the online switch only, which excludes unsupported tables. See [Select Global Reorganization Options](#) on page 70 for more information.

5. **Select Target Tablespaces.** See [Select Global Reorganization Options](#) on page 70 for more information.

i Tip: You can select one tablespace for tables and one for indexes. This allows you to locate tables and their indexes in different tablespaces for better I/O balancing.

6. **Select script options** in the Scripting Options page.

These options let you increase reorganization speed with Oracle's NOLOGGING mode and two types of parallel processing. See [Select Scripting Options](#) on page 72 for more information.

Options that are unique to live reorganizations determine how the switch begins: automatically, upon user approval, or in a time window. They also determine whether original tables are dropped. See [More About Scripting Options \(LiveReorg\)](#) on page 86 for more information.

7. **Select a Partition Reorganization Method.**

If the reorganization includes two or more partitions or subpartitions from the same parent object, the Partition Reorg Method dialog displays. The method selected determines whether partitions or subpartitions are reorganized individually or as part of an entire partitioned object. See [Select a Partition Reorganization Method](#) on page 89 for more information.

8. **Set Storage Strategies.**

Define a global storage strategy in the Storage Strategy window. Based on your extent strategy, values for INITIAL, NEXT, and PCTINCREASE are set automatically for all objects. You can override global values for individual objects in the Customize Object Allocations window. See [Set Storage Strategies](#) on page 90 for more information.

i Note: The window is only displayed when the System tablespace is data dictionary managed. For objects in locally-managed tablespaces, tablespace extent sizes and storage values are used by default.

9. Customize Object Allocations.

Adjust storage for individual objects in the Customize Object Allocations window. If needed, you can override extent values set on a global basis. You can also override global settings for target tablespace and sorted reorganization. See [Customize Object Allocations](#) on page 75 for more information.

10. Review Space Usage.

Check the Space Usage Summary tab to see if target tablespaces and the LiveReorg tablespace have enough free space for a reorganization. This window identifies tablespaces that do not have enough free space overall or enough contiguous free space. To provide free space, select different target tablespaces or adjust storage values. To provide space in the LiveReorg tablespace, add datafiles or increase datafile size with Tablespace Properties. See [Review Space Usage](#) on page 91 for more information.

11. Choose a Scheduling Method.

In a LiveReorg, you can schedule either script execution or script generation. See [Choose a Scheduling Method \(LiveReorg\)](#) on page 78 for more information.

12. Generate and Run the Reorganization Script.

Generate a reorganization script and select an execution option in the Schedule Script Execution window. See [Schedule Script Execution in Wizards](#) on page 120 for more information. The script can be run immediately from the SQL Editor or on a scheduled basis from the database server. Scheduled execution is performed by the QSA Server Agent. Scripts that use FastCopy or parallel processing by QSA must be run on a scheduled basis. See [How to Execute a Script](#) on page 117 for more information.

i **Tip:** You can save Reorg Manager settings as a Reorg Plan to edit and reuse later. See [Save Reorg Manager Settings](#) on page 81 for more information. You can also use saved settings to automate reorganization script execution. See [About Scheduling a Recurring Reorganization](#) on page 122 for more information.

! **CAUTION:** Reorganization commands must not be modified or removed from a script. If they are changed in any way, a reorganization can fail or unpredictable results can occur. Most commands begin with QUEST_EXEC.

13. Monitor Live Reorganizations.

Monitor script execution from the Scripts/Job Monitor and its Live Reorg Detail pane. See [Monitor Live Reorganizations](#) on page 133 for more information.

- Scripts/Job Monitor—This window displays execution status and executing statements.
- Live Reorg Detail—This pane displays activity statistics for live transactions during the first three stages of a live reorganization. It also identifies which stages are complete. You can use the statistics to estimate when a table will be ready for the switch.

i **Tip:** If the switch option is **Upon User Approval**, check the Scripts/Job Monitor to see when each table in a script is ready for approval. “Waiting for approval” is displayed in the **Status Message** column (Detail pane).

14. Approve the Switch.

When a table is ready for the switch to be approved, right-click its script in the Scripts/Job Monitor (Detail pane) and select **Approve Switch**. The switch from original table to reorganized copy table is made and script execution continues. Monitor execution to see when the next table in the script is ready for approval. See [Approve the Switch](#) on page 136 for more information.

Launch Reorg Manager

You can launch the Reorg Manager and the Partitioning Wizard from multiple points in the interface. Select one or more objects to reorganize, then launch the Reorg Manager or Partitioning Wizard.

Review the following notes about selecting objects.

- When you select objects for the Reorg Manager or Partitioning Wizard, their associated objects are added to the object list. These objects include indexes, LOBs, and IOT overflows. IOTs (index organized tables) are replaced with IOT indexes.
- In most cases, related objects descend from their base objects in a hierarchical view.
- Partitions and subpartitions can be selected from all launch points. When a partition/subpartition is selected in the Partitioning Wizard, the entire partitioned object is added to the object list.

How to Launch the Reorg Manager

You can launch the Reorg Manager from several points in the Space Manager workspace: from the Explorer, from the Advanced Search window, and from certain reports. You can select a single object or multiple objects to reorganize simultaneously (in most cases).

Step 1: Select objects to reorganize

- After identifying the object (or objects) in need of reorganization (see [Identify Objects That Need to Be Reorganized](#)), select the object/objects from one of the following areas in the Space Manager workspace:

Selection method	Description
Select objects from the Explorer window	Use the following techniques to find and select objects in the Explorer window: <ul style="list-style-type: none">• One or more objects from the Top Issues list for the selected database.• One or more objects in a database. Use incremental Search and Group By fields to filter objects. Or use the Advanced Search window to find objects using multiple criteria.• One or more objects in one tablespace. Select from the Segment grid.• One or more objects belonging to one owner. Select from the object grid.• One or more objects in a datafile. Use the Datafile Map to find Segments with blocking extents.
Select objects from a report	Select an object from one of the following report types: <ul style="list-style-type: none">• Reorganization Reports• Reorganization Candidates Reports

Step 2: Launch the Reorg Manager

1. Use one of the previous methods to select one or more objects in the Explorer window or in a report.
2. Right-click the object/objects and select **Reorg Manager**.
3. Where applicable, select **LiveReorg** or **Standard Reorg**.

- The Reorg Manager opens to the Select Target Tablespaces page. See [Select Global Reorganization Options](#) for next steps.

i Note: Before you can perform live reorganizations with Space Manager, the LiveReorg option must be licensed. In addition, the QSA Server Agent must be installed, current, and running.

Objects to be Reorganized - Color-Coding

In the **Objects to be Reorganized** list, color is used to differentiate certain objects. Use the following descriptions to identify objects by text color.

Text Color	Description
Black	In most cases, object names are displayed in black when objects are selected by the user.
Blue	Object names are displayed in blue when objects are selected by Reorg Manager. The objects selected by Reorg Manager are those that are related to objects selected by the user. Related objects include indexes and LOBs. They are reorganized along with their base objects.
Grey	Object names are displayed in gray when objects cannot be modified or when they are excluded from reorganization by you or Space Manager. The names of LOB indexes, LOB index partitions, and LOB index subpartitions are always displayed in gray. This is to indicate that storage values cannot be set for these objects even though they are included in a reorganization along with their base tables, partitions, or subpartitions. (The LOB index clause is not displayed in reorganization scripts.) i Note: For all reorganization methods, you can exclude objects from the Customize Object Allocations window. For standard and live reorganization methods, you can also exclude objects in the Select Partition Reorganization Method window.

Specify Options - Reorg Manager and Partitioning Wizard

Select Global Reorganization Options

Use the first page of the Reorg Manager to specify target tablespaces and to specify the table switch style (Live Reorg only).

You can specify one target tablespace to be used for all tables, one to be used for all indexes, and one to be used for all LOBs. If necessary, you can override this setting for individual objects in the Custom Object Allocations page of the Reorg Manager.

i Tip: You can save Reorg Manager settings as a Reorg Plan to edit and reuse later. See [Save Reorg Manager Settings](#) on page 81 for more information. You can also use saved settings to automate reorganization script execution. See [About Scheduling a Recurring Reorganization](#) on page 122 for more information.

Select table switch mode (LiveReorg only)

- If you are creating a LiveReorg script, on the first page of the Reorg Manager (see [Launch Reorg Manager](#) for more information) in the **LiveReorg Table Switch Mode** section, select the switch mode (style). Switch mode determines how the switch from the original table to the reorganized copy table is

made.

- To use the default setting, which is a combination of online and T-Lock switch, take no action.
 With the default setting, the switch mode is determined dynamically. The online switch is used for all tables supported for this switch style. The T-Lock (trigger lock) switch style is used for objects that are supported for LiveReorg, but not supported by the online switch. See [What is Not Supported for Live Reorganizations](#) for a list of objects not supported by the online switch.
 For more information about switch styles, see [Switch Mode](#).
- To use the online switch only, select **Exclude tables requiring the T-Lock switch (online switch only)**.
 Select this mode to use the online switch for all tables supported for this switch style. Unsupported objects are excluded from a reorganization. When the online switch is used, the table being reorganized remains available during the switch.
 See [What is Not Supported for Live Reorganizations](#) for a list of objects not supported by the online switch.

Select target tablespaces

1. On the first page of the Reorg Manager (see [Launch Reorg Manager](#) for more information), use the **Target Tablespaces** section to select whether or not to move objects to another tablespace. You can also use this section to specify a different tablespaces to use during reorganization (Standard Reorg only). For more information about target tablespaces, see [About Selecting Target Tablespaces](#).

Review the following for additional information:

Tables	<p>Leave in Original Tablespace—Select to leave tables in their current tablespace.</p> <p>Reorganize In Place—(Standard Reorg only) Select this option to reorganize tables using space in their current tablespace.</p> <p>Use Interim Tablespace—(Standard Reorg only) Select this option to leave tables in their current tablespace but reorganize them using free space in another tablespace. Then select the other tablespace. Tables are copied to the current tablespace after they are reorganized.</p> <p>Relocate to Tablespace—Select to move tables to a different tablespace. Then select the tablespace from the list.</p>
Indexes	<p>Leave in Original Tablespace—Select to leave indexes in their current tablespace.</p> <p>Relocate to Tablespace—Select to move indexes to a different tablespace. Then select the tablespace from the list.</p>
LOBs and LOB Indexes	<p>Leave in Original Tablespace—Select to leave LOBs and LOB indexes in their current tablespace.</p> <p>Reorganize In Place—(Standard Reorg only) Select this option to reorganize LOBs using space in their current tablespace.</p> <p>Use Interim Tablespace—(Standard Reorg only) Select this option to leave LOBs in their current tablespace but reorganize them using free space in another tablespace. Then select the other tablespace. LOBs are copied to the current tablespace after they are reorganized.</p> <p>Relocate to Tablespace—Select to move LOBs and LOB indexes to a different tablespace. Then select the tablespace from the list.</p>
<p>In the Relocate to Tablespace field, the list displays names and free space for all tablespaces in a database except offline and read-only tablespaces.</p>	

i Tip: Select the **Use Interim Tablespace** option to defragment a tablespace during a standard reorganization. A tablespace is defragmented when all of its objects are included in a reorganization.

Specify FastCopy (DataPump) data movement method

1. On the first page of the Reorg Manager, the **FastCopy (DataPump)** section allows you to specify *FsCopy* or *DataPump* as the only data movement method.

- **If this option is selected.** If selected, FsCopy/DataPump is used for all tables that are supported for this method. Tables that are not supported are excluded from reorganization.
 - **FsCopy**—For Oracle database 9, 10, or 11.
 - **DataPump**—For Oracle database 12.
- **If this option is cleared.** (Recommended) If this option check box is cleared, Space Manager automatically selects the most efficient data movement method, either SQL or FastCopy/DataPump, to use for each object to be reorganized.

i Caution: Care should be taken when selecting this method. FsCopy and DataPump are not always the most efficient data movement methods and should be used only for large tables where table copy might cause Oracle errors.

See [About Data Movement Methods](#) for more information about this option.

2. Click **Next** to proceed to the next page. For next steps, see [Select Scripting Options](#).

After proceeding to the next page of the wizard

View any messages that display when you proceed to the next window. One message identifies objects that Reorg Manager is excluding. (Their names are displayed in gray.) Another message identifies tables with constraints that can be validated for use in posting live transactions. See [Respond to Checks for a Posting Index](#) on page 114 for more information. Reorg Manager excludes the following:

- Objects that are not supported for live reorganization.
- Objects that have components or related objects that are not supported for live reorganizations.
- Indexes that were selected for reorganization without their tables. Indexes can only be reorganized on a live basis when their tables are reorganized.
- Objects that are not supported for the online-only switch style. To include these objects, clear the **Exclude tables requiring the T-Lock switch (online switch only)** option.

Select Scripting Options

You can use the **Scripting Options** page of the Reorg Manager and Partitioning Wizard to increase reorganization speed or to select the switch on approval option.

To select scripting options

1. After specifying options on the first page of the Reorg Manager, click **Next** to open the Scripting Options page.
 - Or in the Partitioning Wizard, after reviewing the Partition Names page, click **Next**.
2. In the Scripting Options page, specify parallel query, dependency, and switch options. Review the following for additional information:

Tables	Parallel Query Option —Use parallel query options to speed up reorganization with Oracle's parallel processing. This type of parallel
---------------	--

	<p>processing uses multiple server processes to reorganize each table in a script.</p> <p>Insert—This option lets you parallelize the INSERT statements used to insert data into copy tables. When CREATE TABLE AS SELECT is the SQL used, the Insert option parallelizes creation of the copy table and insertion of data.</p> <p>Select—This option lets you parallelize the SELECT statements used to select data from original tables.</p> <p>The maximum number of parallel processes specified should be no more than the number of CPUs in your system. The maximum number Oracle uses is determined by its PARALLEL_MAX_SERVERS initialization parameter.</p> <p>Unrecoverable—Select this option to speed up reorganization using Oracle’s NOLOGGING mode. This increases reorganization speed by turning off writing to redo logs as tables are recreated. Immediately after a NOLOGGING reorganization, you should back up reorganized tables.</p> <p>Use Alter Table Move where possible—(Standard Reorg only) Select to use the ALTER TABLE...MOVE statement with the SQL data-movement method during a standard reorganization. It moves table data into a new segment in one move. Constraints and dependencies do not need to be recreated or recompiled.</p> <ul style="list-style-type: none"> • When a table is reorganized with ALTER TABLE...MOVE, its indexes can be recreated or rebuilt with the Rebuild where possible option.
<p>Indexes</p>	<p>Use Parallel Query Option—Use parallel query options to speed up reorganization with Oracle’s parallel processing. See Parallel Processing for Indexes on page 75 for more information.</p> <p>Create <i>n</i> indexes at the same time—Enter a number from two to 20 to use QSA parallel processing. See Parallel Processing for Indexes on page 75 for more information.</p> <p>Unrecoverable—Select this option to speed up reorganization using Oracle’s NOLOGGING mode. This increases reorganization speed by turning off writing to redo logs as indexes are recreated. Immediately after a NOLOGGING reorganization, you should back up reorganized tables and their indexes.</p> <p>Rebuild where possible—(Standard Reorg only) Select to specify ALTER INDEX...REBUILD as the SQL to use for non-partitioned indexes. Rebuilding an index is faster than dropping and recreating it.</p> <ul style="list-style-type: none"> • Rebuild ONLINE—(Standard Reorg only) Indexes can be rebuilt offline or online. When an index is rebuilt online, users can update its table throughout the rebuild process. Select this option if you want to allow DML access to associated tables.
<p>Compile Dependencies</p>	<p>Select this option to compile dependencies on the tables being reorganized.</p> <ul style="list-style-type: none"> • Compile <i>n</i> objects at the same time—Select this option to speed up the compile process with parallel processing by QSA. <p>How much time is required to recompile dependencies depends on the number of dependencies, how busy the database is, and system</p>

	<p>performance. The time required can range from seconds to hours.</p>
Drop original table after successful reorganization	<p>(LiveReorg and Partitioning Wizard only) This checkbox is selected by default so that original tables and indexes are dropped after they are replaced with their reorganized copies. In Oracle 10g databases, original objects are also purged from the Recycle Bin.</p> <p>Do not select this option if you want to save original tables and indexes. Original objects are preserved in their original tablespaces under modified names.</p> <p>See More About Scripting Options (LiveReorg) on page 86 for more information.</p>
Switch Options	<p>(LiveReorg only) See More About Scripting Options (LiveReorg) on page 86 for more information about switch options.</p> <p>Automatically—Select if you want the switch to be made automatically, as soon as a copy table is ready. When multiple tables are included in one reorganization script, the switch is made automatically for each table in turn.</p> <p>Upon user approval—Select if you want the switch to be made after a user approves the switch. Approval should be given as soon as possible after a table is ready for the switch. Until approval is given, QSA keeps the original table in sync with the copy table.</p> <p>When multiple tables are included in a reorganization script, multiple approvals must be given.</p> <p>Approval is given from the Scripts/Job Monitor. See Approve the Switch on page 136 for more information.</p> <p>Between date/time—Use this option if you want the switch to be made within a time window. Then specify a start date/time and end date/time for the window. The switch is made automatically for all tables that are ready for the switch during the time window specified.</p> <p>This option allows the switch to proceed automatically at a time when activity is light.</p> <p>i Note: If the window ends before the switch is made for all tables in a script, execution aborts with this message: “QSA-20502: Target completion date/time expired.” This happens after 20 “retry operation” attempts are made.</p> <p>i Tip: After a live reorganization script has been stored and scheduled, you can select a different switch option, if needed. See Change Approval Method for Online Scripts on page 135 for more information.</p>
Analysis	<p>Collect statistics for the CBO—Select to update Oracle's cost-based optimizer (CBO).</p> <p>Do not collect statistics for the CBO—Select if you do not want to update Oracle's cost-based optimizer (CBO).</p> <p>Restore original statistics in data dictionary for the CBO—(Reorg Manager only) Select this option to preserve existing statistics in the Oracle data dictionary instead of collecting statistics.</p> <p>i Note: Statistics are collected using DBMS_STATS. To collect statistics or restore the data dictionary, the DBMS_STATS package must be installed for the database.</p> <p>See About Statistics Collection Options on page 91 for more information about these options in the Reorg Manager.</p>

Segments being reorganized	Provides information about the objects to be reorganized.
-----------------------------------	---

i Note: Scripting options for tables are not used for FastCopy (or DBMS_DataPump).

i Important: Parallel and NOLOGGING scripting options do not change the permanent parallel and logging attributes for an index. These options are only used during reorganization.

3. Click **Next** to proceed to the next page. For next steps, see [Customize Object Allocations](#).

Parallel Processing for Indexes

There are two types of parallel processing you can use for indexes, and they can be used together. Each works in a different way:

- Oracle parallel processing uses multiple server processes to reorganize one index at a time. All are applied to the same index. If degree of parallelism for a reorg is four, all four parallel processes are applied to the first index being reorganized. All four are then applied to the next index being reorganized, and so on. To use this option, select **Use Parallel Query Option**.
- QSA Server Agent parallel processing uses multiple processes to reorganize multiple indexes at the same time. One QSA process is used per index. If three QSA processes are being used, one process is applied to the first index in a series of indexes, one is applied to the second index in the series, and one is applied to the third index in the series. To use this option, select **Create *n* indexes at the same time**.

What happens when the two types of parallel processing are used together? Multiple QSA processes are used to reorganize multiple indexes simultaneously while multiple Oracle parallel query processes are applied to each index. Keep in mind that using the two together increases the number of CPUs needed for a reorganization.

To calculate the number of CPUs needed for parallel processing of indexes, multiply the number of QSA processes specified by the number of parallel server processes that will be used by Oracle (this is twice the number you specify plus one). For example, if you specified 3 QSA processes and 7 Oracle processes will be used by Oracle, 21 CPUs will be needed for reorganizing indexes.

Customize Object Allocations

Use the Customize Object Allocations page of the Reorg Manager or Partitioning Wizard to specify options for individual objects.

In the Physical Attributes sections, current values are displayed in gray beside entry fields for new values.

i Note: To see the space currently allocated for an object, as well as the post-reorganization estimate, check the Segment Allocation section.

Names are displayed in gray for objects on the Objects to Reorganize list that are not included in a reorganization. Typically, these objects are related to included objects.

i Note: The Partitioning Wizard does not display fields for an object if its partitioning is not changed. Instead, it displays a message: "This functionality does not apply to the currently selected object."

To customize object allocations

1. After specifying options in the Scripting Options page, click **Next** to open the Customize Object Allocations page.
2. In the Customize Object Allocations page, specify storage and space use options for individual objects as needed. You can also override target tablespace settings for an object.

Select an object in the Objects to Reorganize list. Then customize its settings. Review the following for additional information:

Object Properties	
Space Saving tab	<p>Use these fields to modify physical attributes and compression methods.</p> <p>For Tables</p> <p>Compress—Select to use compression. Then select a compression method.</p> <p>Row Store Compress—(Oracle 12c)</p> <ul style="list-style-type: none"> • Basic—Select to use ROW STORE COMPRESS BASIC to enable basic table compression. • Advanced—Select to use ROW STORE COMPRESS ADVANCED to enable Advanced Row Compression. <p>Column Store Compress—(Oracle 12c) Enabled only if underlying storage supports Hybrid Columnar Compression (such as on Oracle Exadata).</p> <ul style="list-style-type: none"> • Query—Select to use COLUMN STORE COMPRESS FOR QUERY to enable query compression. • Archive—Select to use COLUMN STORE COMPRESS FOR ARCHIVE to enable archive compression. <p>For Indexes</p> <p>Compress—Select to use compression. Then select a prefix length.</p> <ul style="list-style-type: none"> • Advanced Low—(Oracle 12c) Select to use Advanced Index Compression. Available only if supported in the underlying Oracle database. <p>i Note: Pct Free reserves a percentage of data block space for updates to existing rows. For example, if the PctFree setting is 10, 10% of each of an object's data block is reserved for UPDATES. The space left (minus block overhead) is the space available for INSERTS of new rows and their overhead. Set a higher value for PctFree if existing rows will be updated heavily and updated values will increase row size. This will help prevent chaining of growing rows to other data blocks.</p>
Detail tab	<p>Use these fields to modify logging, parallelism, indexing, and other object properties.</p> <p>Indexing—(Oracle 12c)</p> <ul style="list-style-type: none"> • For Tables. Use this field to specify which partitions are indexed by turning INDEXING <i>ON</i> or <i>OFF</i>. The table-level setting can be overwritten by each partition-level setting. • For Indexes. Use this field to indicate whether the selected index is <i>PARTIAL</i> or <i>FULL</i>.
In-Memory	<p>Use these fields to enable or modify In-Memory Column Store options.</p> <p>In-Memory—Select this option to enable In-Memory Column Store for the selected object.</p> <ul style="list-style-type: none"> • MemCompress—Select to use in-memory compression. Then select the level of compression to use.

	<ul style="list-style-type: none"> • Priority—Select a priority to use for this table when populating tables in the IM column store. • Distribute—Specify how the table is distributed among Oracle RAC instances. • Duplicate / All—Select to duplicate data on another or on all Oracle RAC instances. <p>Column Clause—Select to specify In-Memory Column Store options per column.</p> <p>Note: In a Live Reorg, the column clause is only supported if the T-Lock switch is used.</p>
Attribute Clustering	<p>Use these fields to apply or modify attribute clustering.</p> <p>Attribute Clustering—Select this option to apply attribute clustering to the reorganized table.</p> <ul style="list-style-type: none"> • Specify the type of attribute clustering to apply: Linear Order or Interleaved Order. • Select the operations for which clustering is triggered: on Load and/or on Data Movement. <p>Columns—Click to specify one or more columns on which to base clustering.</p> <ul style="list-style-type: none"> • Add Group—Click to specify a column group if necessary. Enabled for Interleaved Order clustering. <p>Joins—Click to specify one or more joined columns on which to base clustering (join attribute clustering).</p>
LOB tab	<p>This tab only displays if you have LOB columns.</p> <p>SecureFile—Select this option to store LOBs using the SecureFile storage method and to enable additional SecureFile options from which to select.</p> <p>Note: Oracle only stores data in the row if the data is less than 4K. If the data is greater than 4K, then the LOB value is moved into a LOB segment.</p>
Deprecated tab	<p>Use this tab to modify physical attributes and storage options applicable to features used in older versions of Oracle database, such as for Dictionary-Managed tablespaces.</p> <p>If you change values for Initial, Next, and PctIncrease, this overrides global values.</p> <p>Note: Oracle ignores the PCTUSED, FREELISTS, and FREELIST GROUPS parameters for objects in locally managed tablespaces that use automatic segment space management (ASSM).</p>
Encryption tab	<p>This tab only displays if you have encrypted objects. It is recommended that you provide the encryption password to ensure that the reorganization is successful.</p> <p>Note: If you use the exchange partition or online switch to reorganize a table that uses Transparent Data Encryption (TDE), you must supply the encryption password. If you did not provide a password when you created the table, you cannot reorganize the table with methods.</p>

Tablespace	Use this section to override the global tablespace specification if necessary. See Select Global Reorganization Options on page 70 for more information.
Sorted Reorg	Perform sorted reorg —Select this option if you want to sort a table's data by a certain index. Then select the index to sort by from the Index to Use field.
Add Index	(Reorg Manager) To add an index, select a table and click Add Index . Once you add an index, you can modify or drop it from the Customize Object Allocations page. i Note: If you want to create a partitioned index, it must be locally partitioned. You cannot create a globally partitioned index. To modify or drop an index, select the index and click Modify Index or Drop Index .
Long LONG Columns	Indicates whether or not the object includes one or more long LONG columns.
Segment Allocation	Use this section to view current allocation, as well as post-reorganization allocation estimates.

- Click **Next** to proceed to the next page.
 - For a LiveReorg, see [Choose a Scheduling Method \(LiveReorg\)](#) for next steps.
 - For a Standard Reorg or for the Partitioning Wizard, see [Schedule Script Execution in Wizards](#) for next steps.

Choose a Scheduling Method (LiveReorg)

The Choose Scheduling Method page displays for LiveReorgs only. This page allows you to select one of the following options for scheduling script execution.

- To run, to schedule (to run once), or to store the script, choose **Schedule Script Execution**.
- To schedule a script for recurring execution, choose **Schedule Script Generation**.

Scheduled script generation allows you to save the Reorg Manager settings as a Reorg Plan which is then used to generate and execute the script at a later time. A Windows scheduler task is created to run the Reorg Plan. Scheduled script generation is available for LiveReorg only. See [About Scheduling a Recurring Reorganization](#) to learn about the advantages of this method.

To run the script immediately, select **Schedule Script Execution**.

Prior to scheduling the script, review space usage information on the **Space Usage Summary** tab. See [Review Space Usage](#) on page 91 for more information.

Select a scheduling method

- After specifying options in the Customize Object Allocations page for a LiveReorg, click **Next**.
- In the Choose Scheduling Method page that displays, select one of the following options:
 - Schedule Script Execution**—Select to execute the script immediately, to store the script without executing, or to schedule the script.

Selecting this option opens the Schedule Script Execution page. See [Schedule Script Execution in Wizards](#) for next steps.
 - Schedule Script Generation**—Select to save the Reorg Manager settings as a Reorg Plan and

schedule script generation and execution. Use this method if you want to schedule script execution on a recurring basis. This method also allows you to schedule a script to run once. You can modify the execution schedule in the Space Manager Scheduled Tasks window.

When this method is used, the script is executed immediately after the script is generated.

Selecting this method opens the **Schedule Script Generation** page. See [Schedule Script Generation](#) for next steps.

Schedule Script Execution in Wizards

Use the Schedule Script Execution page of the Reorg Manager or Partitioning Wizard to run, schedule, or store a script. You can also open the script in the Editor to review it (or run it).

Prior to scheduling the script, review space usage information on the **Space Usage Summary** tab. See [Review Space Usage](#) on page 91 for more information.

From the **Schedule Script Execution** page you can do any of the following:

- View (or run) the script in the Editor
- Schedule the script to run once
- Execute the script immediately (schedule script to run immediately)
- Store the script without scheduling it

i Note: For LiveReorg scripts, if you want to schedule the script to run on a recurring basis, return to the Choose Scheduling Method page and select **Schedule Script Generation**. See [Choose a Scheduling Method \(LiveReorg\)](#) and [About Scheduling a Recurring Reorganization](#) for more information.

See [How to Execute a Script](#) to learn more about your script execution options and to review important considerations and best practices.

To view (or run) the script in the Editor

- Click **View in Editor** to open the script in the Editor window to review it before you schedule it.
- You can also run the script interactively from the Editor window. See [Run Scripts from the Editor](#) on page 121 for more information.

i Important: Live reorganization scripts and scripts that use FastCopy or parallel QSA processes cannot be run from the SQL Editor. They must be scheduled and run by QSA. These scripts contain commands that must be interpreted by QSA. To run a script in QSA, it must be scheduled (even when executed immediately).


To schedule the script to run once

1. Select **Run at** to run the script at a certain date and time.

i Note: Reorganization scripts should be run as soon as possible after they are generated. If the data structures of the objects in the script are modified between script generation and execution, the DDL and DML statements in the script can cause errors and/or data loss during script execution.

See [Schedule Scripts](#), to learn more about scheduling scripts.


2. Select a date and time at which you want the scheduled script to run.
3. Enter a name for the script.
4. Click **Submit**. Space Manager stores the scheduled script and displays a confirmation message.
5. Click **Yes** in the prompt to open the Script/Job Monitor (or select **Manage | Job/Script Monitor**).

6. Your script displays in the list of scripts/jobs in the Job Monitor. If not, click  to refresh the list.
7. Use the Script/Job Monitor window to manage your script. You can reschedule, execute, or delete the script. You can monitor status of the script or change the approval option (automatic or wait for approval). You can open the script or view script logs. See [Using Job Monitor to Monitor Scripts](#) on page 129 for more information.
8. After the script is stored, you can close the Reorg Manager or Partitioning Wizard.

To execute the script immediately

1. Select **Execute immediately** to run the script immediately, as soon as it is stored in the Space Manager repository.
2. Enter a name for the script.
3. Click **Submit**. The script is scheduled to run immediately by QSA. Monitor script execution from the Scripts/Job Monitor.

To store the script without scheduling it

1. Select **Store without scheduling** to store the script and schedule it later from the Script/Job Monitor.
2. Enter a name for the script.
3. Click **Submit**. Space Manager displays a confirmation message that the script is successfully stored.
4. Click **Yes** in the prompt to open the Script/Job Monitor (or select **Manage | Job/Script Monitor**).
5. Your script displays in the list of scripts/jobs in the Job Monitor. If not, click  to refresh the list.
6. To schedule the script, right-click the script and select **Schedule**. Select a date and time to execute the script.
 - Or you can right-click the script and select **Properties**. Then select one of the Scheduling Info options in the Script Attributes tab.

i Tip: Click **Auto-Refresh** in the Script/Job Monitor window to specify a frequency at which to automatically refresh the scripts list.

Schedule Script Generation

Scheduled script generation allows you to save the Reorg Manager settings as a Reorg Plan which is then used to generate and execute the script at a later time. A Windows scheduler task is created to run the Reorg Plan. Scheduled script generation is available for LiveReorg only. See [About Scheduling a Recurring Reorganization](#) to learn about the advantages of this method.

Selecting this option is done through in the Reorg Manager when creating a LiveReorg script. See [Choose a Scheduling Method \(LiveReorg\)](#) on page 78 for more information.

To Schedule Script Generation

1. To schedule recurring script execution, in the Choose Scheduling Method page select **Schedule Script Generation**. This allows you to set up a recurring task to generate the script and then execute it.
2. In the Schedule Script Generation page, click **Save Reorg Plan** to save the Reorg Manager settings as an XML file. See [Save Reorg Manager Settings](#) for additional information about Reorg Plans.
3. Enter a file name or use the default name. The name you enter is used as the name for the Windows scheduler task as well.

i Tip: After saving a Reorg Plan, you can open it in the Reorg Manager at any time using **File | Open Saved Reorg Plan**.

4. After saving the Reorg Plan, click **Schedule** to schedule script generation. This process creates the Windows scheduler task.
5. In the Schedule Task window, select an interval or select **One time only**. Click **Next**.
6. Specify schedule details for the recurring task, or specify a date/time for the one-time-only task.
7. Before you run the script for the first time, to manage it you must use the Space Manager Scheduled Tasks window. To open this window, select **Manage | Scheduled Tasks** from the Reorg Wizard or Explorer. You can view the status of a scheduled script, edit the script's schedule, enable/disable the script, run the script, or delete the scheduled execution. See [Manage Scheduled Tasks](#) on page 154 for more information.

If the script requires approval, once the script begins to run, you must open the Script/Job Monitor window to approve the switch. However, you should use the Space Manager Scheduled Tasks window if you need to edit the script's schedule.

Save Reorg Manager Settings

Space Manager provides two ways to save Reorg Manager settings: Save as Default or Saved Reorg Plan.

Saved Reorg Plan

Space Manager allows you to save the settings you select in the Reorg Manager as a Reorg Plan to reuse later. You can save the settings at any stage in the wizard. Saving Reorg Manager settings allows you to save the reorganization parameters selected on each page of the Reorg Manager wizard, as well as the Objects to be Reorganized list. This feature is also available in the Partitioning Wizard.

To use a saved Reorg Plan, open it from the File menu. The Reorg Manager then launches with the saved settings and object list applied. You can have multiple saved Reorg Plans.

i Tip: To save an unfinished reorg preparation in the Reorg Manager or Partitioning Wizard and continue it later, save it as a Reorg Plan.

Save As Default

You can also save the settings in the Reorg Manager or Partitioning Wizard to use as the default settings. Once you save the settings as default, Space Manager displays these settings each time you use the Reorg Manager or Partitioning Wizard.

To save a Reorg Plan

1. You can save a Reorg Plan by clicking the **Save Reorg Plan** button found in the following locations:
 - On any page of the Reorg Manager (LiveReorg or Standard Reorg).
 - On the Schedule Script Generation page of the Reorg Manager (LiveReorg only). A Reorg Plan is used when you schedule script generation. See [Schedule Script Generation](#) on page 80 for more information.
 - On any page of the Partitioning Wizard.
2. Enter a name for the file. The Reorg Plan is saved as an XML file.

To open a saved Reorg Plan

1. Select **File | Open Saved Reorg Plan**.
2. Select a previously-saved Reorg Plan (.xml). Click **Open**.
 - If the settings were saved from the Reorg Manager, the Reorg Manager opens with the saved settings applied automatically.
 - If the settings were saved from the Partitioning Wizard, the Partitioning Wizard opens with the saved settings applied automatically.

3. You can use the settings "as is" or modify them further before saving, executing, or scheduling the script.
4. To edit a saved Reorg Plan, open the plan (**File | Open Saved Reorg Plan**), modify the settings, then save the plan using the same file name.

To Save (settings) as Default

1. In any applicable page of the Reorg Manager or Partitioning Wizard, click **Save As Default**.
2. The current settings are saved and will be applied each time you launch the Reorg Manager or Partitioning Wizard from anywhere except a saved Reorg Plan. Saved settings in the Reorg Plan override the default settings.

i Tip: In addition to the two methods described in this topic, you can also specify some default options for the Partitioning Wizard in the Options dialog (**Tools | Options | Partitioning Wizard**). See [Partitioning Wizard Defaults](#).

Run Scripts from the Editor

Use the **View in Editor** option in the Reorg Manager (or Partitioning Wizard) to run reorganization scripts interactively from the SQL Editor. As a script runs, you can monitor execution directly from the SQL Editor. You can also use the **View in Editor** option to open a script and review it before you schedule it.

i Important: Live reorganization scripts and scripts that use FastCopy or parallel QSA processes cannot be run from the SQL Editor. They must be scheduled and run by QSA. These scripts contain commands that must be interpreted by QSA. To run a script in QSA, it must be scheduled (even when executed immediately).

To run scripts from the Editor

1. Select **View in Editor** in the schedule window of Reorg Manager or the Partitioning Wizard. The script is displayed in the SQL Editor.
2. Run the script from the SQL Editor in continuous mode or single-step mode:
 - To run the entire script in continuous mode, click **Run** in the toolbar or press **F9**.
 - To run the script in single-step mode (one statement at a time), click **Single Step** in the toolbar or press **F8** from the first statement. Repeat for each statement in the script.
 - i Note:** If the Connection dialog displays, enter the user name and password of an Oracle user with the DBA role or privileges for Space Manager. Then click **OK** to connect to your database.
3. As the script is executed, the SQL Editor scrolls through script statements. The current statement is highlighted. Check the status bar at the bottom of the window. If a statement is executed successfully, processing time is displayed. If an error occurs, an error message is displayed.
4. When script execution is finished, an execution log for all script statements displays in the Spool tab. View this to see script start and stop times, any errors that occurred, and the output of all executed statements.

More Options for Reorg Manager

About Selecting Target Tablespaces

Target tablespace options determine where objects are reorganized and whether they are relocated during reorganization. Objects can be left in their current tablespace or relocated to another tablespace.

When objects are left in place during a standard reorganization, they can be reorganized using free space in the current tablespace or free space in another tablespace on an interim basis. (Interim tablespaces are not used for live reorganizations.)

See also [Target Tablespace Considerations](#) for more information about how data movement method, space availability, and speed can factor into choosing a target tablespace.

Specify Target Tablespace Globally or Per Object

Target tablespaces are selected on a global basis in the Target Tablespaces section in the Reorg Manager (see [Select Global Reorganization Options](#)). This means they are used for all tables and indexes included in a reorganization. However, you can override global target tablespace for individual objects in the Customize Object Allocations page. See [Customize Object Allocations](#) on page 75 for more information.

One global target tablespace can be selected for tables and another selected for indexes. This allows you to locate tables and their indexes in different tablespaces in order to avoid table/index contention and improve I/O balancing. You can also select one global target tablespace to use for all LOBs and LOB indexes.

Target Tablespace Considerations

- Global target tablespaces for IOTs—The global target tablespace selected for tables is also used for IOT overflows, and IOT overflow partitions.
- Target tablespaces for partitioned objects—All tablespaces chosen for a partitioned object and its partitions or subpartitions must use the same data block size. This is an Oracle requirement.
- Defragmenting a tablespace—If you want to defragment a tablespace, include all of its objects in a standard reorganization. Also select the Use Interim Tablespace option.

When to Leave Objects in Their Current Tablespace

Use the **Leave in Original Tablespace** option to leave tables, indexes, and LOBs in their current tablespace during a reorganization. In a Standard Reorg, when you leave tables or LOBs in their current tablespace, you can choose whether to reorganize them using space in the current tablespace or space in another tablespace on an interim basis.

When you reorganize using space in the current tablespace, the following space requirements apply:

- For tables without LONGs, the tablespace must have enough space for the largest table being reorganized and a copy of that table. It must also have enough space for the largest set of indexes being reorganized and copies of those indexes.
- For a Standard Reorg on supported tables with LONGs, the filesystem must have enough space for exported data.
- For a LiveReorg on tables where a T-Lock switch is used, the filesystem must have enough space for exported data.

When you reorganize using space in an interim tablespace, the tablespace must have enough concurrent free space for all copy tables it will contain. All are created at the same time.

When to Relocate Objects to a New Tablespace

Use the **Relocate to Tablespace** option to move tables and indexes to a different tablespace during reorganization. The tablespace must have enough free space for copies of all tables and indexes to be relocated there.

Relocating objects when you reorganize optimizes database layout for efficient use of disk space, better I/O balancing, and improved query performance. For example, separating tables and indexes into different tablespaces distributes I/O requests among datafiles and disks for improved performance of index-based queries.

Best Practices for Locally Managed Tablespaces

Use the following guidelines to relocate tables to locally managed tablespaces:

- If you can accurately predict growth rate for the tables, consider moving them to locally managed tablespaces that use the uniform extent allocation type. Ideally, the uniform extent size should be large enough to prevent allocation of an excessive number of extents for the tables.
- If you cannot predict growth rate for the tables, consider moving them to locally managed tablespaces that use the autoallocate extent allocation type.

Best Practices for SAP Objects

SAP objects should only be relocated to tablespaces that were created with SAP administrative tools. After relocating an SAP object with Space Manager, you should update the SAP ABAP Dictionary. This is recommended in SAP note 154193, "SAPDBA:reorganizations and ABAP-DDIC".

To update the ABAP Dictionary, run the following SQL statement:

```
update sapr3.dd091
set tabart = (select tabart
              from sapr3.taora
              where  tabspace = '<tsp>')
where tabname = '<table>'
and as4local = 'A';
```

- i Important:** Check SAP notes for the latest recommendations on reorganizing SAP objects using external applications.

Target Tablespace Considerations

As you select target tablespace options for a reorganization, consider the object types and datatypes being reorganized, the space available in the database and filesystem, the data movement method used, and the speed of different combinations of options.

Object type and datatype determine which data movement method can be used. If both SQL and FastCopy are used, space and speed can be considered in choosing target tablespace options.

Space Requirements

Space availability and data movement method are a factors to consider when selecting target tablespace options:

- **SQL reorganizations** use space in the database. Target tablespace options allow you to reorganize objects in their current tablespace or relocate them to a different tablespace. If you choose to leave objects in their current tablespace, you can reorganize them using free space in that tablespace or free space in another tablespace on an interim basis (objects are moved back to their current tablespace during the reorganization process).
- **FastCopy reorganizations** use space in the database or in the filesystem. Where space is used depends on target tablespace options. If you choose to leave tables in their current tablespace, FastCopy exports data to the filesystem. If you choose to relocate tables to a different tablespace, FastCopy copies data inside the database.

Target Tablespace	Space Required for SQL	Space Required for FastCopy (DBMS_DataPump)
Current	Space in current tablespace for original tables and indexes plus space for copy of largest table and its indexes.	Concurrent space in filesystem for exported data of all tables being reorganized with FastCopy/DBMS_

Target Tablespace	Space Required for SQL	Space Required for FastCopy (DBMS_DataPump)
		DataPump.
Interim	Concurrent space in interim tablespace for copies of all tables being reorganized in that tablespace.	Interim tablespaces are not used by FastCopy.
New	Space in new tablespace for copies of all tables and indexes being relocated to that tablespace.	Space in new tablespace for copies of all tables and indexes being relocated to that tablespace.

Speed Guidelines

Speed is another factor to consider when selecting target tablespace options. Different combinations of options vary in terms of the number of data moves required. Data moves can be one-way or two-way:

- One-way moves are used for SQL reorganizations in the current tablespace and for SQL and FastCopy relocations to a new tablespace. Data is moved once—from the original table to a copy table (in the current tablespace or a new tablespace). A one-way tablespace move with any data movement method is faster than a two-way move with the same method.
- Two-way moves are used for SQL reorganizations in an interim tablespace and FastCopy reorganizations in the current tablespace. For SQL, the first move is from original table to an interim tablespace. The second move is back to a copy table in the original tablespace. For FastCopy, the first move is an export from the original table to the filesystem. The second move is an import back to a copy table in the original tablespace.

Rank	Data Movement Method	Target Tablespace
1	ALTER TABLE... MOVE (SQL)	Current or new
2	ALTER TABLE... MOVE (SQL)	Interim
3	CREATE TABLE AS SELECT (SQL) with Unrecoverable Mode option	Current or new
4	CREATE TABLE AS SELECT (SQL) without Unrecoverable Mode option	Current or new
5	CREATE TABLE AS SELECT (SQL) with Unrecoverable Mode option	Interim
6	CREATE TABLE AS SELECT (SQL) without Unrecoverable Mode option	Interim
7	CREATE TABLE and INSERT (SQL)	Current or new
8	CREATE TABLE and INSERT (SQL)	Interim
9	FastCopy	New
10	PL/SQL (used for SQL reorganizations of tables with LONGs)	Current or new
11	PL/SQL (used for SQL reorganizations of tables with LONGs)	Interim
12	FastCopy	Current

About Data Movement Methods

The data movement method determines how data is moved during a standard or live reorganization. Data is moved from original tables to reorganized copy tables using either SQL or FastCopy (or DBMS_DataPump).

i Note: For Oracle database 12c or later, Space Manager uses DBMS_DataPump instead of FastCopy.

In previous versions of Space Manager (8.0.2 or earlier), the Reorg Manager allowed you to specify a data movement method. However, Space Manager is capable of determining the most efficient data movement method to use for each object you want to reorganize. So, beginning with Space Manager 8.1, the Reorg Manager now automatically selects the most appropriate data movement method (SQL or FastCopy/DBMS_DataPump) and includes it in the reorg script. SQL is used for tables that do not contain LONG or LONG RAW columns. FastCopy/DBMS_DataPump is used for supported tables that contain LONG or LONG RAW columns.

This method of allowing Space Manager to automatically script the most appropriate data movement method is the most inclusive. It allows all tables with LONG columns to be included in a reorganization regardless of the size of these columns. Either SQL or FastCopy/DBMS_DataPump can be used for LONGs up to 32,760 bytes, and FastCopy/DBMS_DataPump supports LONGs that are up to 2 GB in size. This method also allows tables that are not supported for FastCopy (or DBMS_DataPump) to be included in the reorganization.

i Important: If a table is not supported for FastCopy or DBMS_DataPump and contains a LONG or LONG RAW column in which some rows contain data that is 32,760 bytes or larger, the table cannot be reorganized.

Using FsCopy or DataPump

Beginning with release 8.3, you can instruct Space Manager to use either FsCopy or DataPump as the data movement method, depending on your Oracle database version. FsCopy uses an import and export process to move data. It is one of the actions included in the Space Manager FastCopy data movement method. See [Select Global Reorganization Options](#).

i Important: This option is not recommended for most types of reorganizations. Care should be taken when selecting this method. FsCopy and DataPump are not always the most efficient data movement methods and should be used only for large tables where table copy might cause Oracle errors.

Run the Find Long LONGs procedure in order to populate the Find Long LONGs table.

Reorg Manager checks this table for the size of LONG and LONG RAW columns. The size of these columns determines which data movement methods can be used to reorganize the column's table. See [Appendix](#) on page 204 for more information.

Data Movement Method for Related Objects

Indexes and other related objects are automatically reorganized along with their tables. SQL is always the data movement method used for indexes, even when FastCopy (or DBMS_DataPump) is used for their tables.

More About Scripting Options (LiveReorg)

Several scripting options are unique to live reorganizations. All are set from the **Scripting Options** page of Reorg Manager. These options determine how the switch from original table to reorganized copy table begins—automatically, upon user approval, or in a certain time window. They also determine whether original tables are dropped or saved after they are replaced with reorganized copy tables.

Set Switch Option

The switch option (approval method) selected for a live reorganization script is used for all tables, table partitions, and table subpartitions in the script. When multiple tables, partitions, or subpartitions are included, they are reorganized serially, one at a time. There is one switch per object.

How soon a table is ready for the switch after reorganization begins depends on the table's size, the size and number of its indexes, and whether statistics are being collected. Recreating a large table and its indexes could take several hours or more.

i Tip: After a live reorganization script has been stored and scheduled, you can select a different switch option (approval method), if needed. This is done through the Script/Job Monitor. The change must be made before script execution begins. See [Change Approval Method for Online Scripts](#) on page 135 for more information.

Set the Automatic Switch Option

Select the **Automatically** option if you want the switch to be made automatically, as soon as a copy table is ready. When multiple tables are included in one reorganization script, the switch is made automatically for each table in turn.

The automatic switch option is a good choice when the online switch style will be used for all tables. By allowing the switch to occur automatically, you can avoid unnecessary delays that might occur if another switch option were used. You also avoid the need to be on hand to approve the switch.

Set the User Approval Switch Option

Select the **Upon user approval** option if you want the switch to be made after a user approves the switch.

Approval should be given as soon as possible after a table is ready for the switch. Until approval is given, QSA keeps the original table in sync with the copy table.

When multiple tables are included in a reorganization script, multiple approvals must be given. How much time passes before each approval is required depends on the size of each table being reorganized, the size and number of its indexes, and whether statistics are being collected.

Approval is given from the Script/Job Monitor. When any of the tables in a reorganization script are ready for the switch, "Waiting for approval" is the status of the script. See [Approve the Switch](#) on page 136 for more information.

The user approval switch option is a good choice when the T-Lock switch style will be used for one or more tables. It gives you a chance to stop activity against a T-Locked table prior to the switch. This allows you to avoid the message that displays when attempts are made to modify data in a T-Locked table. The purpose of the message ("QSA-20509: Live reorganization in progress") is to alert users that changes cannot be made at the current time. It does not indicate a problem.

Set the Time Window Switch Option

Select the **Between date/time** option if you want the switch to be made within a time window. Also specify a start date and time and end date and time for the window. The default start time is the current date and time. The default end time is one hour later. The switch is made automatically for all tables that are ready for the switch during the time window specified.

The start of the switch window should be coordinated with the start of script execution. Ideally, the switch window should be timed to start before the first table is ready for the switch. To estimate when the first table will be ready, consider its size, the size and number of its indexes, and whether statistics are being collected. Recreating a large table and its indexes could take several hours or more.

The duration of the switch window should be long enough to accommodate the switch for all tables in a script:

- If the window starts after the first table is ready for the switch, QSA continues posting live transactions to its reorganized copy table. Once the window starts, the switch is made for the first table. It is then made for all subsequent tables as soon as they are ready. The status for a script that is waiting for a time window is "Waiting until Day/Date/Time".
- If the window ends before the switch is made for all tables in a script, execution aborts with this message: "QSA-20502: Target completion date/time expired." This happens after 20 "retry operation" attempts are made.

The time-window option provides an alternative to the user-approval option. Where the user-approval option gives you a chance to stop activity against a table and then approve the switch, the time-window option allows the switch to proceed automatically at a time when activity is light.

i Tip: You can approve the switch instead of waiting for a time window to start. The status for a script that is waiting for a time window to start is “Waiting until Day/Date/Time”. See [Approve the Switch](#) for more information.

Making the switch when the database is quiet allows you to avoid the message that displays when attempts are made to modify data in a T-Locked table. The purpose of the message (“QSA-20509: Live reorganization in progress”) is to alert users that changes cannot be made at the current time. It does not indicate a problem.

Set the Drop Original Table Option

You can control whether original objects are dropped or saved after a reorganization using the **Drop original table after successful reorganization** checkbox.

This checkbox is selected by default so that original tables and indexes are dropped after they are replaced with their reorganized copies. In Oracle 10g databases, original objects are also purged from the Recycle Bin.

If you want to save original tables and indexes, clear the checkbox. Original objects are preserved in their original tablespaces under modified names. (When a table is reorganized with the T-Lock switch, unique indexes are kept and indexes that enforce a constraint are not saved.)

Dropping original objects frees up space. Saving original objects provides a backup of table data up to the point where the switch completes.

i Note: Saving original tables does not save objects created by QSA. For example, if a ROWID index is created for use as the posting index, the index is dropped after its table is reorganized.

Names Used for Saved Original Objects

The following chart displays the names used for original objects that are saved after a live reorganization. Each name includes the ID of an original table, partition, or subpartition.

Object type and switch style	Name of original object	Name of saved original object
Table with online switch	NAME	NAME_<object id>_ORIG NAME_<object id>_ORIG PARTITION (LR_DATA) is shown in the Explorer
Related indexes	NAME_IDX	QUEST_SPC_TMP_<object id>_LRIDX QUEST_SPC_TMP_<object id>_LRIDX PARTITION (LR_DATA) is shown in the Explorer
Table with T-Lock switch	NAME	NAME_<object id>_ORIG
Related indexes	NAME_IDX	NAME_IDX_<object id>_ORIG Only non-unique indexes and indexes that do not enforce a constraint are kept
Table partition or subpartition with either switch style	NAME	NAME_<object id>_ORIG
Related indexes	NAME_IDX	QUEST_SPC_TMP_<object

Object type and switch style	Name of original object	Name of saved original object
		id>NNN_LRIDX (used for posting index) QUEST_SPC_TMP_<object id>NNN_EXIDX (used for non-posting indexes)

Select a Partition Reorganization Method

When two or more partitions or subpartitions from the same table or index are included in a reorganization, the **Partition Reorg Method** dialog displays. This dialog allows you to select a partition reorganization method, i.e., whether partitions or subpartitions are reorganized individually or as part of an entire partitioned object.

The **Partition Reorg Method** dialog asks if you want to drop and recreate the entire partitioned object. This gives you the opportunity to reorganize the entire parent object which might be faster than reorganizing its partitions or subpartitions individually.

To select a partition reorganization method

When the **Partition Reorg Method** dialog displays, do one of the following.

- Click **Yes** to reorganize the entire parent object.
All partitions or subpartitions in an object are reorganized as a single object. If this method is selected for a table, all indexes on the table are also dropped and recreated. The parent and all related objects are retrieved and a red arrow displays beside the parent object in the Objects to be Reorganized list.
- Click **No** to reorganize the partitions or subpartitions individually. Green arrows display beside the partitions/subpartitions in the Objects to be Reorganized list.

Note that if there are multiple parent objects listed in the Partition Reorg Method dialog, the partition reorg method you select is applied to all. To select a different partition reorg method for one or more of the parent objects listed, reorganize them separately.

i Note: When selecting two or more partitions/subpartitions (from the same object) to reorganize in a LiveReorg, if you select the **Online switch only** mode, table partitions and subpartitions are reorganized individually.

Arrows Indicate Objects to be Reorganized

After you make your selection in the Partition Reorg Method dialog, arrows display in the Objects to Reorganize list to indicate how partitions are to be reorganized (individually or with the parent).

- Green arrows display beside partitions and subpartitions that will be reorganized individually.
- Red arrows display beside partitioned objects that will be reorganized as a whole.



Set Storage Strategies

After you select scripting options or a partition reorganization method, you can define a global storage strategy in the Storage Strategy window. Options in this window let you adjust your current strategy in order to accommodate future growth or use space more efficiently.

i Note: This window only displays if you have dictionary managed tablespaces. For objects in locally-managed tablespaces, tablespace extent sizes and storage values are used by default.

Defining extent allocation on a global basis saves you time when the same strategy can be used for all or most objects in a reorganization. If needed, you can override global extent values for individual objects in the Customize Object Allocations window. See [Customize Object Allocations](#) on page 75 for more information.

As you define a strategy, check the Estimated Extent Allocation Summary. This displays total current allocation for all objects in kilobytes and number of extents. It also displays estimates of what the new allocation will be after reorganization. Post-reorganization estimates in **Allocation** and **Extents** fields are recalculated each time you adjust a strategy.

Review the following for additional information:

Field	Description
Use Tablespace Defaults	<p>Select to use default tablespace extent values for all objects. The values used for each object are those of its target tablespace:</p> <ul style="list-style-type: none"> • When an object remains in its current tablespace, the default extent values for the current tablespace are used for that object. • When an object is relocated to a different tablespace, the default extent values for the new tablespace are used for that object.
Fixed Extent Size	<p>Select to use the same extent size for all objects. Select one of the following options:</p> <ul style="list-style-type: none"> • All Extents—Select this option if you want extents for all objects to be the same size. Then specify the size in megabytes or kilobytes. This may be rounded up to a multiple of database or tablespace block size. • Choose extent size—Select this option if you want extents for individual objects to be the same size and also want to limit number of extents. Also enter a number for the limit. Extents are sized so that the number initially allocated to each object does not exceed your limit. The same limit is used for all objects in a script. However, the size of extents can vary from object to object.
Size the segment (with appropriate value) using	<p>Select one of the following options to determine how much space is allocated to objects when they are recreated:</p> <ul style="list-style-type: none"> • Current Usage—Select this option if you want initial allocation to be based on space currently used. This option is useful for objects whose current allocation is too large. It resets their high water mark so that the amount of space allocated matches the amount currently used for data and eliminates wasted extents. • Current Allocation—Select this option if you want initial allocation to be based on space currently allocated. • Anticipated size on—Select this option if you want initial allocation to be sufficient for space needs on a future date. Space Manager calculates future space needs for each object based on its current growth rate. <p>If an object is being partitioned or subpartitioned, the allocation is divided among its partitions or subpartitions.</p>

Default Maximum Extent Size

The maximum extent size allowed by Space Manager is two gigabytes minus 16 kilobytes (2097136K). However, the default value for maximum extent size can be changed. For example, the default can be decreased if your system limits database file size. To change the maximum extent size

Insert a row for `maximum_extent_size` in the `QUEST_SPC_PARAMETER` table and specify your maximum in the **VALUE** column. If you are decreasing the 2G maximum, value can be expressed as K, M, or G.

About Statistics Collection Options

In the Reorg Manager, statistics collection options are specified on the Scripting Options page (Analysis section). These options allow you to automatically collect or restore statistics when a reorganization script is run.

See [The Space Manager Repository](#) for an overview of statistics collection in Space Manager.

The Reorg Manager provides the following options for statistics collection:

- **Collect statistics for the CBO (cost-based optimizer)**—This option lets you collect post-reorganization statistics.
- **Restore original statistics in data dictionary for the CBO**—This option lets you preserve existing statistics in the Oracle data dictionary instead of collecting statistics. It is designed for use when data-dictionary statistics are pre-set for Oracle's cost-based optimizer.

i Note: Restore functionality requires the `DBMS_STATS` package.

For information on how to specify statistics collection options in the Reorg Manager, see [Select Scripting Options](#).

How the Restore Option Works During Collection

When you restore statistics during statistics collection, data dictionary statistics are copied before objects are reorganized. The statistics are restored to the data dictionary afterward. The backup and restore are performed using procedures from the `DBMS_STATS` package. Before a reorganization begins, the `export_table_stats` procedure is used to export existing statistics from the data dictionary to a backup table. This is called `QUEST_SPC_REORGANIZATION_STATTAB`. After statistics are collected, statistics in the backup table are imported into the data dictionary with `import_table_stats`. They are then deleted from the backup table with `delete_table_stats`.

i Important: The `DBMS_STATS` procedures used by the restore option processes statistics needed by Oracle's cost-based optimizer. They do not process statistics such as number of chained rows, average free space in data blocks, number of empty blocks, and average space in freelist blocks. As a result, these statistics are not preserved by the restore option. If they exist in the data dictionary before collection is performed with this option, they will not exist there afterward.

Review Space Usage

After you specify storage options for individual objects in the **Customize Object Allocations** page, the Reorg Manager performs a space audit and displays the results in the **Space Usage Summary** tab. Reorg Manager checks for the following:

- Free space in target tablespaces—Reorg Manager checks target tablespaces to see if they have enough free space for original and copy objects.
- Data block size—Reorg Manager checks to see if tablespaces for partitions and subpartitions from the same object use the same data block size. If these tablespaces use different sizes, a warning is raised. Make note of the objects listed in the warning. Then return to the **Customize Object Allocations** page and select tablespaces that use the same block sizes for these objects.

- LiveReorg tablespace—If you are performing a live reorganization, Reorg Manager checks to see if the LiveReorg tablespace has enough space for LiveReorg objects. The LiveReorg tablespace and LiveReorg objects are used by QSA. The LiveReorg tablespace is specified with the agent's LW_TABLESPACE parameter. Space checks by Reorg Manager consider the space needed for creating LiveReorg objects. They do not consider additional space that might be needed for object growth. If a live reorganization fails because a LiveReorg object cannot grow, a collection trigger error is displayed in the script log.

i Note: Another space audit is performed when a script is run to make sure sufficient space is still available.

Space Usage Summary Tab

The **Space Usage Summary** tab displays information for segments and tablespaces involved in a reorganization. This includes original tablespaces, tablespaces being used on an interim basis, tablespaces to which objects are being relocated, and the LiveReorg tablespace.

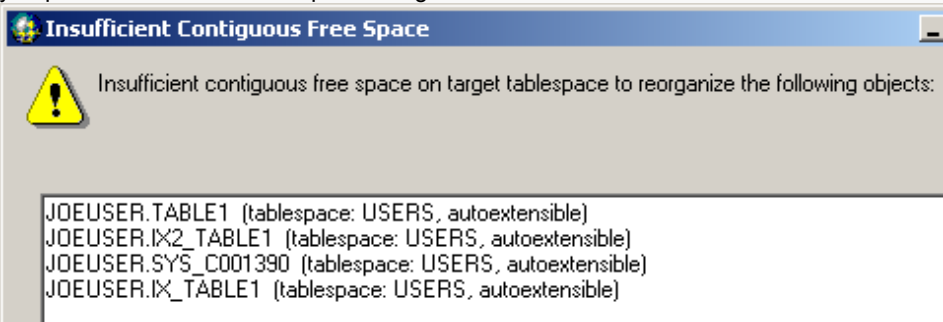
For each segment, the information displayed includes current and ending space allocation values.

For each tablespace, the information displayed includes starting free space and ending free space.

Space Usage Messages

Space Manager may alert you to space availability with various messages. If a message indicates that a target tablespace or the LiveReorg tablespace does not have enough free space, open Tablespace Properties for the tablespace and add datafiles or increase datafile size. Leave Reorg Manager open while you do this. You can also return to previous windows and select different target tablespaces or adjust storage values for the objects being reorganized.

If Space Manager displays a message after estimating space usage (in the Reorg Manager or other areas of the application), review the message description and suggested guidelines below.

Message	Description
Insufficient Free Space	When one or more target tablespaces do not have enough free space, the “insufficient free space” message is displayed at the bottom of the Review Space Usage window. Values for target tablespaces without enough free space are displayed in red. A negative value is displayed in the Lowest Usable Freespace column.
Insufficient Contiguous Free Space	<p>When one or more target tablespaces do not have enough contiguous free space for more objects, a message with a list of these tablespaces and objects displays when you proceed to the Review Space Usage window.</p>  <p>To provide the contiguous space needed in a target tablespace or the LiveReorg tablespace, use one of the methods under “Space Usage Messages”. If your actions do not result in sufficient contiguous free space, one of the following messages is displayed in the Review Space Usage window:</p> <ul style="list-style-type: none"> • Insufficient free space—This message is displayed when one or more target tablespaces does not have enough free space and enough contiguous free

Message	Description
	<p>space.</p> <ul style="list-style-type: none"> Insufficient CONTIGUOUS free space—This message is displayed when all target tablespaces have enough free space but one or more does not have enough contiguous free space.
Current Allocation Notes	<p>The Review Space Usage window sometimes displays a note indicating that some objects were sized based on current allocation instead of current usage. Current allocation and current usage are two sizing options for the Minimize Number of Extents global allocation strategy. You can select either one. However, the current usage option requires current statistics in the Space Manager repository. If statistics are not current for an object, current allocation is used instead of current usage.</p> <p>Notes: Some of the objects were sized based on current allocation because:</p> <ul style="list-style-type: none"> No statistics were found. Statistics are out of date. They are clusters. <p>The note gives one of the following reasons for use of current allocation:</p> <ul style="list-style-type: none"> No statistics were found—This reason is given when an object does not have statistics in the Space Manager repository. It is always given for LOBS as statistics are not collected for LOBs and are not available as a result. Statistics are out of date—This reason is given when an object does not have current statistics in the Space Manager repository. Statistics are considered to be out of date when an object’s allocation has changed since its last statistics collection. They are clusters—This reason is given for clusters. Statistics are not collected for these objects and are not available as a result.
Underestimates and Overestimates	<p>Space checks can sometimes underestimate or overestimate the space required by an object. This happens when statistics in the Space Manager repository are not current. The results of incorrect estimates are as follows:</p> <ul style="list-style-type: none"> Results of underestimates—When the space required by an object is underestimated, a “cannot allocate extent” error occurs during script execution. Results of overestimates—When the space required by an object is overestimated, an “insufficient free space” or “insufficient CONTIGUOUS free space” message is displayed in the Review Space Usage window. This can happen even if sufficient free space is available in the object’s target tablespace. <p>Even when space checks are accurate, “Freespace not available” errors can occur during script execution. This happens when extents are allocated in a different order than anticipated. For example, a space check might anticipate that larger extents will be allocated to larger objects first, but Oracle might allocate these extents to smaller objects first. As a result, they are not available for larger objects.</p>

Respond to Checks for a Posting Index

The wizard checks for indexes that can be used as posting indexes. A posting index is needed when live transactions are copied to a reorganized copy table. The index guarantees uniqueness of rows in the copy and speeds up the posting process. Each table in a script has its own posting index.

In most cases, Space Manager automatically chooses a posting index. Normally, the most selective index available for a table is selected. However, in some cases, Space Manager prompts you to make choices regarding the posting index. This happens when the only candidate for a table’s posting index has a PRIMARY KEY constraint or UNIQUE KEY constraint in the ENABLE NOVALIDATE state.

The message prompts you to validate constraints so that they can be used for posting. If you do not want to do this, you can exclude a table from a reorganization or select one of the two posting methods that are designed for tables that do not have selective indexes. These are the “all columns” posting method and the ROWID posting index. See [Posting Indexes](#) on page 214 for more information.

i Important: If you choose to validate a UNIQUE KEY constraint, posting may be slow if the leading column of the constraint is nullable.

To respond to checks for a posting index

1. From the Unsupported Objects message with two options, select one option:
 - **Exclude these objects from the reorganization**—Select this option to exclude tables whose only candidates for a posting index have PRIMARY KEY or UNIQUE KEY constraints that are ENABLE NOVALIDATE.
 - **Generate a script to validate these constraints**—Select this option to generate a script for validating constraints. The script is displayed in the SQL Editor. Whether you should run the script depends on the following:
 - If you want to validate PRIMARY KEY or UNIQUE KEY constraints for use in posting, run the script by pressing **F9** from the SQL Editor. Then close the SQL Editor and skip the rest of this procedure.
 - If you want to use the “all columns” posting method or ROWID posting index, close the SQL Editor. Then proceed to the next window in Reorg Manager. A second Unsupported Objects message displays with additional options.
2. From the Unsupported Objects message with four options, select one option:
 - **Exclude these objects from the reorganization**—Select this option to exclude tables whose only candidates for a posting index have PRIMARY KEY or UNIQUE KEY constraints that are ENABLE NOVALIDATE.
 - **Generate a script to validate these constraints**— Select this option to generate a script for validating constraints. The script is displayed in the SQL Editor. Run the script by pressing **F9**. Then close the SQL Editor.
 - **Use all columns of these tables for posting**—Select this option to use “all columns” posting for tables in the message. This method checks all columns in a table to guarantee row uniqueness in its reorganized copy. However, it does not check columns with LONG, LOB, XMLType, object type, REF, and VARRAY datatypes. By default, tables with those datatypes are excluded from a reorganization when this option is selected. If you want to include the tables and are sure that rows can be identified uniquely based on columns with other datatypes, set Space Manager’s ALLOW_LONG_WO_UNIQUE parameter to YES. See [All Columns Posting Method](#) on page 217 for more information.
 - **Use a temporary rowid column on the new tables**—Select this option to use ROWID indexes for tables in the message. These indexes check the ROWIDs of all rows in a table to guarantee row uniqueness in its reorganized copy. Unlike the “all columns” method, ROWID indexes guarantee row uniqueness for tables with LONG, LOB, XMLType, object type, REF, and VARRAY datatypes. However, they require DDL during script execution and do not support the online switch or FastCopy/DBMS_DataPump. Also, ROWID indexes are only used for partitions or subpartitions when a Space Manager parameter is set. If that parameter is not set, you are prompted to reorganize a partition’s parent table. See [ROWID Posting Indexes](#) on page 218 for more information.

Partition Tables with the Partitioning Wizard

Designed to streamline the partitioning process, the Partitioning Wizard lets you easily divide existing tables and their indexes into smaller units for better performance and easier maintenance. You can use it to:

- Convert nonpartitioned tables and indexes into partitioned tables and indexes. For example, you can partition a very large table across multiple tablespaces and storage devices for better I/O balancing.
- Convert partitioned tables and indexes into nonpartitioned tables and indexes. For example, you can unpartition a table that is too small to be partitioned and is not likely to grow in the future.
- Change the partitioning scheme for a table that is already partitioned. For example, you can add or remove partitions, change the partitioning method, or change the partitioning key.

After you define a partitioning scheme, the Partitioning Wizard automatically generates a script that contains all the Oracle commands needed to implement the scheme. The script applies your changes to a table and its indexes with a live reorganization. Because the reorganization takes place online, users and applications can continue transactions against a table as it is restructured. This allows you to perform critical database maintenance without interrupting access to data.

The Partitioning Wizard supports all of the partitioning methods available for Oracle. However, you cannot start with the parent of a reference-partitioned table. Additionally, you can start with system- or reference-partitioned tables, but you cannot produce them.

See [Launch Partitioning Wizard](#) to get started using the Partitioning Wizard.

Requirements for the Partitioning Wizard

Before you can use the Partitioning Wizard for a database, the following requirements must be met:

- The LiveReorg option must be licensed.
- The QSA Server Agent must be installed, current, and running for the database.
- Oracle's Partitioning Option must be installed and licensed for the database.

How Live Reorganizations Are Performed

Live reorganizations with the Partitioning Wizard automatically use a subset of the features available for live reorganizations with Reorg Manager:

- SQL is the data movement method. The SQL statements available are CREATE TABLE AS SELECT and CREATE TABLE and INSERT. FastCopy (or DBMS_DataPump) is not used.
- All objects are dropped and recreated.

- Entire tables and their indexes are reorganized. Partitions and subpartitions are not reorganized individually. Indexes cannot be reorganized without their tables.
- The switch is made at the parent table level. It is not made individually for partitions or subpartitions.
- The T-Lock switch is always used. The online switch is not used. If an object is not supported for the T-Lock switch, it is not included in a partitioning script.

Because the Partitioning Wizard automatically uses certain live reorganization features, it does not display windows that contain the LiveReorg option, switch styles, and partition reorganization methods.

The only live reorganization options you select are the following: the option that determines whether original objects are dropped and the option that determines how the switch proceeds from original table to reorganized copy table proceeds. It can begin automatically, upon user approval, or in a time window.

See [How Live Reorganizations Work](#) on page 209 for more information.

Smart Start for the Partitioning Wizard

This Smart Start provides an overview of all the major steps involved in using the Partitioning Wizard.

i Note: The Partitioning Wizard is only available when the LiveReorg option is licensed.

1. [Partitioning Wizard Defaults](#)

Specify default settings for partitioning options in the Options dialog. To open this dialog, select **Tools | Options | Partitioning Wizard**. You can override default settings from the Partitioning Wizard as needed.

2. Select objects and launch the Partitioning Wizard

i Note: See [Launch Partitioning Wizard](#) on page 98 for more information.

3. [Specify Partitioning Method](#)

Use the Partitioning Method page to specify partitioning method, number of partitions, and index locality:

- For tables—Specify partitioning method (range, list, or hash) and number of partitions.
- For partitions—To subpartition a range-partitioned table, specify a partitioning method for partitions (list or hash). Also specify the number of subpartitions each partition should contain. This can vary by partition.
- For indexes—Specify locality for each index associated with a table:
 - Local—Select this to partition an index in the same way as its table. Each index partition or subpartition corresponds to a certain a certain table partition or subpartition. The partitioning scheme for the table is automatically applied to the index.
 - Global—Select this to partition an index independently of its table. The index can be partitioned or nonpartitioned regardless of its table's structure. If both are partitioned, they can have different partitioning schemes.

4. [Specify Partitioning Key Columns](#)

Use the Partitioning Keys page to specify the columns to use in the partitioning keys that defines partitions and subpartitions.

5. [Specify Partitioning Parameters](#)

Use the Partitioning Parameters page to set partitioning options for individual objects. These options can be used to automate various aspects of partitioning. Initially, this window displays default settings from Partitioning Wizard options. You can override defaults as needed. The settings selected for a partitioned object apply to all of its partitions or subpartitions.

i Note: If you want to specify target tablespaces for individual partitions and subpartitions, you can do this in the Customize Object Allocation page.

6. [About the Partition Key Values Page](#)

Use the Partition Key Values page to specify the column values to use in partitioning keys. These define boundaries that determine how rows are assigned to partitions and subpartitions. If you chose to have the Partitioning Wizard set key values, the values are entered automatically. You can override them as needed.

7. [Specify Partition Names](#)

Use the Partition Names page to specify names for individual partitions and subpartitions, if needed. This window provides the final opportunity to specify names. The first is provided in the Partitioning Method page. Options in the first page let you specify names on a global basis or for parent objects. Options in Partition Names let you specify names for individual object.

8. **Select script options** in the Scripting Options page.

These options let you increase reorganization speed with Oracle's NOLOGGING mode and two types of parallel processing. See [Select Scripting Options](#) on page 72 for more information.

9. [Set Storage Strategies](#)

Define a global extent allocation strategy in the Storage Strategy window. This window displays when the SYSTEM tablespace is data dictionary managed. It allows you to size extents and objects on a global basis, for all objects in a script. You can override global values for individual objects in Physical Properties window.

10. **Customize Object Allocations.**

Adjust storage for individual objects in the Customize Object Allocations window. If needed, you can override extent values set on a global basis. You can also override global settings for target tablespace and sorted reorganization. See [Customize Object Allocations](#) on page 75 for more information.

11. Check the Space Usage Summary tab to see if target tablespaces and the LiveReorg tablespace have enough free space for a reorganization.

This tab identifies tablespaces that do not have enough free space overall or enough contiguous free space. To provide free space, select different target tablespaces or adjust storage values. To provide space in the LiveReorg tablespace, add datafiles or increase datafile size with Tablespace Properties. You can leave the Partitioning Wizard open as you do this.

12. Generate a reorganization script and schedule it in the Schedule Script Execution window.

Partitioning scripts must be run on a scheduled basis. This ensures that they are executed by QSA. Only QSA can process the LiveReorg commands they contain. See [Manage and Monitor QSA](#) on page 143 for more information.

! **CAUTION: Reorganization commands must not be modified or removed from a script. If they are changed in any way, a reorganization can fail or unpredictable results can occur. Most commands begin with QUEST_EXEC.**

13. [Using Job Monitor to Monitor Scripts](#)

Monitor script execution from the Script/Job Monitor and its Live Reorg Detail pane:

- Script/Job Monitor—This window displays execution status and executing statements.
- Live Reorg Detail—This pane displays activity statistics for live transactions during the first three stages of a live reorganization. It also identifies which stages are complete. You can use the statistics to estimate when a table will be ready for the switch.

i **Tip:** If the switch option is **Upon User Approval**, check the Script/Job Monitor to see when each table in a script is ready for approval. "Waiting for approval" is displayed in the **Status Message** column.

14. [Approve the Switch](#)

When a table is ready for the switch to be approved, right-click its script in the Script/Job Monitor and select **Approve Switch**. The switch from original table to reorganized copy table is made and script

execution continues. Monitor execution to see when the next table in the script is ready for approval.

Launch Partitioning Wizard

You can launch the Partitioning Wizard from multiple points in the interface. Select one or more objects to reorganize, then launch the Partitioning Wizard.

Review the following notes about selecting objects.

- When you select objects for the Reorg Manager or Partitioning Wizard, their associated objects are added to the object list. These objects include indexes, LOBs, and IOT overflows. IOTs (index organized tables) are replaced with IOT indexes.
- In most cases, related objects descend from their base objects in a hierarchical view.
- Partitions and subpartitions can be selected from all launch points. When a partition/subpartition is selected in the Partitioning Wizard, the entire partitioned object is added to the object list.

How to Launch the Partitioning Wizard

You can launch the Partitioning Wizard from the following areas in the Space Manager workspace. You can select a single object or multiple objects to partition simultaneously.

Step 1: Select objects to partition

1. In the Explorer window, select a tablespace or owner.
2. In the Segment or Object grid, select one or more objects to partition.
3. Alternatively, you can select one or more objects to partition from the Advanced Search window.

Step 2: Launch the Partitioning Wizard

1. Use the previous method to select one or more objects in the Explorer window.
2. Right-click and select **Partitioning Wizard**.
3. The Partitioning Wizard opens to the Partitioning Method page. See [Specify Partitioning Method](#) for next steps.

Objects Not Supported by the Partitioning Wizard

If objects are not supported for the Partitioning Wizard, they are excluded after you select them. The following messages identify which objects are excluded and why.

- Are not supported for partitioning: Objects that are not supported by Oracle for partitioning are excluded. These objects include tables with LONG columns.
- Cannot be reorganized with the T-Lock switch: Partitioning scripts use the T-Lock switch for all objects. Objects are excluded if they are not supported for the T-Lock switch.
- Do not contain columns that can be used in keys: Objects need at least one column that can be used in the partitioning key. Objects are excluded if none of their columns can be used. Columns that cannot be used in partitioning keys are those defined with the following data types:
 - LOB
 - LONG
 - RAW

- LONG RAW
- ROWID
- TIMESTAMP WITH TIME ZONE
- User-defined data types (including VARRAY, REF, OBJECT, and NESTED TABLE)

For a detailed list of objects not supported by the Partitioning Wizard, see [What is Not Supported for the Partitioning Wizard](#).

How the Wizard Streamlines Partitioning

One way the Partitioning wizard streamlines the partitioning process is with a graphical user interface that lets you easily define and see a partitioning scheme. Another is by automating tasks such as specifying partitioning key values, sizing partitions, and selecting tablespaces for I/O balancing.

Graphic Features in the Wizard

Wizard windows let you quickly select or enter partitioning method, number of partitions, index locality, partitioning key columns, and partitioning key values. Tabs beside these windows let you review a partitioning scheme as you define it.

The main windows for defining a partitioning scheme are as follows:

- **Partitioning Method**—This window lets you select partitioning and subpartitioning methods from lists. It also lets you enter number of partitions and subpartitions. You can specify a default number of subpartitions for a table or set a different number for each partition.
- **Partitioning Keys**—This window lets you specify partitioning key columns by selecting the columns from lists in grids. The lists include all columns in a table or index that are eligible to serve as partitioning columns. They exclude columns that are not supported by Oracle for partitioning keys.
- **Partition Values**—This window provides grids and lists for specifying partitioning key values. Both identify partitioning key columns and their datatypes. This allows you to see the types of values that can be entered as partition boundaries. The key values grid for range partitions is arranged in columns that let you easily compare the high values for boundaries. The key values lists for list partitions and subpartitions provide a function that lets you retrieve existing values in the partitioning key column. You can then select the values to use for a partition or subpartition.

i Note If an object is already partitioned, the Partitioning Wizard displays its current partitioning scheme. You can modify this as needed.

Automated Features in the Wizard

The Partitioning Wizard provides multiple options for automating the process of defining a partitioning scheme. You can specify all values yourself or use wizard options to automatically:

- Specify index locality as local or global.
- Set partitioning key values so that rows are evenly distributed among partitions or subpartitions (upper bounds for range partitions; list values for list partitions and subpartitions).
- Estimate partition sizes based on row count (which is calculated based on partitioning boundaries you specify).
- Validate partitioning key values against existing data to ensure there is a partition or subpartition for every row in a table.

- Specify sampling method and sample size for options that automatically set partitioning key values and estimate partition sizes.
- Set partition and subpartition names using naming patterns you specify.
- Assign partitions to tablespaces in order to balance I/O operations across multiple tablespaces.

As you define a partitioning scheme, the Partitioning Wizard ensures that settings conform to Oracle's rules. For example, when you enter partitioning key values, the wizard ensures that syntax is valid; that values do not exceed the maximum length, precision, or scale defined for a key column; that high values for range partitions increase in order; and that list partitions do not use duplicate values.

When you change a setting for an existing partitioning scheme, the Partitioning Wizard tries to preserve other settings. If the change to one setting causes another setting to become invalid, the wizard displays a message. You can adjust the invalid setting based on information in the message.

What is Not Supported for the Partitioning Wizard

The following Oracle features are not supported for the Partitioning Wizard.

Category	Oracle Feature Not Supported
Tables and Indexes	Indexes without their tables To reorganize an index online independently of its table, perform a standard reorg with Rebuild ONLINE
	Temporary tables
	Tables with dimensions
	Advanced Queuing tables
	Replication tables; objects that are part of a replication group
	External tables
	Domain indexes (including interMedia text indexes and ConText indexes)
	Tables owned by SYS or SYSTEM or with names starting with QUEST_%
IOTs and Clusters	IOTs and IOT overflows
	Clusters, clustered tables, cluster indexes, indexes on clustered tables
Materialized Views	Materialized views
	Materialized view logs
LONG Data Types	LONG and LONG RAW data types
User Defined Data Types	Object tables (data types for tables)
	Nested tables

Category	Oracle Feature Not Supported
Various Data Types	URI data types
	XMLType columns and tables
Miscellaneous	Oracle Label Security
	Tables with security policies enabled

i Notes:

- Destination tables cannot be system or reference partitioned.
- Columns with user defined datatypes (UDTs) cannot be used in new partitioning keys. They are preserved in existing partitioning keys as long as you do not change partitioning method or number of partitions. This applies to all supported UDTs. Also, index used as posting index must not contain an object type attribute in its column list.

Key Concepts for the Partitioning Wizard

This section describes key concepts for partitioning. Refer to this section as you use the Partitioning Wizard.

Partitioning Wizard Basics

Before you make changes to an object, select it in the Object list.

When an object is selected in the Object list, all options apply to that object. The last object selected in one window remains selected when you proceed to the next window.

Any settings specified for an object are applied when you select another object. You can change settings multiple times before generating a partitioning script. Changes do not go into effect in your database until the script is run.

Partition and Subpartition

A partition is a section of a table or index that has been divided into smaller parts that can be managed individually. Table and local index partitions can be divided into subpartitions.

To divide an object into partitions or subpartitions, you specify a partitioning scheme. This consists of partitioning method, number of partitions, number of subpartitions (optional), and a partitioning key. The key designates the columns and column values to use in defining the boundaries of partitions and subpartitions. Rows are assigned to a partition or subpartition if their values fall within the boundaries for that partition or subpartition.

Each partition is stored in its own segment and can have its own physical attributes such as TABLESPACE and PCTFREE. However, all partitions in a table or index share logical attributes such as column definitions and constraints.

When a table is subpartitioned, data is stored in the segments of the subpartitions. The partitions themselves are logical structures only. All subpartitions share the same logical attributes. They inherit most of the physical attributes of their parent partition or parent object. Only the TABLESPACE attribute can be specified for subpartitions. Locating partitions or subpartitions in different tablespaces on different storage devices promotes optimal I/O performance.

Range Partitioning

Range partitioning divides a table or index based on ranges of sequential values such as dates. Each partition contains one range of values. Range values increase in order from first partition to last partition. The first partition contains the lowest range of values. The second partition contains a higher range of values. The last partition contains the highest range of values.

The boundaries for ranges are defined with a partitioning key. This determines the highest column values (upper boundary) each partition can contain. `MAXVALUE` can be used for columns in the last partition. This setting allows a partition to include rows with values that are too high for other partitions. It ensures that there is a partition for every row in a table or index.

Range partitioning allows ordered and related sets of data to be grouped together. When a table is range partitioned, its partitions can be subpartitioned with the hash or the list partitioning methods.

To range partition a table or global index, you specify:

- Partitioning method—Range.
- Number of partitions—One or multiple partitions.
- Partitioning key columns—One to 16 table or index columns. These should:
 - Ensure that data is distributed evenly among partitions.
 - Define an ordered list. For example, to partition by date, the first column might be a year column, the second column might be a month, and the third column might be a day column.
- Partitioning key values—One high value per partitioning key column for each partition. High values for at least one column should be in increasing order from first partition to last partition. For example, if a partitioning key uses a month column, key values for that column should be higher for each successive partition.

Hash Partitioning

Hash partitioning divides an object based on a “round robin” distribution of rows. Oracle evenly distributes rows among partitions or subpartitions using a hash of values in partitioning key columns. You do not need to specify partitioning key values as you do with range and list partitioning.

Hash partitioning is useful when you do not know data values well enough to specify range boundaries or list values that would provide evenly-sized partitions.

Hash partitioning and subpartitioning is available for tables in databases that are Oracle 8 or later. Hash partitioning is also available for global indexes in databases that are Oracle10g or later.

To hash partition a table, you specify:

- Partitioning method—Hash.
- Number of partitions—One or multiple partitions. An even number helps to ensure that rows are distributed evenly among partitions.
- Partitioning key columns—One to 16 table or index columns.
- Partitioning key values—None.

List Partitioning

List partitioning divides an object based on lists of discrete values. Rows are assigned to a list partition or subpartition if their values are on its list. This type of partitioning allows unordered and unrelated sets of data to be grouped together. It gives you precise control over how rows are assigned to partitions.

List partitions cannot be subpartitioned. However, range partitions can have list subpartitions.

To list partition a table, you specify:

- Partitioning method—List.
- Number of partitions—One or multiple partitions.
- Partitioning key columns—One table column.
- Partitioning key values—A list of discrete values for each partition or DEFAULT.

i Note One list partition can use DEFAULT as its value instead of a list of values. The DEFAULT partition is used for rows that do not contain values that match lists for other partitions.

Composite Range-Hash Partitioning

Range-hash partitioning combines range partitions and hash subpartitions in a two-level hierarchy. First, a table is divided into range partitions based on ranges of values. Then, each range partition is divided into hash subpartitions based on a hash of values in subpartitioning key columns.

When a table is range-hash partitioned, data is stored in the segments of the subpartitions. The partitions themselves are logical structures. All subpartitions share the same attributes. They inherit most from their parent partition or parent table. Only the TABLESPACE attribute can be specified for subpartitions.

To use composite range-hash partitioning, you specify:

For the table	<ul style="list-style-type: none">• Partitioning method for table—Range.• Number of partitions—One or multiple partitions.• Partitioning key columns—One to 16 table columns. These should:<ul style="list-style-type: none">• Ensure that data is distributed evenly among partitions.• Define an ordered list. For example, to partition by date, the first column might be a year column, the second column might be a month, and the third column might be a day column.• Partitioning key values—One high value per partitioning key column for each partition. High values for at least one column should be in increasing order from first to last partition. For example, if a partitioning key uses a month column, key values for that column should be higher for each successive partition.
For partitions	<ul style="list-style-type: none">• Partitioning method for partitions—Hash.• Number of subpartitions—One or multiple subpartitions. The number can vary by partition.• Partitioning key columns—One to 16 table columns.• Partitioning key values—None.

Composite Range-List Partitioning

Range-list partitioning combines range partitions and list subpartitions in a two-level hierarchy. First, a table is divided into partitions based on ranges of values. Then, each range partition is divided into list subpartitions based on lists of discrete values. There is one list per subpartition. You might use range partitions to group data by sales quarter and list subpartitions to group it by certain states for each quarter.

When a table is range-list partitioned, data is stored in the segments of the subpartitions. The partitions themselves are logical structures. All subpartitions share the same attributes. They inherit most from their parent partition or parent table. Only the TABLESPACE attribute can be specified for subpartitions.

To use composite range-list partitioning, you specify:

For the table	<ul style="list-style-type: none"> • Partitioning method for table—Range. • Number of partitions—One or multiple partitions. • Partitioning key columns—One to 16 table columns. These should: <ul style="list-style-type: none"> • Ensure that data is distributed evenly among partitions. • Define an ordered list. For example, to partition by date, the first column might be a year column, the second column might be a month, and the third column might be a day column. • Partitioning key values—One high value per partitioning key column for each partition. High values for at least one column should be in increasing order from first to last partition. For example, if a partitioning key uses a month columns, key values for that column should be higher for each successive partition.
For partitions	<ul style="list-style-type: none"> • Partitioning method for partitions —List. • Number of subpartitions—One or multiple subpartitions. The number can vary by partition. • Partitioning key columns—One table column. • Partitioning key values—A list of discrete values for each partition or DEFAULT. (DEFAULT can be used for one partition.)

Partitioning Key

Partitioning keys determine which partition or subpartition rows are assigned to. Each partitioning key consists of partitioning key columns and column values. These are used as follows:

- Partitioning key columns tell Oracle which table or index columns to match rows against. The partitioning key for range- and hash-partitioned tables uses up to 16 columns. The partitioning key for list-partitioned tables uses one column. The partitioning key for ranged-partitioned tables uses an ordered list.
- Partitioning key values are the column values that define the boundaries for partitions and subpartitions:
 - Range partitioning keys use high values as boundaries. A row is assigned to a range partition if its value is lower than the boundary for that partition. For each partition, you specify one high value per partitioning key column. The high values for at least one column should be in increasing order from first to the last partition.
 - List partitioning keys use lists of discrete values from a single key column as boundaries. A row is assigned to a list partition or subpartition if it contains a value on the list for that partition or subpartition. Lists should not contain duplicate values.
 - Hash partitioning keys do not require user-specified values. Instead, Oracle uses a hash function to determine values for key columns.

As you define a partitioning scheme, you specify one set of partitioning key columns per parent object. If a table is being subpartitioned, you also specify a set of subpartitioning key columns. You then specify partitioning key values for each partition or subpartition.

Local Indexes

A local index is partitioned in the same way as its table. It has the same number of partitions and subpartitions and uses the same partitioning key. Each index partition or subpartition corresponds to a certain table partition or subpartition.

Local indexes are automatically maintained along with their tables. When you specify Local as the locality for an index, the Partitioning Wizard applies the table's partitioning scheme to the index. When the table's partitioning scheme is changed, the index is automatically changed in the same way. For example, if you add a partition to the table, a corresponding partition is added to the index.

You cannot modify partitioning for a local index independently of its table. You can, however, modify the names of its partitions and subpartitions. You can also modify physical attributes for the index.

- i Note** When a local index is unique, the columns used for its table's partitioning key must be one or more of the columns that define the index. If a table has multiple unique indexes, the columns used for the partitioning key must be common to all of the indexes.

Global Indexes

A global index is partitioned independently of its table. The index can be partitioned or nonpartitioned no matter how its table is structured. If both are partitioned, the partitioning scheme for the index can be different from that of the table. Index partitions do not correspond to certain table partitions.

For global indexes in databases that are earlier than Oracle 10g, the partitioning method can only be range. For global indexes in databases that are 10g or later, the partitioning method can be range or hash. Columns in the partitioning key must be ordered in the same ways as columns in the index. If range partitioning is used, the high value for the last partition must be MAXVALUE. The Partitioning Wizard enters this value automatically and prevents use of another value.

- i Note** Global indexes can only be partitioned if their first column can be used in a partitioning key. Columns in the partitioning key must be in the same order as columns in the index.

View Partitioning Schemes

Tabs in the Partitioning Wizard are designed to let you easily review partitioning schemes as you define them:

- **Objects to be Reorganized**—This tab identifies which parts of a partitioning scheme have been defined for each object. If an object is already partitioned, only changes are reflected in the Objects and Partitions tab. For example, if you change partitioning key values for a range partitioned object, a Y for “yes” is displayed in the Values column. But the Actions and Keys columns are blank.

- i Note:** The Objects to be Reorganized tab is displayed in all windows where options can be selected for individual objects. This tab is not displayed in windows where options are used globally for all objects on the list.

- **Space Usage Summary**—This tab allows you to check space availability in target tablespaces.

A summary displays on the Schedule Script Execution page to allow you to review the reorg method, a list of objects included in the reorganization, and the switch types to be used.

Specify Partitioning Method

Use the Partitioning Method page of the Partitioning Wizard to specify partitioning method, index locality, and number of partitions and subpartitions. You can specify a default number of subpartitions for an object or specify

a number for each partition. Please note that LOBs are automatically partitioned in the same way as their tables.

For an overview of the Partitioning Wizard, see [Smart Start for the Partitioning Wizard](#).

If an object is already partitioned, the Partitioning Method page displays its current partitioning method, number of partitions, index locality, and segment names. You can change partitioning method, unpartition an object, convert to composite partitioning, add or remove partitions or subpartitions, change index locality, and change names.

After you change method settings, the Partitioning Wizard tries to preserve partitioning key settings, if possible. The wizard displays a message when new settings are invalid or cause other settings to become invalid. You can adjust invalid settings based on information in the message.

i **Tip:** You can save Partitioning Wizard settings as a Reorg Plan to edit and reuse later. See [Save Reorg Manager Settings](#) on page 81 for more information.

To specify the partitioning method

1. Launch the Partitioning Wizard. See [Launch Partitioning Wizard](#) on page 98 for more information.
2. Select a table in the Objects to Reorganize list.
3. Select a partitioning method for the table from the **Partition by** field.
4. Enter number of partitions in the **Number of partitions** field.

A table can contain one partition or multiple partitions. When hash is the partitioning method, enter an even number to ensure that rows are distributed evenly. If you do not want to add subpartitions to the table, click **Apply**.

i **Tip:** To convert a partitioned object to a nonpartitioned object, enter 0 as the partition count.

5. Select **Create subpartitions** to subpartition the table. From the **Subpartition by** field, select a partitioning method for all partitions.
6. If you want all partitions in a table to contain the same number of subpartitions, specify a default number in the **Default number of subpartitions** field. Then click **Apply**. Skip this step and proceed to the next if you want to customize the number of subpartitions per partition.
7. If you want to vary the number of subpartitions per partition, select the **Partition-level subpartitioning** checkbox. Specify the number of subpartitions each partition should contain in the **# Subpartitions** column. Then click **Apply**.

i **Note:** If most partitions should contain the default number of subpartitions, click **Reset subpartitions to default** to populate the # Subpartitions column with the default number. Then modify the number of subpartitions for individual partitions.

8. If a table has indexes, specify locality for each index. When locality is global, you can also specify the number of partitions and partitioning method.

Select an index in the list and then select one of the following options:

- **Local**—Select Local to partition an index in the same way as its table, with the same number of partitions and subpartitions, and the same partitioning key columns and values. If Local is the default setting, partitions and subpartitions for all indexes on a table are added to the Object list when you click **Apply** for the table.
- **Global**—Select Global to partition an index independently of its table. Then enter a number in the Number of partitions field. (Enter 0 if an index should be non-partitioned.) If your database is Oracle 10g or later, also select a partitioning method from the Partition by field. For pre-10g databases, range partitioning is used. Click **Apply** to apply all settings.

i Note: If you change partitioning for an index but do not change partitioning for its table, a message asks if you want to reorganize the table along with the index. Click **Yes** to include the table in the partitioning script. If you click **No**, you must make changes to the table in a later partitioning-related window. If no changes are made, the table and its indexes are excluded from the script.

9. If you want to change names for all new and existing partitions and subpartitions in an object, click **Change Default Names**. (You can change names for individual segments in the Partition Names window.) The Default Partition Names dialog displays. Enter names as follows:
 - Prefix for default partition names—Enter a name for partitions. (P is the wizard's default.) This is used for all partitions in a table and its local indexes, including existing partitions. It is used for all partitions in a global index. A sequential partition number is appended to the name of each partition to guarantee that it is unique.
 - Prefix for default subpartition names—Enter a name for subpartitions. (SP is the wizard's default.) This is used for all subpartitions in a table and its local indexes, including existing subpartitions. A sequential subpartition number is added at the end of each name; the parent partition name is added at the beginning.
 - Include base object name in default partition names—Select this checkbox to include the name of the parent table or parent index in the names of new partitions and subpartitions. The parent name is added at the beginning of each partition or subpartition name.

Click **OK** to close the Default Partition Names dialog. Then click **Apply** to apply new names.

i Note: Names are stored in uppercase characters if you do not enter them with double quotes. Each name can have up to 30 characters. That maximum is shared by all parts of a name, including numbers and parent names. If all parts add up to more than 30 characters, certain parts are truncated. However, sequential numbers are always included.

10. Click **Next** to proceed to the next page. See [Specify Partitioning Key Columns](#) for next steps.

i Tip: You can specify default values for some partitioning options in **Tools | Options | Partitioning Wizard | Defaults**. See [Partitioning Wizard Defaults](#).

Specify Partitioning Key Columns

Use the Partitioning Keys page of the Partitioning Wizard to specify the columns to use for partitioning keys.

If an object is already partitioned, the grid displays its current partitioning key columns. You can select different columns as needed. You must select different columns when you change partitioning method for an object. In this case, partitioning key grids are blank. You may need to select different columns when you change locality from global to local for a unique index. This must be done if the table's current key does not use one or more of the columns that define the index.

Lists in partitioning key grids include all columns in a table or index that are eligible to serve as partitioning columns. If a column is not supported by Oracle for use in partitioning keys, the column is not displayed.

i Tip: Columns are used in the order displayed in a partitioning key grid. In some cases, the columns in a key must be in the order used for columns in an index. To see how column order is defined for an index, open it in Object Properties and view the Columns tab.

To specify partitioning key columns

1. Select a table in the Object list. If you select a partition, the key columns for that partition are used as a key column for its table.
2. Select the first column to use for the table's partitioning key.

- If the table is list partitioned or you want to use only one column in the key, then proceed to Step 6.
 - If the table is range or hash partitioned and you want to add more columns, then proceed to Step 4.
- i Note:** If a table is range partitioned, columns should be in an ordered list. If a table has unique local index, columns in the table key must be part of the index definition. If a table has multiple unique indexes, columns in the key must be common to all the indexes.
3. (Range partitioning only) If the table is range partitioned and you want to use an interval, enter or select an interval value in the **Interval** field. This will enable interval partitioning. Oracle will automatically select the range based on the interval and range value you specify.
 - a. If you enter a value, the format must be consistent with the partitioning column's data type.
 4. Select additional columns for the table's partitioning key, if necessary. Select a column from the drop-down list immediately below the last column selected. Repeat this step for each column you want to add to the key (up to 16 columns for range and hash partitioned tables).
 5. If a table is subpartitioned, select columns for the table's subpartitioning key.
 6. Select partitioning key columns for global indexes

If a table has partitioned global indexes, the first one is selected after you apply key columns to the table. (Nonpartitioned global indexes and local indexes are skipped.)

i Note: Columns in the key for a global index must be in the same order as columns that define the index.
 7. Repeat Steps 2-6 for additional tables and global partitioned indexes on the Object list.
 8. If you want to make changes after key columns have been applied, select an object in the Object list and make one or more changes.
 - To add a column, click below the last column selected.
 - To remove a column, select the column and press the **DELETE** key.
 9. Click **Next** to proceed to the next page. See [Specify Partitioning Parameters](#) for next steps.

Specify Partitioning Parameters

Use the Partitioning Parameters page to set partitioning options for individual objects. These options let you automate various aspects of partitioning. For example, they let you automatically set partitioning key values and select target tablespaces.

Partitioning options are organized in three panels. A panel is displayed for an object if the options it contains can be used for that object.

Initially, most options in this page use default settings from the Partitioning Wizard pages in the Options dialog (**Tools | Options**). You can override defaults as needed. The settings selected for a partitioned object apply to all of its partitions and subpartitions.

To set partitioning options for an object

1. Select a table or global index in the Object list.
2. **Key Values and Sizing.** Select the **Automatically set upper bounds / list values...** option if you want the Partitioning Wizard to automatically set partitioning key values so that rows are distributed equally and partitions are sized evenly.

How the "automatically set" option is labeled depends on the partitioning method for an object:

- **Automatically set upper bounds for partitions to evenly distribute rows among partitions** is displayed when range is the partitioning method. In this case, the option tells the wizard to automatically set the high values to use as the upper boundary for range partitions.
- **Automatically set list values for list partitions to evenly distribute rows among partitions** is displayed when list is the partitioning method. In this case, the option tells the wizard to automatically specify the list values to use for list partitions and subpartitions.

i Note: Partitioning key values can be set automatically for objects that contain at least one row per partition or subpartition.

3. Select the **Estimate partition sizes based on row count** option if you want the Partitioning Wizard to automatically calculate partition sizes based on expected row count.

i Note: The estimate sizes option should not be used when the option for automatically setting partitioning key values is selected. When both are selected, space allocation is estimated twice.

4. Select the **Validate partitioning key values against existing data** option if you want the Partitioning Wizard to automatically check partitioning key values against existing data.

This ensures that a table and its local indexes contain partitions or subpartitions for all rows when MAXVALUE or DEFAULT is not used as a partitioning value.

To validate partitioning key values, the Partitioning Wizard checks them against all rows in a table. (This may take a long time for large tables.) When the wizard finds rows that cannot be assigned to any partition or subpartition, a message displays in the Partition Values window. Use information in the message to adjust partitioning key values.

i Note: The validate option is ignored for tables when MAXVALUE or DEFAULT is used as a partitioning value.

5. **Table Scan Options.** Select a table scan option to use for the **Key Values and Sizing** method/methods you specified.

- **Scan entire table if less than N KB**—Select this option to sample a table or determine whether to sample or scan based on its size. Also specify a threshold size and sample size.
 - **Threshold size**—Enter a threshold size in kilobytes in the field beside Scan entire table if less than N KB. The table is sampled if its size is equal to or larger than this. The entire table is scanned if its size is smaller than this. To sample a table regardless of size, enter 0.
 - **Sample size**—Enter the percentage of table rows to sample in the **Otherwise sample table at** field. The percentage can have two decimal points, for example, 50.25%.
- **Scan entire table**—Select this option to scan the entire table, regardless of size.

Table scan options determine whether tables are scanned or sampled when the Partitioning Wizard sets key values or estimates sizes. You can choose to scan entire tables, sample a percentage of table rows, or dynamically determine whether to scan or sample based on table size.

A full table scan may take a long time, depending on table size. Sampling a table takes less time and may be the best choice for very large tables. Although all rows are not read, they are evenly sampled so that results accurately represent a table.

6. Use tablespace options to set a global target-tablespace strategy for a partitioned table and its local indexes or for a partitioned global index. You can override global tablespaces in the Customize Object Allocations page.

a. Select a target tablespace option as follows:

- **Use the parent object's default tablespace for (sub)partitions**—Select this option to use a parent object's default tablespace for all its partitions or subpartitions, new and existing. If you want existing partitions or subpartitions to remain in their current tablespace, select that tablespace in the Customize Object Allocations page.

- **Use the tablespaces used by current (sub)partitions in a round-robin distribution**—Select this option to use current tablespaces for an object's partitions or subpartitions. If an object is being partitioned, its current tablespace is used for all of its partitions and subpartitions. If an object is already partitioned, the current tablespaces of its existing partitions or subpartitions are used. Segments are distributed among the tablespaces in an alternating fashion. Each tablespace is used in turn for the next partition or subpartition in a script.
- **Use the following tablespaces for (sub)partitions in a round-robin distribution**—Select this option to specify the tablespaces to use for an object's partitions and subpartitions. You can select multiple tablespaces from the Tablespace Name grid. The tablespaces you specify are used for both new and existing partitions and subpartitions. Segments are distributed among the tablespaces in an alternating fashion. Each tablespace is used in turn for the next partition or subpartition in a script. Tablespaces are used in the order displayed in the grid. To change tablespace order, drag and drop tablespace rows to new positions in the grid.

i Tip: To create a tablespace, type a name for it in the grid, press **Enter**, and click **Yes**. When Tablespace Properties displays, specify properties and click **Generate Script**. A script to create the tablespace runs automatically.

- Set the option for local indexes if you want to locate each local index partition or subpartition in the tablespace used for its associated table partition or subpartition. The option is labeled **Put local index (sub)partitions in the tablespaces of their table (sub)partitions**. Local index segments are kept with their table segments regardless of the target tablespace option selected. When the local index option is not selected, local index segments are assigned to tablespaces based on the target tablespace option selected. (This option is not displayed when a global index is selected in the Object list.)

7. Click **Next** to proceed to the next page. See [About the Partition Key Values Page](#) for next steps.

i Tip: You can specify default values for some Partitioning Wizard options in **Tools | Options | Partitioning Wizard**. See the [Partitioning Estimates](#) and [Partitioning Key Options](#) pages.

Best Practices for the Partitioning Wizard

If you select different target tablespaces in the Partitioning Wizard, be aware of how the new tablespace fits in with the strategy selected in the Partitioning Parameters page:

- Strategy for locating local index segments with table segments—This strategy automatically locates local index partitions or subpartitions in the tablespace for their table partitions or subpartitions. If you select a different tablespace for an index segment, it will not be located with its table segment.
- Strategy for using a parent object's default tablespace—This strategy automatically uses the default tablespace of a parent table or index for all of its partitions or subpartitions, both new and existing. If you want existing partitions or subpartitions to remain in their current tablespace, select that tablespace in the Customize Object Allocations page.
- Strategy for using current tablespaces—This strategy automatically uses current tablespaces for an object's partitions or subpartitions. These are the tablespaces where the parent object or existing partitions or subpartitions are currently located. Segments are distributed among the tablespaces in an alternating fashion. Each tablespace is used in turn for the next partition or subpartition in a script. This strategy is designed to distribute segments for optimal I/O balancing. If you select a different tablespace for a segment, it should be one that will promote I/O balancing.
- Strategy for using selected tablespaces—This strategy automatically uses one or more tablespaces you specify for an object's partitions and subpartitions. Segments are distributed among the tablespaces in an alternating fashion. Each tablespace is used in turn for the next partition or subpartition in a script. This strategy is designed to distribute segments for optimal I/O balancing. If you select a different tablespace for a segment, it should be one that will promote I/O balancing.

Specify Partitioning Key Values

The Partition Key Values page of the Partitioning Wizard is used to specify the key-column values used for determining each partition.

About the Partition Key Values Page

The page title, **Range Partition Key Values** or **List Partition Key Values**, is determined by the partitioning method you selected.

If you selected the **Automatically set...** option in the Partitioning Parameters page, the window displays values set by the Partitioning Wizard. If you did not select this option, use this page to manually specify partitioning key values.

Values grids and lists identify partitioning key columns and their datatypes. This lets you see the types of values that can be specified as partitioning boundaries. If you chose to automatically set partitioning values, the Partitioning Wizard's values are displayed in the grid or list for an object.

If an object is already partitioned, you can change partitioning key values as needed. You must readjust key values if you modify the specified key columns or the number of partitions.

Before specifying key-column values, you may want to review [Best Practices for Partitioning Key Values](#).

About Partitioning Key Values

The key-column values define the boundaries that determine how rows are assigned to partitions and subpartitions.

- The boundary for a range partition determines the highest partitioning key column value its rows can contain. The boundary is not inclusive. The partition will include all values less than (but not equal to) the value you specify.
- The boundary for a list partition or subpartition consists of the list of values its rows must contain.

Manually Specify Key Values

If you selected the **Validate partitioning key values** option, the Partitioning Wizard checks values against all existing data in a table to make sure there is a partition or subpartition for every row.

Based on message information, you can adjust values so that all data fit in a partitioned object. This lets you avoid the error that occurs when rows cannot be assigned:

“ORA-14400: Inserted partition key does not map to any partition”.

i Note If you selected the **Estimate partition sizes based on row count** option, be aware that key values you specify may change the partition sizes automatically estimated by the Partitioning Wizard.

To manually specify Range partitioning key values

1. Select a range-partitioned table or global index in the Object list.
2. For the first range partition, specify a high value to use to define the partition boundary.
 - a. Click in the field for the first partitioning key column and enter a value for that column. The partition will include values less than (but not equal to) the value you specify.

- b. If the partitioning key uses more than one column, press **Tab** to move the cursor to the second column in the key. Enter a boundary value for the second column. Repeat for each column in the key.

See [Key Value Grids for Range-Partitioned Objects](#) on page 113 for more information about object and partition grids.

3. Specify boundary values for the remaining range partitions.

MAXVALUE is automatically entered in the last partition and indicates that the highest value in the column is used as the boundary value for that partition.

4. Use the key values grid for the first partition to check high values. Modify values, if needed. Repeat this step for each partition in a range-partitioned object.
5. After specifying all partition key values, click **Next** to proceed to the next page. See [Specify Partition Names](#) for next steps.

To manually specify List partitioning key values

1. Select a list partition or subpartition in the Object list.
2. Specify a list of values for the partition or subpartition
 - a. Click **Get existing values** to populate the Existing Values list with existing column values. For a partition, all values in the column are retrieved. For a subpartition, column values for rows that belong to the parent partition are retrieved.
 - b. Select the values to include in the partition or subpartition. To do this, use the **UP** and **DOWN** arrows to move values to the Partitioning Key Values list for that partition.
 - c. Adjust the list, if needed. To delete a value, select it and press the DELETE key. To refresh the Existing Values list, click **Get existing values**.

See [Key Value Lists for List Partitioned Objects](#) on page 114 for more information about object and partition lists.

3. Repeat for each partition or subpartition in a list-partitioned object.

DEFAULT is automatically entered as the value for the last partition. This indicates that any values not entered in the previous partitions are included in the last partition.

4. After specifying all partition key values, click **Next** to proceed to the next page. See [Specify Partition Names](#) for next steps.

i Note: If tables have constraints that can be validated for use in posting live transactions, a message prompts you to validate the constraints when you attempt to proceed from the Partition Key Values window. See [Respond to Checks for a Posting Index](#) on page 114 for more information.

Best Practices for Partitioning Key Values

Before specifying partitioning key values, you might want to review these best practices.

- For all partitioned objects, specify key values so that rows are distributed evenly among partitions or subpartitions. Performance is best when rows are equally distributed.
- For range-partitioned tables, specify high values in increasing order from first partition to last partition for at least one partitioning key column. If values do not increase for at least one column, the wizard alerts you that range partition boundaries are inconsistent and identifies the objects in question. Use MAXVALUE for the last partition to ensure that there is a partition for rows with values too high for other partitions.
- For range partitioned global indexes, MAXVALUE must be used as the high value for all key columns for the last partition. The wizard automatically enters this value and prevents use of another value.

- For list partitioned tables and partitions, specify values so that the list for each partition or subpartition is unique. When you try to assign values that are used for other segments, the wizard identifies duplicate values and the other segments. Use DEFAULT for the last partition or subpartition to ensure that there is a segment for rows that do not match the value lists for other segments.
- For range partitions, specify values that exist in a table. Enter them so that they match table values exactly. Also, use the correct syntax for column datatypes and observe any limits, such as maximum length.
- For numeric values, do not exceed the precision or scale defined for partitioning key columns that use the NUMBER datatype. Precision is the maximum number of digits allowed in numeric values. Scale is the number of digits to the right of the decimal point allowed for numeric values.
- For date values, use the following format and do not use quotes:

YYYY-MM-DD HH24:MI:SS

Y=Year; M=Month; D=Day; H=Hour; MI=Minute; S=Second

Key Value Grids for Range-Partitioned Objects

There are two key value grids for range-partitioned objects. One is for all partitions in an object. The other grid is for individual partitions. You can use either grid to enter or adjust partitioning key values. Both are designed to let you see partitioning key values in relation to each other. This helps you avoid entering high values that fall in the range for another partition.

i Tip: As you specify partitioning values, you can see lists of the values contained in key columns by running SELECT statements from Space Manager's SQL Editor.

Key Value Grids for All Partitions

The grid for all partitions is displayed when a parent partitioned object is selected in the Object list. It allows you to enter boundary values for all partitions in the object.

All columns in the partitioning key are displayed on the right side of the grid. This lets you see what types of values can be entered in high-value fields. For each partition, there is one high-value field per key column.

Partition Name	Key Columns and Datatypes - Enter key values		
	SALES_YEAR	SALES_MONTH	SALES_DAY
	INTEGER	INTEGER	INTEGER
SALES_Q1	2005	04	01
SALES_Q2	2005	07	01
SALES_Q3	2005	10	01
SALES_Q4	2006	01	01

The advantage of this grid is that it lets you view all partitions and their partitioning key values in relation to each other.

Key Value Grids for Individual Partitions

The grid for individual partitions is displayed when a partition is selected in the Object list. It allows you to enter boundary values for just that partition.

All columns in the partitioning key are listed on the left side of the grid. These are presented vertically in the order they were specified. Their datatypes are listed beside them. This lets you see what types of values can be entered in high-value fields. There is one high-value field for each key column.

The advantage of this grid is that it allows you to focus on a certain partition. Because partitioning key columns are presented in a vertical list, the grid for individual partitions makes it possible to see all columns in a partitioning key without scrolling.

Key Value Lists for List Partitioned Objects

Fields for list values are displayed when a list partition or subpartition is selected in the Object list. These consist of two list fields and a **Get existing values** button. All provide an easy way to specify list of partitioning key values.

Click **Get existing values** to retrieve a list of values in the partitioning key column. These are displayed in the **Existing Values in Table** list field:

- If an object is a partition, all values in the column are retrieved.
- If an object is subpartition, column values for rows that belong to the parent partition are retrieved.

After values have been retrieved for one partition or subpartition, they remain displayed in the existing values field for all of its associated partitions or subpartitions. The field is cleared when you select another table.

To select values for a partition or subpartition, select one or more values in the Existing Values list and click the **DOWN** arrow to move the value to the Partitioning Key Values list.

To delete a value from the list, select it and press the **DELETE** key.

To refresh the Existing Values list, click **Get existing values**.

Respond to Checks for a Posting Index

The wizard checks for indexes that can be used as posting indexes. A posting index is needed when live transactions are copied to a reorganized copy table. The index guarantees uniqueness of rows in the copy and speeds up the posting process. Each table in a script has its own posting index.

In most cases, Space Manager automatically chooses a posting index. Normally, the most selective index available for a table is selected. However, in some cases, Space Manager prompts you to make choices regarding the posting index. This happens when the only candidate for a table's posting index has a PRIMARY KEY constraint or UNIQUE KEY constraint in the ENABLE NOVALIDATE state.

The message prompts you to validate constraints so that they can be used for posting. If you do not want to do this, you can exclude a table from a reorganization or select one of the two posting methods that are designed for tables that do not have selective indexes. These are the "all columns" posting method and the ROWID posting index. See [Posting Indexes](#) on page 214 for more information.

i Important: If you choose to validate a UNIQUE KEY constraint, posting may be slow if the leading column of the constraint is nullable.

To respond to checks for a posting index

1. From the Unsupported Objects message with two options, select one option:
 - **Exclude these objects from the reorganization**—Select this option to exclude tables whose only candidates for a posting index have PRIMARY KEY or UNIQUE KEY constraints that are ENABLE NOVALIDATE.
 - **Generate a script to validate these constraints**—Select this option to generate a script for validating constraints. The script is displayed in the SQL Editor. Whether you should run the script depends on the following:
 - If you want to validate PRIMARY KEY or UNIQUE KEY constraints for use in posting, run the script by pressing **F9** from the SQL Editor. Then close the SQL Editor and skip the rest of this procedure.

- If you want to use the “all columns” posting method or ROWID posting index, close the SQL Editor. Then proceed to the next window in Reorg Manager. A second Unsupported Objects message displays with additional options.
2. From the Unsupported Objects message with four options, select one option:
 - **Exclude these objects from the reorganization**—Select this option to exclude tables whose only candidates for a posting index have PRIMARY KEY or UNIQUE KEY constraints that are ENABLE NOVALIDATE.
 - **Generate a script to validate these constraints**— Select this option to generate a script for validating constraints. The script is displayed in the SQL Editor. Run the script by pressing **F9**. Then close the SQL Editor.
 - **Use all columns of these tables for posting**—Select this option to use “all columns” posting for tables in the message. This method checks all columns in a table to guarantee row uniqueness in its reorganized copy. However, it does not check columns with LONG, LOB, XMLType, object type, REF, and VARRAY datatypes. By default, tables with those datatypes are excluded from a reorganization when this option is selected. If you want to include the tables and are sure that rows can be identified uniquely based on columns with other datatypes, set Space Manager’s ALLOW_LONG_WO_UNIQUE parameter to YES. See [All Columns Posting Method](#) on page 217 for more information.
 - **Use a temporary rowid column on the new tables**—Select this option to use ROWID indexes for tables in the message. These indexes check the ROWIDs of all rows in a table to guarantee row uniqueness in its reorganized copy. Unlike the “all columns” method, ROWID indexes guarantee row uniqueness for tables with LONG, LOB, XMLType, object type, REF, and VARRAY datatypes. However, they require DDL during script execution and do not support the online switch or FastCopy/DBMS_DataPump. Also, ROWID indexes are only used for partitions or subpartitions when a Space Manager parameter is set. If that parameter is not set, you are prompted to reorganize a partition’s parent table. See [ROWID Posting Indexes](#) on page 218 for more information.

Specify Partition Names

Use the Partition Names window to specify names for individual partitions and subpartitions, including local index partitions and subpartitions. This window provides the second and final opportunity to specify names in the Partitioning Wizard. (The first is provided in the Partitioning Method page.)

Names are stored in uppercase characters if you do not enter them with double quotes. Each name can have up to 30 characters. That maximum is shared by all parts of a name, including numbers and parent names. If all parts add up to more than 30 characters, certain parts are truncated. However, sequential numbers are always included.

To specify names for individual objects

1. Specify names for multiple partitions in an object
 - a. Select the table or index in the Object list.
 - b. Click in the **Partition Name** field for a partition and modify its current name. Repeat this step for each partition name you want to modify. Names for partitions in the same object can use different formats.
2. Specify names for a partition and its subpartitions
 - a. Select the partition in the Object list.
 - b. Click in the **Partition** field and modify the partition’s current name.

- c. Click in the **Subpartition Name** field for a subpartition and modify its name. Repeat this step for each subpartition name you want to modify. Names for subpartitions in the same partition can use different formats.
3. Specify name for one subpartition
 - a. Select the subpartition in the Object list.
 - b. Click in the **SubpartitionName** field and modify the current name.
4. Click Next to proceed to the next page. See [Select Scripting Options](#) for next steps.

Run and Monitor Scripts

Run and Schedule Scripts

There are several ways to execute a reorganization script. Space Manager allows you to manually execute a script, to schedule a script, or to store a script without scheduling it. You can execute or schedule a script from the reorganization wizard or from the Job Monitor.

How to Execute a Script

You can execute scripts using the following methods:

- **Run scripts in the Editor**—Some scripts can be opened and executed in the Editor from the reorganization wizard or the Job Monitor. See [Run Scripts from the Editor](#).
 - **i Important:** Live reorganization scripts and scripts that use FastCopy or parallel QSA processes cannot be run from the SQL Editor. They must be scheduled and run by QSA. These scripts contain commands that must be interpreted by QSA. To run a script in QSA, it must be scheduled (even when executed immediately).
- **Schedule scripts**—You can schedule a script from a reorganization wizard, from the Job Monitor, or from script Properties. See [Schedule Scripts](#).
- **Schedule script generation**—You can schedule a recurring script generation process for a LiveReorg script. Use this method to *generate* (and execute) the script at a later time or on a recurring basis. See [Schedule Script Generation](#).

Script Execution Choices in Wizards

When you create a script in a reorganization wizard, you must either run the script immediately (in Editor, or by selecting **Execute Immediately**), schedule the script, store the script, or schedule script generation (LiveReorg only). See [Schedule Script Execution in Wizards](#).

- **i Tip:** As an alternative to creating and executing a script in the Reorg Manager, you can save your Reorg Manager settings as a Reorg Plan to use later. See [Save Reorg Manager Settings](#).

Before Running a Script

Review the following information prior to running a reorganization script.

Important Considerations and Best Practices

Review the following important considerations and best practices prior to running a reorganization script.

Consideration / Best Practice	Description
Stop DML activity for Standard reorg scripts	Before you run a standard reorganization script, stop DML activity against the objects being reorganized. All DML activity should be suspended until the script completes successfully.
Same tables in multiple scripts	If the same tables are included in multiple scripts, do not run the scripts at the same time.
Tables that share referential integrity	If tables that share referential integrity are included in multiple scripts, do not run the scripts at the same time.
Script and Space Manager version compatibility	Execute a script using the same version of Space Manager that was used to generate the script.
Reduce time between script generation and execution	Reorganization scripts should be run as soon as possible after they are generated. If the data structures of the objects in the script are modified between script generation and execution, the DDL and DML statements in the script can cause errors and/or data loss during script execution.
Execute script only once	Each reorganization script should be run only once. If a live reorganization script is run more than once, the second run fails if any of the objects being reorganized have system-named constraints. You can avoid this problem by saving Reorg Manager settings. See Save Reorg Manager Settings on page 81 for more information.
Interruption of interactive execution	If execution fails in the SQL Editor, resolve the problem that caused the failure. Then press F9 to resume execution from the failed statement. Do not ignore an error and resume execution after the failed statement.
Interruption of scheduled execution	If scheduled execution fails, resolve the problem that caused the failure. Then use the Restart function in the Scripts/Job Monitor to restart the script. This function can be used after an error, a cancellation, or a system shutdown. See Restart Scheduled Scripts on page 124 for more information.
Monitor a switch	If the switch for a live reorganization requires user approval, check the Script/Job Monitor to see when a table is ready for the switch. The status of its script is "Waiting for approval".
Cancellation and a switch	Although you can cancel reorganization scripts that are run on a scheduled basis, do not cancel a live reorganization script during the switch stage for any of the tables it contains.
Workaround when you cannot store a script	If you cannot store or save a script, a temporary file for the script is saved in the Reorg Scripts folder (under the Space Manager client folder). Once you can store or save the script, open its temporary file in the SQL Editor. Then save or store the file as a script.

A Space Audit is Performed at Run Time

At script run time, Space Manager audits target tablespaces to see if they still have enough free space for a reorganization. This is a follow-up to the audit done before a reorganization script is generated. See [Review Space Usage](#) on page 91 for more information.

Audit information is displayed near the beginning of a script in a remark that starts with "Auditing reorganization plan". For each target tablespace involved in a reorganization, the information includes net movement and peak movement:

- Net Movement is how much data a tablespace is expected to contain after reorganization compared to how much it contained before reorganization. (It corresponds to the Starting Free Space value minus the Ending Free Space value displayed in the Review Space Usage window in Reorg Manager.)
- Peak Movement is the maximum amount of data to be moved into or out of a tablespace during reorganization. Peak Movement can sometimes be greater than Net Movement for a tablespace. This happens when a tablespace is used on an interim basis or when one or more objects are moved out of a tablespace.

Schedule Scripts

After creating a reorganization script, you can schedule the script to execute the reorganization at a later time. This is useful if you want to run a reorganization during a period when database activity is light. You can schedule a script through one of the reorganization wizards, the Script/Job Monitor, or script Properties.

i Important: Reorganization scripts should be run as soon as possible after they are generated. If the data structures of the objects in the script are modified between script generation and execution, the DDL and DML statements in the script can cause errors and/or data loss during script execution.

The scheduling methods described in this topic allow you to schedule a one-time execution. You can also schedule a live reorganization to run on a recurring basis. This can be done with recurring script generation and can be set up at the time you create the script in the Reorg Manager. This method also provides the advantage of reducing the time between script generation and execution. See [Schedule Script Generation](#) on page 80 for more information.

After scheduling a script, you can monitor and manage script execution in the Script/Job Monitor. See [Manage Scheduled Scripts](#).

About Scheduled Scripts and QSA

Scheduled scripts are executed (and scheduled) by QSA. If you want QSA to execute a script, you must schedule it. Some scripts, such as Live reorganization scripts and scripts that use FastCopy or parallel QSA processes, must be executed by QSA because they contain commands that must be interpreted by QSA. That is why these scripts must be scheduled and cannot be run manually through the Editor.

i Note: Even when you select the **Execute Immediately** option in the reorganization wizard (Schedule Script Execution page), the script is scheduled with QSA.

Schedule Scripts in the Reorg Manager

In the Reorg Manager (or Partitioning Wizard), select the **Run at** option on the Schedule Script Execution page to schedule and run a reorganization script. Scheduled scripts are run by QSA after they are stored in the Space Manager repository. See [Schedule Script Execution in Wizards](#) to learn how to schedule a script in the Reorg Manager.

Schedule Scripts from the Job Monitor

If you do not schedule a script when you store it from the Reorg Manager or the Partitioning Wizard, you can schedule it later from the Script/Job Monitor window.

To schedule a script from the Job Monitor

1. Right-click a script in the Job Monitor and select **Schedule**.
2. In the Schedule Script dialog, select a date and time to execute the script. Click **OK**.

Schedule Scripts from Script Properties

You can also schedule or reschedule script execution with schedule functions in the Script Properties dialog (Job Monitor window).

To schedule a script from Script Properties

1. Right-click a script or script group in the Job Monitor and select **Properties**.
2. Select one of the Scheduling Info options in the Script Attributes tab:
 - Execute Immediately—Select this option to run the script immediately.
 - Schedule to Run at—Select this option to run the script on a certain date at a certain time.
3. Click **OK** to save your schedule and display it in the Scheduled column of the Script/Job Monitor.

i Tip: You can cancel a scheduled script execution with the **Unschedule** function in the Script/Job Monitor. This function prevents execution of scripts at their scheduled time, but leaves them stored in the Space Manager repository.

Schedule Script Execution in Wizards

Use the Schedule Script Execution page of the Reorg Manager or Partitioning Wizard to run, schedule, or store a script. You can also open the script in the Editor to review it (or run it).

Prior to scheduling the script, review space usage information on the **Space Usage Summary** tab. See [Review Space Usage](#) on page 91 for more information.

From the **Schedule Script Execution** page you can do any of the following:

- View (or run) the script in the Editor
- Schedule the script to run once
- Execute the script immediately (schedule script to run immediately)
- Store the script without scheduling it

i Note: For LiveReorg scripts, if you want to schedule the script to run on a recurring basis, return to the Choose Scheduling Method page and select **Schedule Script Generation**. See [Choose a Scheduling Method \(LiveReorg\)](#) and [About Scheduling a Recurring Reorganization](#) for more information.

See [How to Execute a Script](#) to learn more about your script execution options and to review important considerations and best practices.

To view (or run) the script in the Editor

- Click **View in Editor** to open the script in the Editor window to review it before you schedule it.
- You can also run the script interactively from the Editor window. See [Run Scripts from the Editor](#) on page 121 for more information.


i Important: Live reorganization scripts and scripts that use FastCopy or parallel QSA processes cannot be run from the SQL Editor. They must be scheduled and run by QSA. These scripts contain commands that must be interpreted by QSA. To run a script in QSA, it must be scheduled (even when executed immediately).

To schedule the script to run once

1. Select **Run at** to run the script at a certain date and time.

i Note: Reorganization scripts should be run as soon as possible after they are generated. If the data structures of the objects in the script are modified between script generation and execution, the DDL and DML statements in the script can cause errors and/or data loss during script execution.


See [Schedule Scripts](#), to learn more about scheduling scripts.

2. Select a date and time at which you want the scheduled script to run.
3. Enter a name for the script.
4. Click **Submit**. Space Manager stores the scheduled script and displays a confirmation message.
5. Click **Yes** in the prompt to open the Script/Job Monitor (or select **Manage | Job/Script Monitor**).
6. Your script displays in the list of scripts/jobs in the Job Monitor. If not, click  to refresh the list.
7. Use the Script/Job Monitor window to manage your script. You can reschedule, execute, or delete the script. You can monitor status of the script or change the approval option (automatic or wait for approval). You can open the script or view script logs. See [Using Job Monitor to Monitor Scripts](#) on page 129 for more information.
8. After the script is stored, you can close the Reorg Manager or Partitioning Wizard.

To execute the script immediately

1. Select **Execute immediately** to run the script immediately, as soon as it is stored in the Space Manager repository.
2. Enter a name for the script.
3. Click **Submit**. The script is scheduled to run immediately by QSA. Monitor script execution from the Scripts/Job Monitor.

To store the script without scheduling it

1. Select **Store without scheduling** to store the script and schedule it later from the Script/Job Monitor.
2. Enter a name for the script.
3. Click **Submit**. Space Manager displays a confirmation message that the script is successfully stored.
4. Click **Yes** in the prompt to open the Script/Job Monitor (or select **Manage | Job/Script Monitor**).
5. Your script displays in the list of scripts/jobs in the Job Monitor. If not, click  to refresh the list.
6. To schedule the script, right-click the script and select **Schedule**. Select a date and time to execute the script.
 - Or you can right-click the script and select **Properties**. Then select one of the Scheduling Info options in the Script Attributes tab.

i Tip: Click **Auto-Refresh** in the Script/Job Monitor window to specify a frequency at which to automatically refresh the scripts list.

Run Scripts from the Editor

Use the **View in Editor** option in the Reorg Manager (or Partitioning Wizard) to run reorganization scripts interactively from the SQL Editor. As a script runs, you can monitor execution directly from the SQL Editor. You can also use the **View in Editor** option to open a script and review it before you schedule it.

i Important: Live reorganization scripts and scripts that use FastCopy or parallel QSA processes cannot be run from the SQL Editor. They must be scheduled and run by QSA. These scripts contain commands that must be interpreted by QSA. To run a script in QSA, it must be scheduled (even when executed immediately).

To run scripts from the Editor

1. Select **View in Editor** in the schedule window of Reorg Manager or the Partitioning Wizard. The script is displayed in the SQL Editor.
2. Run the script from the SQL Editor in continuous mode or single-step mode:
 - To run the entire script in continuous mode, click **Run** in the toolbar or press **F9**.
 - To run the script in single-step mode (one statement at a time), click **Single Step** in the toolbar or press **F8** from the first statement. Repeat for each statement in the script.
 - i Note:** If the Connection dialog displays, enter the user name and password of an Oracle user with the DBA role or privileges for Space Manager. Then click **OK** to connect to your database.
3. As the script is executed, the SQL Editor scrolls through script statements. The current statement is highlighted. Check the status bar at the bottom of the window. If a statement is executed successfully, processing time is displayed. If an error occurs, an error message is displayed.
4. When script execution is finished, an execution log for all script statements displays in the Spool tab. View this to see script start and stop times, any errors that occurred, and the output of all executed statements.

About Scheduling a Recurring Reorganization

You can schedule a live reorganization to run at a specific interval, date, and time using the Schedule Task wizard. Space Manager saves the reorganization settings as a Reorg Plan (XML file). By saving a Reorg Plan, you can update or complete the settings, generate current reorganization scripts, and schedule recurring reorganizations.

Saving the Reorg Manager settings is different from saving the resulting reorganization script. Saving the settings is more versatile because it accommodates changes that may happen to the objects and database over time.

For example, assume you select a table with five columns to reorganize. You save the Reorg Manager settings as a Reorg Plan and you save the current reorganization script. A week later, you add a sixth column to the table. Then, you open the saved Reorg Plan and generate and run a new reorganization script. Space Manager reorganizes all six columns instead of the original five columns that existed when you saved the file. If you had run the script that you saved a week before, the new table would only have the original five columns and you would lose the sixth column and its data.

To learn how to setup a recurring reorganization for a LiveReorg script (including saving a Reorg Plan), see [Schedule Script Generation](#).

To learn about Reorg Plans, see [Save Reorg Manager Settings](#).

Manage Scheduled Scripts

You can manage scheduled script execution from the Job Monitor and Script Properties dialog. There are functions in both windows that let you run scripts on demand; cancel and reschedule script execution; restart or resume script execution; approve the switch for online scripts; and change approval type for online scripts.

i Note: If you execute a command-line reorganization or schedule a reorganization through Windows Scheduled Tasks, the resulting script displays in the Job Monitor once script generation is complete.

How are the two windows different?

- **The Job Monitor** lets you manage multiple scripts and use functions that apply to multiple scripts. It also lets you open a script in the SQL Editor, open Script Properties for a script, open script logs, and delete scripts from the repository.
- **Script Properties** allows you to manage individual stored scripts. It also allows you to enter script comments.

i Note: Most script management functions are disabled when QSA is not running.

To manage a script from the Job Monitor

1. Select **Manage | Script/Job Monitor**.
2. You can right-click a script or script group in the top pane of the Job Monitor and select an action from the menu.
3. To open a script or retrieve a script log, right-click the script and select **Show Detail**. In the Detail pane, right-click the script and select an action from the menu.

To manage a script from Script Properties

- Right-click a script or script group in the top pane of the Job Monitor and select **Properties**.

i Note: To view full execution information in the Script Properties dialog, right-click a script from the Detail pane of the Job Monitor and select **Properties**.

Retrieve Script Logs

You can display a list of logs for a script with the Retrieve Script Logs function (only available in the Job Monitor). QSA generates an execution log each time a script is run. For each run of a script, its list of logs displays job number, script status, and script start and stop times. Higher job numbers are for more recent runs of a script.

Script logs are stored along with the QSA log in the Log directory under QSA's installation directory. How long log files are kept is determined by the MAX_LOG_AGE parameter. See [QSA Server Agent Parameters](#) on page 189 for more information.

Script logs contain information that can be used by Quest Software Technical Support. This information includes platform version, Oracle version, execution start and stop times, steps performed by a script, and any errors that occur. Logs can be emailed to support.

i Note: The QSA log contains information on all activity performed by the agent. This information includes commands received, processes spawned, and scripts executed. The log for the current date is called qexecd.log. The names of logs for previous dates reflect the date a log was generated (for example, qexecd-01-01-2009.log).

To retrieve script logs in the Job Monitor

1. Right-click a script or script group and select **Show Detail**.
2. In the Detail pane, right-click a script and select **Retrieve Script Logs**.
3. To open a log in the list, double-click the log. The log is displayed in the SQL Editor in read-only mode.

i Note: Logs may be displayed on a list even if they do not exist on the database server. If you try to open a nonexistent log, this error occurs: "Error retrieving script log: There is no log for the script." Log files do not exist on the server after an upgrade of QSA.

4. To view the latest logs for a script group, right-click the group name in the main list and select **Retrieve Script Group Logs**.

Delete Scripts

You can delete stored scripts with the Delete function (only available in the Job Monitor). This function lets you remove stored reorganization scripts from the Space Manager repository. When a script is removed from the repository, it is cleared from the Job Monitor.

Reorganization and partitioning scripts can be deleted before they are executed or after they have been executed successfully. Although you can leave them in the repository for as long as needed after execution, they should not be run again.

To delete a script from the Job Monitor

- Right-click a script or script group in the top pane and select **Delete**.

How Log Files Are Deleted

Execution log files are *not* removed by the Delete function. Instead, they are deleted automatically when they are older than the number of days specified by the QSA Server Agent parameter: `MAX_LOG_AGE`. See [QSA Server Agent Parameters](#) on page 189 for more information.

Restart Scheduled Scripts

When execution of a scheduled script is interrupted, you can use the **Restart** function to start it again from the Scripts/Job Monitor. This function lets you easily resume execution after it is cancelled or stops due to an error, a system failure, or a system shutdown. Execution automatically resumes from the correct point.

Review the following best practices for script restart:

- If execution is interrupted due to a failure, resolve the problem that caused the error before you restart the script. To determine why a script failed, check the Status Message column of the Scripts/Job Monitor. Also check the script log.
- Before you restart a script, do not make changes to the structure of the original table or copy table (if this exists). Also, do not make changes to indexes, constraints, or triggers. Finally, do not change or drop LiveReorg objects or collection triggers. LiveReorg objects should be left in place in the LiveReorg tablespace while a script is in an interrupted state.
- Do not restart a script with the **Execute** or **Schedule** functions. These functions run a script from the beginning instead of from a restart point.
- If you want to review a script before restarting it, double-click the script in the Detail pane of the Scripts/Job Monitor to open it in the SQL Editor. The script is displayed in read-only mode. Changes cannot be made before the script is restarted. If you open the script in another way and make changes, the **Restart** function is disabled afterward and the script's status is set to **Unscheduled**.

To restart a script with the Restart function:

1. If a script was interrupted due to failure, resolve the problem behind the failure. See [Troubleshoot Script Execution Errors](#) for troubleshooting tips.
 - i Tip:** To determine why a script failed, check the **Status Message** column of the Scripts/Job Monitor. Also check the script log. To view the log, right-click the script in the Detail pane and select **Retrieve Scripts Logs**.
2. Right-click an interrupted script in the Scripts/Job Monitor and select **Restart/Resume**.
3. When a message asks if you want to restart a script immediately, click **Yes**.

Restart for Live and Partitioning Scripts

Restart for live reorganization and partitioning scripts is managed by QSA. The agent determines whether a script can be restarted, at what point it should be restarted, and which statements must be run. Only one restart command is used.

The restart point for live reorganization and partitioning scripts is the “point of failure”. This is the point at which execution failed or was cancelled:

- If a script failed before the copy table was created or if it failed due to a collection trigger error, execution restarts with creation of the copy table.
- If a script failed after the copy table was created, execution restarts from the statement on which it failed. This can be at any point after creation of the copy table. Restart does not recreate a table and indexes that were successfully created when a script was first run. This saves time during restart for large tables and tables with many indexes.

How QSA Determines Whether a Script Can Be Restarted

When a live reorganization or partitioning script is restarted, QSA first validates the script. QSA makes sure all LiveReorg commands are scripted correctly and that LiveReorg objects and triggers have not been changed or dropped.

If QSA detects problems with LiveReorg commands or objects, a script cannot be restarted. In this case, the script is given a status of **324 Aborted no restart** or **325 Aborted restart** after you try to restart it. For help with cleanup of LiveReorg objects, contact Quest Software Support. Once cleanup is performed, a new script can be generated for objects in a failed or cancelled script.

How QSA Determines Where a Script Should Be Restarted

If a live reorganization or partitioning script can be restarted, QSA determines where it should restart—with creation of the copy table or at a point after that. To determine the restart point, QSA checks its QUEST_QSA_PDATA table. This table records all tasks performed during each run of a live reorganization or partitioning script.

The agent also “plays back” or reviews commands and statements from the last run of a script and identifies which statements do not need to be repeated and which ones do. This ensures that restart is efficient. LiveReorg commands sometimes must be repeated. SQL statements are never repeated. They are only executed if they failed or were not executed during the last run. For example, if a CREATE INDEX statement was successfully executed for a copy index, the copy index is not created again after restart.

Restart from Creation of the Copy Table

Restart begins with creation of a copy table if:

- Execution was interrupted before a copy table was completed during the last run of a script.
- Execution failed due to a collection trigger error during the last run of a script. See [What to Do If a Trigger Error Occurs](#) on page 220 for more information.

In both cases, execution is initiated with the ALWAYS_RUN and RESTART_POINT commands. The first SQL statement executed is the CREATE TABLE statement that creates a copy of the original table. All remaining SQL statements and LiveReorg commands for the table are then executed.

Restart After Creation of the Copy Table

Restart bypasses creation of a copy table if that table was created during the last run of a script. In this case, execution is initiated with the ALWAYS_RUN command. The first SQL statement

executed is the statement on which execution was interrupted. This can be anywhere after the CREATE TABLE statement that creates a copy of the original table.

If execution failed on any of the statements inside a “parallel block”, execution resumes from the start of the parallel block and the entire block is executed. However, any indexes created during the first run of a parallel block are not created again. (A parallel block is a section of script where parallel QSA processes are used to create indexes or compile dependencies.)

If any WHENEVER, WAIT, or RETRY commands occur after the restart point, all are active and used for the tasks they apply to, including specific tasks within parallel blocks.

- i Note** If the switch is approved before a script is interrupted, it is still approved after the script is restarted. You do not need to approve the switch for the table that was being reorganized at the time of interruption. However, you do need to approve it for all subsequent tables in the script.

Activity While a Script Is Interrupted

If a live reorganization or partitioning script is interrupted before the switch for a table, DML activity can continue against the table. This is the case whether the table is being reorganized with the online switch or the T-Lock switch. Live transactions are posted to the reorganized copy table when the script is restarted.

If a live reorganization or partitioning script is interrupted during the switch, whether DML activity can continue depends on switch style:

- Online switch—When this switch style is used for a table, DML activity can continue against the original table. Live transactions are posted to the reorganized copy table when the script is restarted.
- T-Lock switch—When this switch style is used for a table, activity against the original table should be stopped until after the script is restarted. The T-Lock placed on the original table at the start of the switch stage remains in effect while the script is interrupted.

Logging for Restarted Live and Partitioning Scripts

Execution logs for restarted live reorganization and partitioning scripts contain comments that identify the restart statement. One comment is displayed near the top of the log and reads as follows: “Restarting at statement 39” (where 39 is the statement number in the log). A second comment is displayed at the point where QSA restarted execution. This reads: “Restart playback is now off at line 39”.

Logs also identify which script statements QSA played back and which ones were executed:

- Statements that QSA plays back but does not execute are marked with asterisks.
- Statements that QSA executes do not have asterisks.

In both scripts and logs, the restart command for each table is assigned a sequential number. This is displayed in parentheses beside the command. It associates a table with its parallel blocks (used for tasks performed in parallel). The following is an example of the command and sequential number for the second table in a script:

```
EXECUTE QUEST_EXEC.RESTART_POINT('2');
```

Restart for Standard Scripts

Restart for standard reorganization scripts is managed mainly with restart commands. These commands provide two types of restart—statement-level restart and section-level restart:

- When statement-level restart is used, execution resumes from the statement on which it was interrupted.
- When section-level restart is used, execution resumes from the last “restart point” reached when execution was interrupted. A restart point marks the beginning of each restart section in a script. A restart section consists of a sequence of statements that must be executed when a script is restarted. As a script is executed, QSA keeps track of its progress through restart points.

For both types of restart, restart is initiated with the ALWAYS_RUN command.

Restart for Standard Reorganization Scripts

In most cases, a statement-level restart is used for standard reorganization scripts. Execution resumes from the statement on which it was interrupted. However, if execution stopped inside a parallel block, a section-level restart is used. In this case, execution resumes from the restart point that precedes the parallel block.

Restart Commands

This section describes the commands used for script restart. Like LiveReorg commands, all begin with `QUEST_EXEC`.

ALWAYS_RUN_BEGIN and ALWAYS_RUN_END

The `ALWAYS_RUN_BEGIN` command marks the start of the “always run” section of a script. The `ALWAYS_RUN_END` command marks the end of this section.

The “always run” section is always run first when a script is restarted. It includes a connection string and any session parameters set from Space Manager. It can also include the `ALTER_USER_QUOTA` statement and the `CHECK_ENGINE` statement. (The latter is contained in live reorganization scripts, partitioning scripts, and scripts that use FastCopy or parallel QSA processes).

RESTART_POINT

The `RESTART_POINT` command tells QSA to perform a section-level restart:

- In live and partitioning scripts, a `RESTART_POINT` command precedes the `LW_START_REORG` command for each table. If execution is interrupted before a copy table is created for an original table, execution is restarted from the table’s `RESTART_POINT` command. If execution is interrupted after a copy table is created, execution is restarted from the point at which it failed.

The `RESTART_POINT` command for each table in a script is assigned a sequential number. This is displayed in parentheses beside the command. It associates a table with its parallel blocks (used for tasks performed in parallel). The following is an example of the command and sequential number for the second table in a script:

```
EXECUTE QUEST_EXEC.RESTART_POINT('2');
```

The same number is displayed in the `BEGIN_PARALLEL` command for each of a table’s parallel blocks. Number of parallel QSA processes is displayed beside the sequential number. The following is an example of the `BEGIN_PARALLEL`, sequential number, and number of parallel processes for the second table in a script:

```
EXECUTE QUEST_EXEC.BEGIN_PARALLEL('2', 3);
```

STATEMENT_LEVEL_RESTART

The `STATEMENT_LEVEL_RESTART` command tells QSA to use a statement-level restart:

- In standard scripts, there is one `STATEMENT_LEVEL_RESTART` command per script. It applies to all tables. The command occurs at the beginning of the script. When a statement level restart is performed for a standard reorganization script, execution is restarted from the statement on which it was interrupted.

Monitor Scripts

After you schedule and store a script, you can monitor script execution using the Job Monitor and other dialogs and tools. Space Manager provides the following methods to monitor scripts:

- In the Job Monitor
- Using the Script Properties dialog
- In the SQL Editor
- Using email notifications
- Using Live Reorg Shell

Job Monitor vs Properties Dialog

Both windows display the status of the scripts that are stored in the Space Manager repository for execution by QSA. You can use both the Job Monitor and the Properties dialog to do the following monitoring tasks:

- View script status
- View script execution schedule, as well as execution start and stop times
- View each script statement as it is processed (for currently-executing scripts)

How do the windows differ?

- **Job Monitor**—The Job Monitor provides a list of all stored scripts for a database.
 - Use this window to view the status, script type, start and stop times, and status code for all scripts.
 - If a script fails, the error code is displayed for the individual script that failed.
 - This window allows you to drill down to view the status of each individual script in a group.
 - View script logs for individual script executions.
 - Open a script in the Editor window.

See [Using Job Monitor to Monitor Scripts](#) on page 129 for more information.

- **Group Properties**—The Group Properties dialog allows you to perform tasks on a script group, for example:
 - View or modify the execution schedule
 - Change the approval method for the switch

See [Using Group and Script Properties to Monitor Scripts](#) on page 132 for more information.

- **Script Properties**—The Script Properties dialog allows you to focus on individual stored scripts.
 - Use this dialog to monitor the execution progress of an individual script.
 - View the number of errors and warnings that occur during script execution
 - View total number of statements in a script
 - View the number of statements successfully executed so far
 - View complete script statements for an executing script

See [Using Group and Script Properties to Monitor Scripts](#) on page 132 for more information.

Information on a script is available in both windows as soon as the script is stored in the repository. This is the case even if scripts are not scheduled when they are stored.

Monitor Interactive Execution in Editor

You can monitor interactive execution of reorganization scripts from the SQL Editor. See [Run Scripts from the Editor](#) on page 121 for more information.

Use Email to Monitor Script Status and Results

You can configure Quest Service Agent (QSA) to send notifications/alerts via email to help you monitor the progress of script execution. QSA can be configured to automatically send email notifications at almost any step in the reorganization process or if the script execution fails. You can also configure QSA to send an email containing the reorganization results.

Notifications can be sent in two ways: email (SMTP) or SNMP trap. To enable one or both of these methods, you must first define the parameters in QSA.

- To enable email notifications, see [Enable Email \(SMTP\) Notification](#).
- To enable SNMP trap notifications, see [Enable SNMP Traps Notification](#).

After enabling the email or SNMP trap notification feature, you can configure additional notification events. See [Configure Additional Notification Events](#) on page 138 for more information.

Use Live Reorg Shell

Live Reorg Shell (LRSH) is a command line tool you can use to monitor Live Reorg scripts when you do not have access to the Space Manager desktop application. See [Monitor Scripts Using Live Reorg Shell](#) on page 140 for more information.

Using Job Monitor to Monitor Scripts

The Job Monitor displays the list of all scripts stored in the Space Manager repository for a database. Use the Job Monitor to view script status and monitor script execution.

You can use the Job Monitor to perform the following tasks:


- View a list of scripts currently in the Repository
- View the status of each script
- Monitor execution of a script or script group
- Monitor a live reorganization
- View error codes
- Review execution logs
- Open a script in the SQL Editor

Most scripts remain listed in the Job Monitor throughout their history: before, during, and after execution. Scripts are cleared from the monitor when they are deleted from the repository.


To open the Job Monitor

- In the Explorer, select a database for which you want to view scripts.
- Select **Manage | Script/Job Monitor**. The Job Monitor opens displaying the scripts for the selected database.

To refresh, filter, and modify the display











- To refresh the Job Monitor display, click . For more display options, see [Job Monitor Display Options](#).





To monitor script execution in the Job Monitor

- Select a script or script group from the list in the Job Monitor. To filter the list, use the **Group by** field or click the filter icon  in a column header.
 - Script status and a status code display in the Status Message column. See [Status Codes for Script Execution](#) for code descriptions.
 - If a script fails, the error code is displayed for the individual script that failed. For help troubleshooting errors, see [Troubleshoot Script Execution Errors](#) and [QSA Error Messages](#).
- To see the list of scripts in a script group, right-click the script group and select **Show Detail**.
 - Script status for each individual script displays in the Status Message column in the Detail pane. If a script fails, the error code is displayed.
- To view details about an individual script, right-click a script in the Detail pane and select **Show LiveReorg Detail**. You can use the LiveReorg Detail pane to monitor the progress of a live reorganization from the Job Monitor window. See [Monitor Live Reorganizations](#) on page 133 for more information.
- To view script logs, right-click a script in the Detail pane and select **Retrieve Script Logs**. To view the latest logs for a script group, right-click the group name in the main list and select **Retrieve Script Group Logs**.
- To change the switch approval method for online scripts, right-click the script group in the upper pane and select **Properties**. See [Change Approval Method for Online Scripts](#) on page 135 for more information.
- To learn more about tasks for managing scripts, see [Manage Scheduled Scripts](#).

Toolbar Functions

Use the following toolbar functions to view, monitor, and manage scripts from the Job Monitor window.

Toolbar Button	Function
	Execute the selected script.
	Stop the selected script.
	Restart the selected script.
	Open the selected script in the SQL Editor.
	Open Properties for the selected script or script group.
	Delete the selected script.
	Schedule the selected script.
	Unschedule the selected script.
	Approve the switch.
	Change approval type.

Toolbar Button	Function
	Refresh the Job Monitor window.
	Retrieve script logs for the selected script. Double-click a log in the list to view it in the SQL Editor in read-only mode. Enabled only for scripts in the Detail pane.
	Show/Hide Detail pane.
	Show/Hide LiveReorg Detail pane. Enabled only for Live Reorg scripts in the Detail pane.

i Note: To learn about managing DML Profile scripts, see [Smart Start for DML Profiler](#).

Job Monitor Display Options

The Job Monitor provides several options that allow you to customize the display of scripts, including:

- Group scripts by status or type
- Filter the list of scripts
- Save multiple custom layouts (displays)
- Specify refresh rates for the Job Monitor window and the QSA status display.

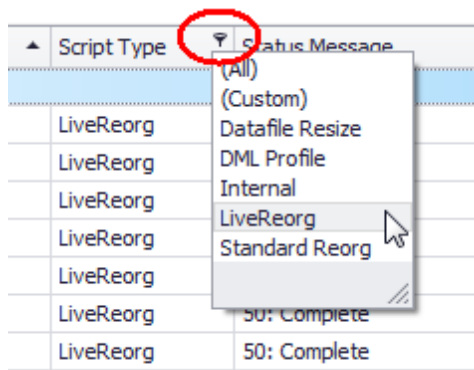
i Tip: Use grouping and filtering together to further narrow the list of scripts.

To group scripts

1. In the Job Monitor toolbar, select a category in the **Group by** list.
2. Select one of the following options:
 - Script Type
 - Script Status
3. To quickly find active scripts, group by *Script Status*.

To filter scripts

1. You can filter scripts by column values. Click ▼ in the column header and select a value.



2. To filter using two or more values or to create an advanced filter, click ▼ and select **Custom**.

3. When the filter is applied, the filter description displays in a box in the footer of the upper pane. To remove the filter, click the close button in the filter box.
4. To modify the current filter, click **Edit Filter** in the filter box.

To save a custom layout (display)

1. Use grouping, filtering, and sorting to create a custom display in the Job Manager window.
2. When finished, click **Layout | Save**.
3. Give your custom layout a name and click **OK**.
4. To apply the layout, click **Layout | Load** and select the custom layout. When applied, the name of the custom layout displays in the status bar at the bottom of the window. If the saved layout includes a filter, it is displayed in the footer of the upper pane.

Refresh Rates

For executing scripts, refreshing the monitor updates it with the statements currently being processed. Refreshing the QSA status panel updates it with information on whether QSA is running, not running, or not installed.

To refresh Job Monitor immediately

- Click  in the Job Monitor toolbar.

To set refresh rate for Job Monitor

1. Click the **Auto-Refresh** button in the Job Monitor toolbar.
2. Select the **Automatically refresh scripts list every n seconds** option. Then specify a refresh rate. The minimum refresh rate is an interval of five seconds.

To set refresh rate for QSA Server Agent status

1. Select **Tools | Options | QSA Server Agent**.
2. Specify a refresh rate. The minimum refresh rate is an interval of one minute.

Using Group and Script Properties to Monitor Scripts

You can use the Properties dialog for scripts and script groups to perform the following tasks:

- View or modify the execution schedule
- Change the approval method for the switch
- Monitor execution progress
- Monitor a live reorganization

To open the Group Properties dialog

- In the Job Monitor, right-click a script or script group and select **Properties**. This dialog provides high-level information for a script or script group.

To change or view schedule information (using Group Properties)

1. Open the Group Properties dialog for a script or script group. Select the Group Attributes tab.
2. In the Scheduling Info section, modify the schedule as necessary. See [Schedule Scripts from Script Properties](#) on page 120 for more information.

To change approval method (using Group Properties)

1. Open the Group Properties dialog for a script or script group. Select the Group Attributes tab.
2. In the Approval Info section, change the approval method as necessary. See [Change Approval Method for Online Scripts](#) on page 135 for more information.

To open the Script Properties dialog

1. In the Job Monitor, double-click a script or script group to open the **Detail** pane.
2. In the **Detail** pane, right-click a script and select **Properties**. This dialog provides detailed information for the selected individual script.

To monitor execution progress (using Script Properties)

1. Open Script Properties for an individual script.
 - a. In the Job Monitor, double-click a script or script group to open the **Detail** pane.
 - b. In the **Detail** pane, right-click a script and select **Properties**.
2. Select the **Execution Information** tab.
3. To monitor execution progress, check statement and count fields in the Execution Info tab.
4. To see how much of a script has been executed, compare the number of statements processed so far (in the **Success Count** field) to the total number of statements in a script (in the **# of statements** field).

i Note: The Script Properties dialog (opened from the Detail pane) displays information only about the selected individual script.

Monitor Live Reorganizations

You can monitor the progress of a live reorganization or partitioning script from the Detail pane and the Live Reorg Detail pane in the Job Monitor. The Live Reorg Detail pane displays transaction and operation statistics for the stages of a live reorganization as they are performed. It lets you see which stages are complete and which stage is underway. See [Stages of a Live Reorganization](#) on page 209 for more information.

During the first three stages, the Live Reorg Detail pane displays the number of live transactions being made against the original table. During the third stage, the Live Reorg Detail pane also displays how many live transactions have been posted to the copy table and how many still need to be posted (outstanding transactions). For each posting session, it displays start and stop times, elapsed time, and posting count.

The value for outstanding transactions allows you to see how much work must be done before the switch. The switch can occur when all live transactions have been posted to the reorganized copy table. To estimate the speed of posting, divide the elapsed time for the last completed posting session by the session's transaction count. You can then use this posting rate to estimate how much time will be required to post outstanding transactions.

- i Tip:** Estimated posting transactions are displayed in the Status Message column of the Script/Job Monitor. The estimates indicate how many transactions need to be posted and how much time is required. (Because they are estimates, they may differ from statistics in the Live Reorg Detail pane.) You can use posting estimates to determine when to approve the switch. The lower the number of transactions, the sooner the switch is made after approval is given.

Use the Live Reorg Detail Pane

The Job Monitor provides two panes (hidden by default) to display detailed information: the **Detail** pane and the **Live Reorg Detail** pane. The Detail pane displays the list of individual table scripts in a live reorg script group. Use this pane to view script status for the individual scripts in a script group. The Detail pane also displays the table name for each individual script.

The Live Reorg Detail pane is only used for live reorganization scripts and partitioning scripts. (The latter apply partitioning schemes via live reorganizations.) No information is displayed for standard reorganization scripts.


To view a script in the Live Reorg Detail pane

1. Select **Manage | Script/Job Monitor** to open the Job Monitor.
2. Right-click a script group and select **Show Detail**. The Detail pane opens displaying the list of scripts in the group.
3. Right-click a script in the Detail pane and select **Show LiveReorg Detail**. The Live Reorg Detail pane opens.
4. Check the progress of the selected script from both panels of the Live Reorg Detail pane as follows:
 - Tables panel—View the name of the table reorganized by that script. A check mark indicates the table has already been reorganized.
 - Live Reorg Progress panel—View this panel to see stages and statistics for the table currently being reorganized. Information is not displayed until the LW_START_REORG command for the table has been executed. If the script has not started, the panel is blank. If the script has completed, summary information is displayed.
5. To display information for another individual script in the group, click that script in the Detail pane.
6. To display information for another script group, click the group name in the Job Monitor top pane.


i Note: Times in the **Live Reorg Detail** pane are server clock times..

i Tip: Information in the Live Reorg Detail pane is automatically refreshed at the rate set for the Job Monitor. To refresh information as needed, press F5.

To open the Detail pane

- » Use one of the following methods to open the Detail pane in the Job Monitor:
 - Select a group name and click  in the Job Monitor toolbar.
 - Right-click a group name and select **Show Detail**.

To open the Live Reorg Detail pane

- » Use one of the following methods to open the Live Reorg Detail pane in the Job Monitor:
 - Select a table script and click  in the Job Monitor toolbar.
 - Right-click a table script and select **Show LiveReorg Detail**.

Status in Live Reorg Detail

The Live Reorg Detail pane uses colored-coded check marks to indicate completion status for the tables in a live reorganization script. It also uses colored-coded check marks to indicate completion status for the stages each table undergoes.

Check marks	For tables	For stages
Green	Is the status for the table currently being reorganized.	Is the status for the stage currently being performed.
Blue	Is the status for tables that have been reorganized.	Is the status for stages that have been completed.

Check marks	For tables	For stages
Red	Is the status for a table if reorg fails on that table.	Is the status for a stage if reorg fails during that stage.
Yellow	N/A	Is the status for a stage if a script is cancelled during that stage.

Stages in Live Reorg Detail

Stage	Description
Stage 1	During Stage 1, a structural copy of the original table is created and all existing data is copied from the original table to the copy table. As the copy table is created, QSA collects information on the live transactions made against the original table.
Stage 2	During Stage 2, the first index is recreated on the reorganized copy table. This index serves as the posting index that QSA uses to copy live transactions to the copy table.
Stage 3	<p>During Stage 3, all remaining indexes on the copy table are created. As this happens, QSA posts live transactions to the copy table and keeps it in sync with the original. First, QSA posts the transactions collected during Stages 1 and 2. Afterward, QSA continually collects and posts transactions made as Stage 3 proceeds.</p> <p>i Tip: To approve the switch from the Live Reorg Detail pane, right-click anywhere in the pane and select Approve Switch.</p> <p>Posting Detail Fields</p> <ul style="list-style-type: none"> • Post Count—This field displays the number of posting sessions that have occurred so far. (A posting session is a single execution of the stored procedure and OCI calls that are used to post live transactions to the copy table.) If the number for Post Count is greater than the number for Current, this means that QSA has attempted one or more posting sessions when there were no transactions to be posted. • Current—This field displays the sequential number of the current posting session. • Previous—This field displays the sequential number of the last completed posting session. <p>i Note: Only fields for the current and last posting sessions are displayed under Posting Detail. Fields for other posting sessions are not displayed.</p>
The switch	<p>In this stage, the switch from original table to reorganized copy table is made. This happens after QSA obtains an exclusive lock on the original and all live transactions have been posted.</p> <p>When the online switch is used, no fields are displayed for the switch stage. When the T-Lock switch is used, the only fields that might be displayed are Session Info fields. However, a T-Lock switch is often so brief that Session Info fields are not displayed.</p>
Stage 4	Tasks for Stage 4 always include dropping the temporary LiveReorg objects created and used by QSA. They can also include recompiling dependencies, validating data against constraints, and dropping the original table.

Change Approval Method for Online Scripts

You can change the approval method (switch option) for online scripts. Approval method determines how the switch proceeds for a live reorganization script or a partitioning script.

Approval method can be changed for online scripts that are waiting to run or waiting to be restarted. A change in approval method applies to all tables in a script for which the switch has not yet been made.

For more information about approval methods (also known as *Switch Options*), see [Select Scripting Options](#) and [More About Scripting Options \(LiveReorg\)](#).

To change approval method

1. In the Script/Job Monitor, do one of the following to change approval method:
 - Right-click a script group and select **Change Approval Type**.
 - Right-click a script group and select **Properties**. Then select the Script Attributes tab.
2. Select one of the approval types.
3. Click **OK** to save your changes and close the dialog.

Approve the Switch

When the switch option for a live reorganization script is **Upon user approval**, you must approve the switch for each table in the script. You can do this from the Scripts/Job Monitor or from its Live Reorg Detail pane using one of the following procedures.

To see if a table is waiting for the switch to be approved, check the **Status Message** column in Scripts/Job monitor. "Waiting for approval" is the status for a script when one of its tables is waiting for approval.

After you approve the switch for one table in a script, check to see when the next table is ready for the switch. How much time passes before approval is required depends on the size of the table being reorganized, the number and size of its indexes, and the level of activity against the table.

To approve the switch from the Scripts/Job Monitor

1. Open the Scripts/Job monitor by selecting **Manage | Script/Job Monitor**.
2. Check the **Status Message** column. "Waiting for approval" is displayed for each live reorganization script with a table that is ready for the switch.
3. Right-click a live reorganization script and select **Approve Switch**.

To approve the switch from Live Reorganization Detail

1. Open the Scripts/Job monitor by selecting **Manage | Script/Job Monitor**.
2. Select a live reorganization script. Information on the script is displayed in the Live Reorganization Detail pane.
3. When "Stage 3 (Waiting for approval)" is displayed for a table in the Live Reorg Detail pane, right-click anywhere in the pane and select **Approve Switch**.

Enable Email (SMTP) Notification

When you are away from the Space Manager client computer, you can monitor scheduled execution with email notifications from QSA. These alert you when scheduled execution fails. QSA sends an email immediately after a failure occurs.

Each email notification identifies script name, execution error, execution start and stop times, database server hostname, and Oracle SID for the database. The subject field displays "Script Failure Notice!".

QSA emails notifications when its SENDMAIL parameters are configured. These parameters allow you to specify who should receive notifications. You can specify both a primary and a secondary recipient. A recipient can be an individual or an email group.

i Important: Before you can use QSA's email notification capability, Oracle's UTL_SMTP package must be installed on the database server. The package must have access to an SMTP mail server or relay server.

Use the following procedure to enable email notification. To configure notifications for other events, see [Configure Additional Notification Events](#).

To enable email notifications

1. Run Space Manager and connect to the target database for QSA. Connect as the Oracle DBA that installed the agent.
2. Select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
3. Select **Configure Quest Server Agent** and click **Next**.
4. If your database resides on a UNIX or Linux server, a login window displays. Enter the login name and password of the UNIX or Linux account that installed the agent.
5. On the QSA Parameters page, set the following SENDMAIL parameters in order to enable email notifications (values are required for all parameters except SENDMAIL_TO1):

QSA Parameter	Description
SENDMAIL_DOMAIN	Specifies the domain name of the mail server or relay server. For example: yourcompany.com.
SENDMAIL_HOST	Specifies the address or fully-qualified name of the mail server or relay server host, for example: www.quest.com.
SENDMAIL_PORT	Specifies the port number on which the mail server or relay server is listening.
SENDMAIL_FROM	Specifies who sent the email notification. The text string you enter should be the email address of an individual or group, such as Administrator@yourcompany.com.
SENDMAIL_TO	Specifies the email recipient. Must be the email address of an individual or group, such as DBA@yourcompany.com. The address must be valid. Only one address may be entered.
SENDMAIL_TO1	Specifies a second recipient for email notifications. To specify three or more recipients, insert a row for each recipient in the QUEST_EXEC_PARAMETER table, where NAME is SENDMAIL_TO# and VALUE is a valid email address. Each SENDMAIL_TO# entry should increment by 1; for example, SENDMAIL_TO3, SENDMAIL_TO4. If there is a break in the number sequence, emails are not sent to recipients after the break. If you delete a recipient row, update the NAME rows that follow so the number sequence is preserved. You can specify a maximum of 99 recipients.

6. Click **Next** and then **Finish** to save your entries and close the QSA Installer.
7. After enabling email notification, see [Configure Additional Notification Events](#) to learn how to configure notifications for other events.

i Note: To disable SMTP email notification, set the SENDMAIL_PORT value to "0".

Enable SNMP Traps Notification

You can monitor scheduled execution for reorgs using Simple Network Management Protocol (SNMP) Trap notifications from QSA. If you enable these notifications, QSA alerts you when a scheduled execution fails. QSA sends a notification immediately after a failure occurs.

Use the following procedure to enable SNMP trap notification. To configure notifications for other events, see [Configure Additional Notification Events](#).

i Note: In addition to using an SNMP trap watcher program, you can also use Quest Foglight® to receive SNMP notifications.

To enable SNMP Trap notifications

1. Run Space Manager and connect to the target database for QSA. Connect as the Oracle DBA that installed the agent.
2. Select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
3. Select **Configure Quest Server Agent** and click **Next**.
4. If your database resides on a UNIX or Linux server, a login window displays. Enter the login name and password of the UNIX or Linux account that installed the agent.
5. On the QSA Parameters page, set the following SNMP parameters in order to enable SNMP trap notifications.

QSA Parameter	Description
SNMP_TO_HOST	Enter the host name for the SNMP monitor.
SNMP_PORT	Enter the port number that the SNMP monitor is listening on.
SNMP_COMMUNITY	Enter the name of your SNMP community (or any unique string value).

6. Click **Next** and then **Finish**.
7. Stop and then restart QSA from the **Tools | Admin | Quest Server Agent** menu.
8. After enabling SNMP trap notification, see [Configure Additional Notification Events](#) to learn how to configure notifications for other events.

i Note: To disable SNMP trap notification, set the SNMP_PORT value to "0".

Configure Additional Notification Events

After enabling SMTP email or SNMP trap notification methods, you can specify additional notification events by configuring parameters in Quest Service Agent (QSA). You can specify which events will trigger a notification and which notification method to use for each event.

In addition, you can configure QSA to send a notification when a LiveReorg finishes that includes the reorganization results. This option is available for the SMTP email method only.

To configure notification details (SMTP email or SNMP trap)

1. Run Space Manager and connect to the target database for QSA. Connect as the Oracle DBA that installed the agent.
2. Select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
3. Select **Configure Quest Server Agent** and click **Next**.

4. If your database resides on a UNIX or Linux server, a login window displays. Enter the login name and password of the UNIX or Linux account that installed the agent.
5. In the QSA Parameters page of the QSA Installer, for each event you want to monitor, enter one of the following four values based on the notification method you want to use (and have enabled):

Value	Description
"NONE"	Notification is disabled.
"EMAIL"	Send SMTP email notification.
"SNMP"	Send SNMP trap notification.
"BOTH"	Send both SMTP email and SNMP trap notification.

6. Enter one of the values listed above into the value field for each parameter that corresponds to an event you want to monitor. Select from the following parameters:

i Note: To view all parameters, clear the **Show only required fields with missing value** checkbox.

Parameter	Description
SEND_GROUP_STATUS	Send notification of a static change to the group status.
SEND_SCRIPT_START	Send notification when script starts (101).
SEND_SCRIPT_RESTART	Send notification when script is restarted (104).
SEND_SCRIPT_UNCHEDULED	Send notification when script is unscheduled (110).
SEND_SCRIPT_COMPLETE	Send notification when script completes (210).
SEND_SCRIPT_CANCELLED	Send notification when script is cancelled (310).
SEND_SCRIPT_ABORT	Send notification when script aborts (320).
SEND_SCRIPT_HALTED	Send notification when script is halted (370).
SEND_SCRIPT_SCHEDULED	Send notification when a script is scheduled (410).
SEND_LR_WAIT_WINDOW	Send notification when a LiveReorg is waiting for window (620).
SEND_LR_IN_WINDOW	Send notification when a LiveReorg is in the window (621).
SEND_LR_WAIT_APPROVAL	Send notification when a LiveReorg is waiting for approval (630).
SEND_LR_APPROVED	Send notification when a LiveReorg is approved (631).
SEND_LR_START	Send notification when a LiveReorg starts (710).
SEND_LR_RESTART	Send notification when a LiveReorg restarts (715).
SEND_LR_COPY_START	Send notification when a LiveReorg table copy starts (720).
SEND_LR_COPY_COMPLETE	Send notification when a LiveReorg table copy completes (730).

Parameter	Description
SEND_LR_SWITCH_START	Send notification when a LiveReorg starts to switch (740).
SEND_LR_SWITCH_COMPLETE	Send notification when a LiveReorg switch completes (750).
SEND_LR_COMPLETE	Send notification when a LiveReorg completes (760). Setting this parameter sends an email without a results report. To include a results report, use one of the SEND_LR_COMPLETE parameters listed below.

i Tip: The parameter prefix indicates the level of the event. SEND_SCRIPT prefixes script-level events, which occur no more than once per script execution. SEND_LR prefixes live-reorganization-level events, which occur one or more times depending on the number of live reorganization objects in the script.

To configure SEND_LR_COMPLETE parameters (SMTP email only)

1. To specify the type of results report you want to include in a notification that is sent after a LiveReorg finishes, select from the following parameters:

Parameter	Results Report Type	Description
SEND_LR_COMPLETE	No report	No report
SEND_LR_COMPLETE_DATA	Detail objects, summary by object type and total savings reports.	This report is the most detailed and includes the object details, detailed storage data, and a storage data summary by object.
SEND_LR_COMPLETE_SUM	Summary by object type and total savings reports.	This report delivers a summary of the storage data by object.
SEND_LR_COMPLETE_TOTAL	Total savings report.	This report lists the storage data totals only.

i Note: To use this feature, you must also enable the type of notification method you want to use. See [Enable Email \(SMTP\) Notification](#) and [Enable SNMP Traps Notification](#) for more information.

Monitor Scripts Using Live Reorg Shell

Live Reorg Shell (LRSH) is a command line tool you can use to monitor Live Reorg scripts. Many of the tasks you can perform using the Scripts/Job Monitor you can perform using Live Reorg Shell.

This tool can be useful for server Administrators and others who do not have access to the Space Manager desktop application, but would like to monitor or run Live Reorg scripts.

Live Reorg Shell vs Desktop Application

When to use Live Reorg Shell

Use Live Reorg Shell when you do not have access to the Space Manager desktop application, but you need to stop a script, run a script, approve a switch, or just monitor a job.

How to access Live Reorg Shell

Live Reorg Shell is installed on the server along with QSA. You can access Live Reorg Shell directly from the server or remotely. To access Live Reorg Shell remotely, use Telnet or any similar secure-shell remote-access utility.

What you can do using Live Reorg Shell

Live Reorg Shell allows you to perform many of the same tasks you normally perform using the Scripts/Job Monitor. You can do the following using Live Reorg Shell:

- Start a job
- Stop (Cancel) a job
- Restart a job
- Approve a switch
- View job lists and script lists
- View job logs
- View scripts

! **Important:** You cannot use LiveReorg Shell to clean up jobs. If you cancel a job using LiveReorg Shell, you must then use the Space Manager application to perform clean-up tasks, if required.

How to Use Live Reorg Shell

To start Live Reorg Shell

1. Start Telnet or other secure-shell remote-access utility.
2. Connect to the server where the database is located.
3. Using the command line, navigate to the directory containing the database on which the scripts you want to monitor are running.
4. When you are in the database directory, at the prompt type `./lrsh` to start Live Reorg Shell.
5. When prompted, enter your Oracle user name and password.
6. At the lrsh prompt, type commands to monitor jobs or perform tasks.

To get help (list commands)

- At the lrsh prompt, type `help`. Live Reorg Shell lists all relevant commands and their descriptions.

i **Note:** Live Reorg Shell help is context-sensitive. LRSB lists only the commands that are relevant to the level at which the prompt is currently located.

To monitor jobs and scripts

1. After starting LRSB, at the lrsh prompt, type `list` to list the current jobs for this database. A list of jobs (script groups) displays, including start, schedule, and finish dates, and job status.

i **Note:** If the list is incomplete, type ENTER to display additional jobs (script groups).

2. To select a job, type the job number after the list displays. The job's status information displays.

i **Note:** To refresh the job status information, press ENTER.

3. After selecting a job, you can view all scripts or take action on this job.
 - To view all scripts in the job, type `list`. The list includes the table name and the script status.
 - To take action on the job, type a command to start, cancel, restart, un-schedule, or approve the job (script group).
 - i Note:** Type `h` here to list relevant commands.
4. After the script list displays, type `a` to approve the script waiting for approval.
5. To select a script, type the script number (after the script list displays).
6. After you select a script, you can view log files for the selected script.
 - a. To view script log files, press ENTER after selecting the script.
 - To select and view a particular log, type the log file number. Press ENTER to view more log file data.
7. After completing a task, you can type `q` to go up a level and/or to eventually exit from LRSB.

List of Commands

Use the following commands to start Live Reorg Shell or to display run options.

```
lrsh           Starts Live Reorg Shell
lrsh.sh -?    Displays Live Reorg Shell information and a list of run options.
```

The following table lists some of the most-frequently used commands in Live Reorg Shell.

Command	Abbreviation	Description
quit exit	q	Use to exit out of Live Reorg Shell. Type <code>q</code> at any prompt in LRSB to go back up one level from a job, script, or action.
help ?	h	Lists all relevant commands and their descriptions. i Note: Lists only the commands relevant to the level at which the prompt is located.
list	ls	At the database level, lists the jobs for that database. At the job level, lists the scripts in that job.
<i><job/script number></i>		After a list displays, type the job or script number to select that job or script. Then you can perform an action or view details of that job/script.
Press the ENTER key		Press ENTER after an incomplete list displays to view more list items. Press ENTER after a job is selected to refresh the status. Press ENTER after a script is selected to view log files.
r		Run the selected job (script group). i Note: After running or restarting, press ENTER to refresh job status.
r		Restart an interrupted job (script group). i Note: After running or restarting, press ENTER to refresh job status.

Command	Abbreviation	Description
c		Cancel (pause) the selected job (script group).
a		Approve the selected job (script group).
u		Un-schedule the selected job (script group).

i Note: Live Reorg Shell commands are case-sensitive.

i Tip: To exit Telnet, type `exit`.

Manage and Monitor QSA

As you monitor and manage scheduled scripts from the Scripts/Job Monitor window, you can also monitor and manage the Quest Server Agent (QSA). This allows you to easily check the status of QSA, which is used to run stored scripts. Most management functions can only be used when QSA is running.

Check QSA Status

You can check the status of QSA from the QSA Status panel at the top of the Scripts/Job Monitor window. You can also check its status in the QSA field in the top-right corner of the Script Properties dialog.

QSA can be in one of four states:

- **Running**—When QSA is running for a database, *Running* is displayed in the monitor. *QSA is running* is displayed in the Script Properties dialog.
- **Not running**—When QSA is installed but not running for a database, *Not running* is displayed in the monitor. *QSA is not operable* is displayed in the Script Properties dialog. QSA must be running before scheduled scripts can be run.
- **Not installed**—When QSA is not installed for a database, *Not installed* is displayed in the monitor. *QSA is not operable* is displayed in the Script Properties dialog.
- **Obsolete**—When QSA is installed for a database, but not current, *Obsolete* is displayed in the monitor. QSA is considered to be current when its version number is the same as Space Manager's version number.

QSA starts automatically during installation on UNIX servers and local Windows servers (those that reside on your Space Manager client computer). To start QSA on a remote Windows server, use the Windows Services window on the server.

i Notes:

- You can specify the interval at which Space Manager checks the Server Agent status in Options (**Tools | Options | Server Agent**).
 - If you are running Space Manager in an active-active OPS or RAC environment, the acronym **RAC** is displayed beside QSA Status panel. The Space Manager client must be connected to the instance used for QSA before you can start and stop QSA.
 - If QSA is not installed (or not current), you can install (or upgrade) it using the QSA Installer or the manual install script available for UNIX and Linux. See [About QSA Server Agent](#) on page 184 for more information. For detailed information about installing the Quest Server Agent, see the *Space Manager with LiveReorg Installation Guide*.

Start QSA from the Scripts/Job Monitor


In most cases, QSA starts automatically when it is installed. It then runs continuously, waiting to execute scripts at their scheduled times. Normally, QSA only needs to be restarted if it was stopped using the Stop Quest Server

Agent function.

The Start Quest Server Agent function can be used to start QSA on local Windows servers (those that reside on your Space Manager client computer) and on UNIX and Linux servers to which you have both SSH and SFTP access. See [Start QSA Under Special Circumstances](#) on page 186 for more information about starting QSA with other configurations.

i Note: If QSA is not running when a script's scheduled execution time occurs, you have 10 minutes in which to start it. The script is executed when QSA starts. If QSA starts after the 10-minute time window ends, the script aborts with this error: "QSA-20312: Scheduled launch time expired". To run a script whose launch time has expired, execute it on demand or reschedule it from the Scripts/Job Monitor.


To start QSA from the Scripts/Job Monitor

1. Select **QSA | Start Quest Server Agent** or click  in the monitor toolbar.
2. If your database resides on a UNIX or Linux server, the server login dialog displays. Enter the login name and password of the UNIX or Linux account that runs QSA.
3. When the *Quest Server Agent Started* message displays, click **OK** to clear it.
4. Check the QSA Status panel at the top of the monitor to confirm that QSA is now running.

Restart QSA from the Scripts/Job Monitor

Use the Restart Quest Server Agent function when you do not have the permission level necessary to start the Quest Server Agent. The Restart Quest Server Agent function automatically stops and restarts the agent. This only requires the same permission level as that for stopping the agent.


To restart QSA from the Scripts/Job Monitor

1. Click  in the monitor toolbar.
2. After the agent restarts, a *Quest Server Agent restarted* message displays.

Stop QSA from the Scripts/Job Monitor

The Stop Quest Server Agent function can be used to stop QSA on local Windows servers (those that reside on your Space Manager client computer) and on UNIX and Linux servers to which you have both SSH and SFTP access. See [Stop QSA Under Special Circumstances](#) on page 187 for more information about stopping QSA with other configurations.

To stop QSA from the Scripts/Job Monitor

1. Select **QSA | Stop Quest Server Agent** or click  in the monitor toolbar.
2. When the *Quest Server Agent Stopped* message displays, click **OK** to clear it.
3. Check the Quest Server Agent Status panel at the top of the monitor to confirm that QSA is no longer running.

i Tip: You can also start, restart, and stop QSA using the **Tools | Admin | Quest Server Agent** menu. See [Start or Stop QSA Server Agent](#) on page 185 for more information.

Monitor Statistics Collection Scripts

i Note: Analysis Groups is a discontinued feature in Space Manager. This topic is provided for users who may continue to run existing statistics-collection scripts that were created in a previous version of Space Manager.

As you monitor statistics collection scripts, be aware that three scripts can be run for one analysis group:

- Collection script—This is used to collect statistics. It is the only script that remains displayed in the Scripts/Job Monitor after collection is performed.
- Collection rescheduler script—This is used to schedule statistics collection. Collection rescheduler scripts have an “s” in parentheses beside their names, for example “Rescheduler (s)”.
- Collection worker script—This tells QSA to perform collection using parallel processes. Collection worker scripts have a “p” in parentheses beside their names, for example “Worker (p)”.

The statistics collection script runs first. Immediately after this script completes, its status is “Completed”. After the rescheduler script runs, the collection script’s status is changed to “Scheduled”. If a collection script is scheduled to run only once, it is cleared from the monitor when it completes.

If QSA parallel processing is selected for an analysis group, the worker script runs next. When it completes, its status is displayed as “Completed” and it is cleared from the Scripts/Job Monitor (it remains in the monitor if statistics collection is aborted).

The rescheduler script runs last. After it completes, its status is “Completed”.

Manage Scheduled Tasks

A scheduled task is created when you schedule a Space Manager activity such as report generation or script generation. When you schedule one of these activities, a Windows scheduled task is created. You can view and manage these tasks using Space Manager.








Scheduled tasks that are managed using the Scheduled Tasks window include:

- Scheduled script generation for Live Reorg scripts. See [About Scheduling a Recurring Reorganization](#) on page 122 for more information.
- Scheduled report generation, such as Predicted Benefit Reorg reports.

When a scheduled task is executing, Space Manager runs in the background.

To manage scheduled tasks

1. Select **Manage | Scheduled Tasks**. The Space Manager Scheduled Tasks window opens.
2. Review the list of scheduled tasks. The grid displays the status of each task and the next scheduled run time.
3. Select a task and click one of the following toolbar buttons to perform the action. Or right-click a task and select from the menu.

	Click to modify the selected task's schedule.
	Click to enable selected task.
	Click to disable selected task.
	Click to run selected task.
	Click to run selected task interactively.
	Click to delete selected task.
	Click to refresh.

i Note: You can also manage tasks and view task properties using the Windows Task Scheduler.

Run Reports

Space Manager provides Reorganization reports to identify the benefits of reorganizing tables or indexes and to capture historical data on reorganization activity.

Space Manager provides Size and Growth reports for capacity and reorganization planning.

All reports can be viewed, printed, and exported. Space Manager allows you to easily export a report and then email it to colleagues. You can export a report to one of a number of different file formats, including PDF, Excel, or a text file.

i Notes:

- Reports are generated from statistics stored in the Space Manager Repository. To accumulate statistics in the Repository, make sure the Repository is updated on a regular basis.
- The date that a Repository was last updated is displayed in the Repository Status panel in the Explorer window when a database is selected.
- To check or modify the update schedule, select **Tools | Admin | Change Repository Update Schedule**.

Report	Description
Reorganization Reports	<p>There are two types of Reorganization reports you can generate.</p> <ul style="list-style-type: none"> • The <i>Predicted Reorganization Benefit</i> report shows you the amount of space you can reclaim if you reorganize a selected table or index. • The <i>Reorganization History</i> report shows which tables or indexes were reorganized during the selected time period, which allows you to extrapolate space savings due to reorganization activity. For planning reorganizations in the future, this report provides you with historical information on how much time it took to run reorganizations on specific objects. <p>See Generate Reorganization Reports on page 147 for more information.</p>
Size and Growth Reports	<p>There are three types of Size and Growth reports you can generate.</p> <ul style="list-style-type: none"> • The <i>Largest Segments Report</i> identifies the largest segments in a database. See Generate Largest Segments Report on page 151 for more information. • The <i>Fastest Growing Segments Report</i> identifies the segments with the highest growth rate. See Generate Fastest Growing Segments Report on page 152 for more information. • The <i>Tablespace Growth Report</i> predicts growth trends for individual tablespaces. See Generate Tablespace Growth Report on page 152 for more information.
Reorganization Candidates Reports	<p>There are two types of reports you can generate to identify Reorganization Candidates.</p> <ul style="list-style-type: none"> • The <i>Tables in Need of Reorganization Report</i> identifies the tables with highest reorg need.

Report	Description
	<ul style="list-style-type: none"> The <i>Indexes in Need of Reorganization Report</i> identifies the indexes with the highest reorg need. <p>See Generate Reorganization Candidates Report on page 150 for more information.</p>

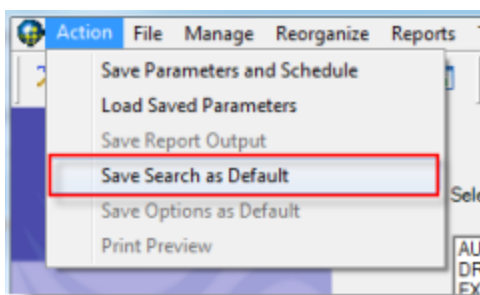
Generate Reorganization Reports

There are two types of Reorganization reports you can generate.

- The *Predicted Reorganization Benefit* report shows you the amount of space you can reclaim if you reorganize a selected table or index.
- The *Reorganization History* report shows which tables or indexes were reorganized during the selected time period, which allows you to extrapolate space savings due to reorganization activity. For planning reorganizations in the future, this report provides you with historical information on how much time it took to run reorganizations on specific objects.


To generate a Predicted Reorganization Benefit report

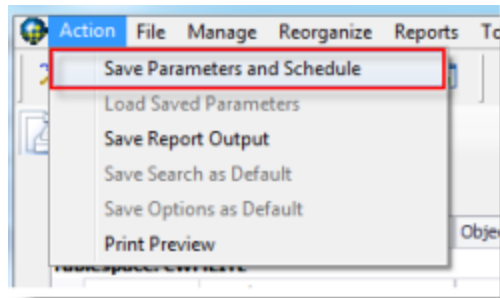
- In the main menu, select **Reports | Reorganization Reports | Predicted Reorganization Benefit Report**.
- If historical reports exist, the wizard opens on the Historical Reports page. Select the **Create new report** option and click **Next**. If no historical reports exist, the wizard opens on the Select Objects page.
- On the Select Objects page of the wizard, from the **Selection Mode** drop-down list, select a category to use for displaying objects.
 - i Note:** When using the *Table* selection mode, you can use incremental search to find objects. Only the first 20 search results are displayed in the drop-down list.
- To select the objects you want to include in the report, use the left/right arrows to move selected objects to the right pane.
 - i Tip:** Select **Action | Save Search as Default** from the main menu to save the search criteria as your default search criteria.




When you have finished selecting objects, click **Next**.

- On the Report Options page of the dialog, specify options to further define the result set. The Predicted Reorganization Benefit report uses data and statistics from the Space Manager repository.
 - i Tip:** Select **Action | Save Options as Default** from the main menu to save the specified options as your default report options (saves only the report options, not search criteria).
- Click **Finish** to generate and display the report.

7. After the report displays, you can perform the following actions from the report window:
 - To reorganize an object, right-click the object in the report and select **Reorg Manager**.
 - To export the report as a pdf file (or other file format), click . See [Preview and Export Reports](#) on page 153 for more information.
 - To save report parameters to use when generating subsequent reports or to schedule a report generation, select **Action | Save Parameters and Schedule**. See [Save report parameters or schedule report generation](#).



To generate a Reorganization History report

1. In the main menu, select **Reports | Reorganization Reports | Reorganization History Report**.
 2. If historical reports exist, the wizard opens on the Historical Reports page. Select the **Create new report** option and click **Next**. If no historical reports exist, the wizard opens on the Select Objects page.
 3. On the Select Objects page of the wizard, from the **Selection Mode** drop-down list, select a category to use for displaying objects.
 - i Note:** When using the *Table* selection mode, you can use incremental search to find objects. Only the first 20 search results are displayed in the drop-down list.
 4. To select the objects you want to track, use the left/right arrows to move selected objects to the right pane.
 - i Tip:** Select **Action | Save Search as Default** from the main menu to save the search criteria as your default search criteria.
 5. Specify the time period. Click **Next**.
 6. On the Report Options page of the wizard, enter options to further define the result set. You can include dependent objects, show only the last reorganization for each object, and group the objects by Reorg date, Reorg ID, Tables and their dependencies, Target tablespace, or Owner.
 - i Tip:** Select **Action | Save Options as Default** from the main menu to save the specified options as your default report options (saves only the report options, not search criteria).
 7. Click **Finish** to generate and display the report.
 8. After the report displays, you can perform the following actions from the report window:
 - To reorganize an object, right-click the object in the report and select **Reorg Manager**.
 - To export the report as a pdf file (or other file format), click . See [Preview and Export Reports](#) on page 153 for more information.
 - To save report parameters to use when generating subsequent reports or to schedule a report generation, select **Action | Save Parameters and Schedule**. See [Save report parameters or schedule report generation](#).
- i Tip:** To view the details of an object's reorganization, you can right-click an object in the report window and select **Details**.

Launch the Reorg Manager from a report

You can launch the Reorg Manager from the report window.

- » In the report, right-click the object you want to reorganize and select **Reorg Manager**.

Save report parameters or schedule report generation

If you selected **Action | Save Parameters and Schedule**, in the Save and Schedule Process dialog, do one of the following:

- To save report parameters, enter a name for the saved parameters, then select a database connection, and click **Save without Scheduling**.
- To schedule report generation, enter a name for the scheduled task, select a database connection, and click **Schedule**. In the Schedule Task dialog, specify schedule options and provide your Windows user name and password.

i Notes:

- To email the report after generation, in the Save and Schedule Process dialog, select an output file format and specify email settings.
- When you schedule report generation, the report parameters are also saved, so you can use them for subsequent report generation.
- A Windows scheduled task is created when you schedule report generation. See [Manage Scheduled Tasks](#) on page 154 for more information.

To open an existing report

1. In the main menu, select **Reports | Reorganization Reports | <report type>**.
2. On the Historical Reports page, select a report from the list and click **Finish**.

i Note: Click **Next** to go to the Report Options page where you can modify report options before viewing the report.

To generate a report using saved parameters

1. In the main menu, select **Reports | Reorganization Reports | <report type>**.
2. In the report window, from the main menu, select **Action | Load Saved Parameters**. Select from the named parameters (saved as xml files).

Export a report

You can export a report to one of a number of different file formats, such as PDF or Excel. See [Preview and Export Reports](#) on page 153 for more information.


i Tips:

- You can manage scheduled reports by selecting **Manage | Scheduled Tasks** from the main menu. See [Manage Scheduled Tasks](#) on page 154 for more information.
- To view the properties of an object, you can right-click an object in the report window and select **Object Properties**.

Generate Reorganization Candidates Report

Reorganization Candidates reports help you identify objects in need of reorganization. You can generate a report to find either the tables or the indexes with the highest reorg need.

To generate a Tables in Need of Reorganization report

1. Select **Reports | Reorganization Candidates | Tables in Need of Reorganization**.
i Note: Alternatively, you can click  and select **Tables in Need of Reorganization**.
2. Specify the number of tables to display.
3. (Optional) Select minimum segment size.
4. Click **Generate** to generate the report.


i Note: After the report displays, click a column header to sort rows by that column.

To launch Reorg Manager

You can launch the Reorg Manager directly from a *Tables in Need of Reorganization* report.

- In the report, right-click a table name in the list of tables in need of reorganization and select **Reorg Manager**.

To generate a Indexes in Need of Reorganization report

1. Select **Reports | Reorganization Candidates | Indexes in Need of Reorganization**.
i Note: Alternatively, you can click  and select **Indexes in Need of Reorganization**.
2. Specify the number of indexes to display.
3. (Optional) Select minimum segment size.
4. Click **Generate** to generate the report.

i Note: After the report displays, click a column header to sort rows by that column.

Export a report

You can export a report to one of a number of different file formats, such as PDF or Excel. See [Preview and Export Reports](#) on page 153 for more information.

Generate DML Profile Reports

To generate a DML Profile report

1. In the Explorer, right-click an object or segment and select **DML Profile | DML Profile Report**.
2. Select a **From** and **To** date at the bottom of the report, if needed.
3. Select each of the tabs to view report details.

i **Tips:**

- To create a DML Profile, right-click an object or segment and select **DML Profile | Create/Modify Collection**. See [Smart Start for DML Profiler](#) on page 22 for more information.
- To open the DML Profile Collections window for an owner in the selected database, expand the DML Profile Collection node in the database tree-list and select the owner.

Export a report

You can export a report to one of a number of different file formats, such as PDF or Excel. See [Preview and Export Reports](#) on page 153 for more information.

Generate Largest Segments Report

Size and Growth reports provide information to help you plan future capacity. The *Largest Segments* report identifies the largest segments in a database.

Use the *Largest Segments* report to show the growth rate of the largest segments in a database. A convenient graph shows the distribution of the largest segments. You can generate a report to identify the largest segments per tablespace or owner.

To generate a Largest Segments report

1. Select **Reports | Size and Growth Reports | Largest Segments**.

i **Note:** Alternatively, you can click  and select **Largest Segments**.

2. Specify the number of largest segments to include in the report.
3. Specify whether to select the largest segments per tablespace or per owner.
4. (Optional) Select minimum segment size.
5. Click **Generate**.

i **Note:** After the report displays, click a column header to sort rows by that column.

To launch Reorg Manager

You can launch the Reorg Manager directly from a *Largest Segments* report for any tables listed.

- In the report, right-click a table name in the list and select **Reorg Manager**.

Other Size and Growth reports

- [Generate Fastest Growing Segments Report](#)
- [Generate Tablespace Growth Report](#)

Export a report

You can export a report to one of a number of different file formats, such as PDF or Excel. See [Preview and Export Reports](#) on page 153 for more information.

Generate Fastest Growing Segments Report

Size and Growth reports provide information to help you plan future capacity. The *Fastest Growing Segments* report identifies those segments with the highest growth rate in a database.

Use the *Fastest Growing Segments* report to show the growth rate of the fastest growing segments in a database. You can generate a report to identify the fastest growing segments per tablespace or owner.

To generate a Fastest Growing Segments report

1. Select **Reports | Size and Growth Reports | Fastest Growing Segments**.

i Note: Alternatively, you can click  and select **Fastest Growing Segments**.

2. Specify the number of segments to include in the report.
3. Specify whether to select segments with the highest growth rate per tablespace or per owner.
4. (Optional) Select minimum segment size.
5. (Optional) Select the date range you want Space Manager to use when calculating highest growth rate.
6. Click **Generate**.

i Note: After the report displays, click a column header to sort rows by that column.

To launch Reorg Manager

You can launch the Reorg Manager directly from a *Fastest Growing Segments* report for any tables listed.

- In the report, right-click a table name in the list and select **Reorg Manager**.

Other Size and Growth reports

- [Generate Largest Segments Report](#)
- [Generate Tablespace Growth Report](#)

Export a report

You can export a report to one of a number of different file formats, such as PDF or Excel. See [Preview and Export Reports](#) on page 153 for more information.

Generate Tablespace Growth Report

Size and Growth reports provide information to help you plan future capacity. The *Tablespace Growth* report predicts growth trends for individual tablespaces.

Use a *Tablespace Growth* report to predict the growth rate and future size of all tablespaces in a database. The report provides you with current tablespace size, projected growth rate, and projected tablespace size per quarter for each selected tablespace over the next twelve months.

To generate a Tablespace Growth report

1. Select **Reports | Size and Growth Reports | Tablespace Growth**.

i Note: Alternatively, you can click  and select **Tablespace Growth**.

2. Select the **Exclude Oracle internal tablespaces** checkbox in order to exclude the internal tablespaces from your report.
3. Click **Generate** to generate the report.

i Note: After the report displays, click a column header to sort rows by that column.

Other Size and Growth reports

- [Generate Largest Segments Report](#)
- [Generate Fastest Growing Segments Report](#)


Export a report











You can export a report to one of a number of different file formats, such as PDF or Excel. See [Preview and Export Reports](#) on page 153 for more information.


Preview and Export Reports

You can export a report to one of a number of different file formats, such as PDF and Excel. You can customize the report layout prior to exporting. You can also customize and then print a report without saving it as a file.

To preview and export a report

1. After generating a report, click  in the report window to open the Print Preview window.
 - i Note:** In the DML Profile report, click **Print Preview**.
2. From the Print Preview window, you can customize the report layout, print the report, or export the report as a file. Select from the following toolbar buttons.

		Click to modify report layout (margins, paper size, orientation) before printing or exporting the report.
		Click to specify advanced data display options. i Note: This option is only available with <i>Predicted Reorganization Benefit</i> , <i>Reorganization History</i> , and <i>DML Profile</i> reports.
		Click to configure report header and footer options. i Note: This option is only available with <i>Predicted Reorganization Benefit</i> , <i>Reorganization History</i> , and <i>DML Profile</i> reports.
		Click to export the report. Click the drop-down arrow to select a file format. Formats include PDF and Excel.
		Click to print the on-screen report without saving it to a file.
		Click to toggle print layout view on and off. i Note: This option is not available with <i>Predicted Reorganization Benefit</i> , <i>Reorganization History</i> , and <i>DML Profile</i> reports.
		Click to add a watermark to the report. i Note: This option is only available with <i>Predicted Reorganization Benefit</i> , <i>Reorganization History</i> , and <i>DML Profile</i> reports.

	Click to change the report background color. i Note: This option is only available with <i>Predicted Reorganization Benefit</i> , <i>Reorganization History</i> , and <i>DML Profile</i> reports.
---	--

Manage Scheduled Tasks

A scheduled task is created when you schedule a Space Manager activity such as report generation or script generation. When you schedule one of these activities, a Windows scheduled task is created. You can view and manage these tasks using Space Manager.








Scheduled tasks that are managed using the Scheduled Tasks window include:

- Scheduled script generation for Live Reorg scripts. See [About Scheduling a Recurring Reorganization](#) on page 122 for more information.
- Scheduled report generation, such as Predicted Benefit Reorg reports.

When a scheduled task is executing, Space Manager runs in the background.

To manage scheduled tasks

1. Select **Manage | Scheduled Tasks**. The Space Manager Scheduled Tasks window opens.
2. Review the list of scheduled tasks. The grid displays the status of each task and the next scheduled run time.
3. Select a task and click one of the following toolbar buttons to perform the action. Or right-click a task and select from the menu.

	Click to modify the selected task's schedule.
	Click to enable selected task.
	Click to disable selected task.
	Click to run selected task.
	Click to run selected task interactively.
	Click to delete selected task.
	Click to refresh.

i Note: You can also manage tasks and view task properties using the Windows Task Scheduler.

Reorg Need Options

Use this page to specify settings for Space Manager to use in Reorg Need calculations for the selected database.

i Note: Specify options for each database connection, separately.

To specify Reorg Need Options

1. In the Explorer window, select the database for which you want to specify options.
2. Select **Tools | Options**.
3. Select **ReorgNeed**.
4. Enter weight and threshold settings. Review the following for additional information:

Option	Description
Wasted Space Weight	Enter a number to be used to weight the Wasted Space factor. Space Manager multiplies the Wasted Space factor by this number when calculating the Reorg Need for an object. <i>default = 70.00</i>
Chained Row Weight	Enter a number to be used to weight the Chained Row factor. Space Manager multiplies the Chained Row factor by this number when calculating the Reorg Need for an object. <i>default = 30.00</i>
Medium ReorgNeed Range	Enter a number to set as the lower threshold for medium reorg need. Reorg need values that are above this number, and at or below the high reorg need threshold, are considered <i>Medium Reorg Need</i> . The color <i>orange</i> is used to identify medium reorg need in the datafile map and in the Reorg Need column in the Explorer. <i>default = 25.00</i>
High ReorgNeed Range	Enter a number above which all reorg need values are considered <i>High Reorg Need</i> . The color <i>red</i> is used to identify high reorg need in the datafile map and in the Reorg Need column in the Explorer. <i>default = 50.00</i>

5. Click **Save** to save your settings.

i **Notes:**

- See [How Reorg Need Is Calculated](#) on page 33 for more information.
- If you modify reorg need threshold settings, press **F5** in the Explorer window to refresh your view.
- The color *green* is used to identify low reorg need.

Repository Schedule Options

Use this page to specify options for the Repository.

i **Note:** Specify options for each database connection, separately.

To specify Repository schedule options

1. In the Explorer window, select the database for which you want to specify options.
2. Select **Tools | Options | Repository Schedule**.
3. Specify the following Repository options:

Choose number of segments updated during each Repository Job Run	Select the number of segments that you want the Repository Job to update. <ul style="list-style-type: none">• Update All Segments—Select to update all segments.• Recently Analyzed—Select to update only recently analyzed segments.• Percentage—Select to update only a percentage of the segments with wasted space. Then specify a percentage. When the Repository job runs, Space Manager begins updating the segments with the oldest wasted space calculations until it reaches the percentage you specify.
--	---

4. Click **Save** to save your settings.

Explorer Object Lists

Use this page to instruct Space Manager how to gather the list of database objects to display in the Explorer.

To specify Explorer Object Lists Options

1. Select **Tools | Options**.
2. Select **Explorer Object Lists**.
3. Select one of the following options:
 - **Use only Space Manager Repository to list objects in Explorer**—Select this option to use only the information collected in the Space Manager Repository when displaying object lists in the Explorer. This option displays object lists quicker, but lists might not be current.
 - i** **Note:** If you select this option, object lists are current to the date and time the Repository was last updated.
 - **Use DBA views to list objects in Explorer**—Select this option to use Data Dictionary views when displaying object lists in the Explorer. This option is slower, but the list of objects is current.

i Note: When using this option, the additional object information presented in the Explorer is gathered from the Repository. This information includes estimates and calculations, such as reorg need.

4. Click **Save**.
5. To refresh the list of objects in the Explorer after modifying this option, return to the Explorer window and press **F5**.

i Note: To set or modify the frequency with which the Repository is updated and to schedule the update time, select **Tools | Admin | Change Repository Update Schedule**.

Reorg Manager Options

Use this page to specify options for the Reorg Manager.

i Note: Specify options for each database connection, separately.

To specify Reorg Manager options

1. Select **Tools | Options**.
2. Select **Reorg Manager**.
3. Specify the following Reorg Manager options:

Option	Description
Always use T-Lock switch	Select this option to force the Reorg Manager to always use the T-Lock switch, even when the Online switch is supported for the table and the Partitioning Option is enabled in the database. i Note: If the Partitioning Option is enabled but not <i>licensed</i> in the database, you can select Always use T-Lock switch to ensure that the Online switch is not used in the Reorg script.

4. Click **Save** to save your settings.

Partitioning Wizard

Partitioning Wizard Defaults

Use this page to specify default values for options used in the Partitioning Wizard. Default settings are saved for the Space Manager client and used for all tables on the Partitioning Wizard's Object list. You can override default settings for individual tables. Most are displayed in the Partitioning Method and Partitioning Parameters pages.


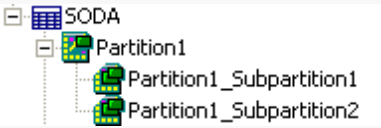
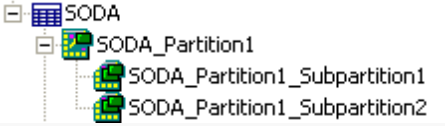
i Note: Specify options for each database connection, separately.

See also:

- [Partitioning Key Options](#)
- [Partitioning Estimates](#)

To specify Partitioning Wizard defaults

1. Select **Tools | Options**.
2. Select **Partitioning Wizard | Defaults**.
3. Specify default values for the Partitioning Wizard options. Review the following for additional information:

<p>Default Locality</p>	<p>Specify a default method to use in the Partitioning Wizard for index partitioning:</p> <ul style="list-style-type: none"> • Make indexes global by default—Select this option to partition indexes independently of their tables. Partitioned global indexes must be prefixed. This means columns in the partitioning key must be in the same order as columns in the index. • Make indexes local by default—Select this option to partition indexes in the same way as their tables. These can be prefixed or nonprefixed. <p>You can override the default in the Partitioning Wizard. See Specify Partitioning Method on page 105 for more information.</p>
<p>Default Partition Names</p>	<p>Use these fields to specify default prefixes for Space Manager to use when creating names for new partitions and subpartitions.</p> <p>Prefix for default partition names—Enter the default name to use for partitions. (P is the default.) A sequential partition number is added to the end of each partition name. If you enter <i>Partition</i> as the default name, new partition names look like this in the wizard's Object list:</p>  <p>Prefix for default subpartition names—Enter the default name to use for subpartitions. (SP is the default.) A sequential subpartition number is added at the end of each name; the parent partition name is added at the beginning. If you enter <i>Subpartition</i> as the default name, new subpartition names look like this in the wizard's Object list:</p>  <p>Include base object name in default partition names—Select this checkbox to include the name of a parent table or index in the names of new partitions and subpartitions.</p>  <p>You can override the default values in the Partitioning Wizard. See Specify Partitioning Method on page 105 for more information.</p>

i Note: Names are stored in uppercase characters if you do not enter them with double quotes. Each name can have up to 30 characters. That maximum is shared by all parts of a name, including numbers and parent names. If all parts add up to more than 30 characters, certain parts are truncated. However, sequential numbers are always included.

4. Click **Save** to save your changes.

Partitioning Key Options

Use this page to specify default values for partitioning key options used in the Partitioning Wizard.

i Note: Specify options for each database connection, separately.

To specify partitioning key options

1. Select **Tools | Options**.
2. Select **Partitioning Wizard | Partitioning Keys**.
3. Specify default values for the partitioning key options. Review the following for additional information:

Automatic Key Values	<p>Selecting these options instructs Space Manager to select them by default in the Partitioning Wizard.</p> <ul style="list-style-type: none">• Automatically set upper bounds for range partitions—Select to instruct the wizard to automatically set the high values to use as the upper boundary for range partitions.• Automatically set list values for list partitions and subpartitions—Select to instruct the wizard to automatically specify the list values to use for list partitions and subpartitions. <p>i Note: Partitioning key values can be set automatically for objects that contain at least one row per partition or subpartition.</p> <p>See Specify Partitioning Parameters on page 108 for more information.</p>
Validate Partitioning Key Values	<p>Select this option if you want the Partitioning Wizard to automatically check partitioning key values against existing data. This ensures that a table and its local indexes contain partitions or subpartitions for all rows when MAXVALUE or DEFAULT is not used as a partitioning value.</p> <p>See Specify Partitioning Parameters on page 108 for more information.</p>

4. Click **Save** to save your changes.

Partitioning Estimates

Use this page to specify default values for options used in the Partitioning Wizard.

i Note: Specify options for each database connection, separately.

To specify partitioning estimates default values

1. Select **Tools | Options**.
2. Select **Partitioning Wizard | Estimates**.

- Specify default values for these partitioning options. Review the following for additional information:

Estimate Partition Sizes	<p>Estimate partition sizes based on row count—Select if you want the Partitioning Wizard to automatically calculate partition sizes based on expected row count.</p> <p>Use the estimate sizes option to automatically size partitions and subpartitions. The Partitioning Wizard calculates segment sizes based on the number of rows they are expected to contain. This ensures that the space allocated to each partition or subpartition is sufficient for its data.</p> <p>i Note: The estimate sizes option should not be used when the option for automatically setting partitioning key values is selected. When both are selected, space allocation is estimated twice.</p> <p>See Specify Partitioning Parameters on page 108 for more information.</p>
Table Scan Options	<p>Table scan options determine whether tables are scanned or sampled when the Partitioning Wizard sets key values or estimates sizes. You can choose to scan entire tables, sample a percentage of table rows, or dynamically determine whether to scan or sample based on table size.</p> <p>A full table scan may take a long time, depending on table size. Sampling a table takes less time and may be the best choice for very large tables. Although all rows are not read, they are evenly sampled so that results accurately represent a table.</p> <p>See Specify Partitioning Parameters on page 108 for more information.</p>

- Click **Save** to save your changes.

Server Agent Options

Use this page to specify options for the Server Agent.

i Note: Specify options for each database connection, separately.

To specify Server Agent options

- Select **Tools | Options**.
- Select **Server Agent**.
- Specify the following Server Agent options:

Option	Description
Automatically check Server Agent status every <i>N</i> minutes	Specify an interval at which Space Manager checks the status of the Server Agent.

- Click **Save** to save your settings.

Miscellaneous Options

Some of these options are for use when troubleshooting an issue with Space Manager. Modify these settings if instructed to by Support.

i Note: Specify options for each database connection, separately.

To specify *Miscellaneous Options*

1. In the Explorer window, select the database for which you want to specify options.
2. Select **Tools | Options | Miscellaneous**.
3. Modify options by entering a new value. Review the following for additional information:

Option	Description
ALWAYS USE SAMPLE_LONG ORA600 WORKAROUND FOR ANALYSIS	<i>default = FALSE</i>
SPC QUERIES HINT MODE	<i>default = LEGACY</i>

4. Click **Save** to save your settings.

Troubleshooting

Troubleshoot Script Execution Errors

Use the following information to help you investigate, troubleshoot, and resolve errors that occur during script execution.

See also:

- [Status Codes for Script Execution](#)
- [QSA Error Messages](#)

General Guidelines

1. **Determine how to proceed based on the reorg type and the reorg stage.**

After you encounter an error, identify the type of reorganization that failed and the stage at which the reorganization process failed. Use this information to determine how to proceed. Use the following table to guide you through this process:

Type of Reorganization	Stage at which script failed	General Guidelines
Live Reorganization	Before the switch	Try to correct the issue and restart the script (see step 2). If you cannot resolve the issue, you can clean up and abandon the job (see step 3).
	Inside the switch	Contact Quest Software Support.
	After the switch	The database is operational but you can call Quest Software Support to report the error. However, this is not an urgent issue because the database is operational.
Standard Reorganization		If the script fails during script execution for a standard reorg, you must fix the error and complete the script. If you do not, you would have tables without indexes, tables with wrong names, and other issues that would prevent you from using the database. Proceed to step 2. The Clean up command is disabled for Standard Reorganizations.

2. **For errors that did NOT occur inside the switch, try to resolve the issue and then restart the script.**

Identify and resolve issue: Review the error message and the script execution logs to identify the cause of the error and then fix the issue. See the following for additional information:

- a. For Oracle errors, see [Common Oracle Errors Found in Script Execution](#) or the Oracle documentation.
- b. For more information about the status code, see [Status Codes for Script Execution](#).
- c. For more information about QSA errors, see [QSA Error Messages](#).
- d. For help finding script logs, see [Where to Find Script Logs](#).

Restart script: After you resolve the problem that caused a reorganization script to fail, you can restart script execution using the **Restart** function in the Script/Job Monitor. See [Restart Scheduled Scripts](#) on page 124 for more information.

3. **If you are unable to resolve the issue or restart the script, clean up the script.**
 - a. To clean up a script, right-click the script in the Script/Job Monitor and select **Clean up**. If the **Clean up** option is disabled, call Quest Software Support.
 - b. After cleaning up the script, you can delete it (or attempt to re-run it if the objects to be reorganized are unchanged).

Common Oracle Errors Found in Script Execution

Review the following Oracle error messages and resolution guidelines.

Error Code	Description / Guidelines
ORA-01651 to ORA-01656 Unable to extend <object>	<p>If you encounter an "ORA-01653 unable to extend table..." or another <i>unable to extend</i> error, there is not enough space for the reorganization. To resolve the issue, add space to accommodate the reorganization.</p> <ol style="list-style-type: none"> 1. Add space where needed. 2. Restart the script. 3. If you are unable to restart the script, use the Clean up command. Then delete the script (or attempt to re-run the script).
ORA-00054 Resource busy and acquire with NOWAIT specified	<p>If you encounter an ORA-00054 error, the script is waiting for one or more tables that are currently locked by another session. You may also see the following Space Manager status message: 610 Blocking.</p> <p>Background: Space Manager must get a very brief exclusive lock on a table during a reorganization and also may perform an ALTER TABLE/INDEX statement which requires a DDL lock. This can lead to locking issues.</p> <p>To resolve this issue, do the following:</p> <ol style="list-style-type: none"> 1. Review the execution logs to determine which table or index is causing the locking issue. 2. If possible, find the session that is locking this table. Then resolve any issues with this session, if necessary, to unlock the table allowing the script to continue. 3. If you are unable to unlock the table, Cancel the script. Then do one of the following. <ul style="list-style-type: none"> • Restart the script at a later time when database activity is light. • Use the Clean up command and then delete the script.

Error Code	Description / Guidelines
	<p>Then recreate and run the script later, when database activity is light.</p> <ul style="list-style-type: none"> • If you cannot restart the script and the Clean up command is disabled, contact Quest Software Support. <p>Wait to Restart vs Clean Up and Recreate Script</p> <p>Wait to restart: While the script is waiting to be restarted, Space Manager is tracking changes to the table. This information is stored in a table which grows as changes are tracked. In addition, the tracking activity adds traffic to the database. The advantage of this method is that you do not need rerun all the reorg steps from the beginning or recreate the script.</p> <p>Clean up and recreate script: If you use the Clean up command, Space Manager stops tracking changes, removes tracking triggers and procedures, and deletes the copy table. This removes any tracking items and traffic that might interfere with database performance. However, if you want to run the script again, you must recreate it.</p>

Where to Find Script Logs

You can investigate script execution errors from several different areas in the Space Manager interface. The area you work from is determined by how the script was executed: interactively in the Editor or as a scheduled script.

Scheduled Script Execution

Use the Script/Job Monitor to identify any errors that occur during scheduled script execution. The monitor displays a list of all scripts that are stored in the Space Manager repository for a database.

As you check for execution errors in the Script/Job Monitor, you can view script execution logs. A script log is generated by QSA for each script it runs. An additional log is generated each time a script is restarted.

- To open script logs, right-click a script in the Detail pane and select **Retrieve Script Logs**.

Script logs contain information that can be used by Quest Software Support to diagnose script problems. This information includes platform version, Oracle version, execution start and stop times, steps performed by a script, and errors. Logs can be e-mailed to Support.

Log generation begins as soon as script generation begins. Script logs are stored along with the QSA log in the Log directory under QSA's installation directory. How long log files are kept is determined by QSA's MAX_LOG_AGE parameter. See [QSA Server Agent Parameters](#) on page 189 for more information.

i Note: The QSA log contains information on all activity performed by QSA. This information includes commands received, processes spawned, and scripts executed. The log for the current date is called QEXECD.LOG. The names of logs for previous dates include the date a log was generated (for example, QEXECD-01-01-2004.LOG).

Interactive Execution (in the Editor)

Interactive execution of reorganization scripts is performed from the SQL Editor. The status of each script statement is displayed in the status bar at the bottom of this window. If an error occurs, an error message is displayed in the status bar.

An execution log for all script statements is displayed in the Spool tab of the SQL Editor.

Status Codes for Script Execution

This section provides a list of status codes for scheduled server-side script execution. These are displayed in the Status Message column of the Script/Job Monitor.

Codes for Script Groups

Code	Message
0	New
10	Unscheduled Displayed for script groups that are stored, but not scheduled.
50	Completed Displayed for script groups that have executed successfully.
100	Scheduled Displayed for script groups that are scheduled and waiting to run.
200	Active
300	Interrupted

Codes for Unscheduled Scripts

Code	Message
110	Unscheduled Displayed for scripts that are stored, but not scheduled.

Codes for Completed Scripts

Code	Message
210	Completed Displayed for scripts that have executed successfully.
220	Completed with Warnings Displayed for scripts that have executed with one or more warnings (but no errors).
230	Completed with Errors Displayed for scripts that have executed, but with one or more errors (and possibly warnings).

Codes for Scripts that Are No Longer Running

Code	Message
310	<p>Canceled by request</p> <p>Displayed for executing scripts that have been canceled by the user.</p>
320	<p>Aborted</p> <p>Displayed for scripts that have been aborted due to an error. An error message is also displayed.</p>
324	<p>Aborted no restart</p> <p>Displayed for live reorganization and partitioning scripts that cannot be restarted. This status is displayed after an attempt to restart a script is made. Usually, it indicates that a QUEST_EXEC command is not scripted correctly. It can also indicate that a LiveReorg object or control structure needed by QSA is missing. For help in cleaning up LiveReorg objects, contact Quest Software Support. Once cleanup is performed, a new script can be generated for objects in the failed script.</p>
325	<p>Aborted restart</p> <p>Displayed for live reorganization and partitioning scripts that cannot be restarted. This status is displayed after an attempt to restart a script is made. Usually, it indicates that a LiveReorg object or control structure needed by the QSA is missing. For help in cleaning up LiveReorg objects, contact Quest Software Support. Once cleanup is performed, a new script can be generated for objects in the failed script.</p>
350	<p>Aborted internal</p> <p>Displayed for scripts that failed due to a fatal internal error. Information on the error is displayed in the script execution log.</p>
360	<p>Aborted timeslot</p> <p>Displayed for scripts that did not start within 10 minutes of their scheduled time. There is a 10-minute time window for the start of script execution. If QSA is running or started during this time window, a script is executed by QSA. If QSA is started after the time window ends, a script is aborted. Reschedule the script or run it using the Execute function available in the Scripts/Job Monitor. Check the monitor to see QSA's status.</p>

Codes for Halted and Scheduled Scripts

Code	Message
370	<p>Halted for datafile maintenance</p> <p>Displayed by default for scripts that have been paused for datafile maintenance or another purpose. Your own reason for a pause is displayed after "Halted for..." if you entered one when you selected the script-pause option. This option was available for standard reorganizations with Reorg Manager in previous releases (8.0.2 or earlier) of Space Manager.</p>
410	<p>Scheduled</p> <p>Displayed for scripts that are scheduled and waiting to run.</p>

Codes for Executing Scripts

Code	Message
510	Executing Displayed for scripts that are currently executing.
515	Processing with posting paused

Codes for LiveReorg Scripts

Code	Message
520	Posting live modifications Displayed for live reorganization scripts for which live updates made to the original tables are being posted to reorganized copy tables.
610	Blocked Displayed for live reorganization scripts that are waiting to obtain a lock on one or more tables that are locked by another session. Tables are locked briefly at the beginning of the data copy stage of a live reorganization. They are also locked briefly during the switch from reorganized table to copy table when the T-Lock switch style is used.
620	Waiting until Day/Date/Time Displayed for live reorganization scripts that are waiting for a user-specified time window to make the switch from original tables to reorganized copy tables.
630	Waiting for approval Displayed for live reorganization scripts that are waiting for the user to approve the switch from an original table to a reorganized copy table.

QSA Error Messages

This section provides a list of the error messages generated by QSA. The description for each error includes cause and recommended action. Most QSA error messages are displayed in the Status Message column of the Scripts/Job Monitor.

Oracle errors are also displayed in the Scripts/Job Monitor. For more information on Oracle errors, see the Oracle documentation and [Common Oracle Errors Found in Script Execution](#).

- i Note** Most QSA error messages have a prefix of "QSA". The error messages QSA generates for FastCopy have a prefix of "FC".

QSA Server Agent Errors (20300 – 20359)

Error	Cause	Action
QSA-20301 Server not available / No response from server	QSA is not installed or not running. This error also occurs if you try to stop QSA when you are not connected to the instance on which it is installed in an active-active OPS or RAC environment.	Install QSA again or start it. See Start QSA Using the Admin Function on page 185 for more information. If QSA still doesn't start, examine the QEXECD.LOG file located in \$INSTALL_DIR/log (where INSTALL_DIR is the installation directory for QSA). If you are trying to stop QSA in an active-active OPS or RAC system, first connect to the instance on which it is installed.
QSA-20302 No such script	An attempt was made to schedule a script that does not exist in the QUEST_SCRIPT table.	Make sure the correct folder name, owner name, and script_id are specified for the script when you store it.
QSA-20303 Script already running	An attempt was made to unschedule a script currently being executed.	Consider canceling the script. This will stop execution. You can then unschedule or reschedule the script.
QSA-20304 Script not running	An attempt was made to cancel a script not currently being executed. Only running scripts can be canceled.	Check the status of the script and cancel it when its status is displayed as Executing.
QSA-20305 Script not waiting	An attempt was made to approve a script that is not waiting for approval for the switch from original table to reorganized copy table.	Continue checking the status of the script and give approval when its status is displayed as Waiting for approval or Waiting until day/date/time.
QSA-20306 Script is not scheduled	An attempt was made to unschedule a script that is not scheduled for execution.	Check the status of the script. No action is necessary.
QSA-20307 Function not implemented	An attempt was made to use a QSA function that has not been implemented.	Contact Quest Software Support.
QSA-20308 No such log file	An attempt was made to retrieve a log file that does not exist. This can occur when log files are purged due to aging.	Try the operation again. If you get repeated failure, contact Quest Software Support.
QSA-20309 Owner, folder and script cannot contain certain characters	The owner name, folder name, or script_id specified for a script contains a period, quote, forward slash, or backward slash. These characters are not allowed.	Specify an owner, folder, or script_id without a period, quote, forward slash, or backward slash.
QSA-20311 Table conflict with script already executing	This error can occur when: A script includes a table that is also included in a script that is currently executing. A script includes a table with referential integrity on tables in a script that is currently executing. BEQUEATH_DETACH is set to No in	Wait until the currently executing script has completed before executing a conflicting script or set BEQUEATH_DETACH to Yes in the SQLNET.ORA file.

Error	Cause	Action
	the SQLNET.ORA file.	
QSA-20312 Scheduled launch time expired	A script was not executed when its scheduled time occurred, most likely because the server host or QSA was not available at that time. (QSA prohibits script execution at a time other than the one scheduled. This prevents unintended execution.)	Reschedule the script.
QSA-20313 Invalid login name for daemon	The username or password for the Oracle account that installed QSA was changed after installation. QSA uses this account to log in to the database. When the password is changed, QSA can no longer log in. The error is displayed in the QSA log file after username or password is changed and QSA is restarted.	Uninstall QSA and then reinstall it using the new credentials for the Oracle account.
QSA-20314 Active jobs in progress	An attempt was made to stop QSA while one or more scripts were being executed.	Wait until all scripts have completed or cancel all scripts that are being executed. Then stop QSA.
QSA-20315 Daemon running	An attempt was made to start QSA while it was already running. There should only be one QSA daemon or service running per database instance.	If the current QSA daemon or service is unusable, contact Quest Software Support.
QSA-20316 Couldn't start the daemon	QSA could not be started.	Check the QEXECD.LOG file to determine why QSA could not be started. Fix the problem, then try to start QSA again.
QSA-20317 Attempt to schedule script in the past	An attempt was made to schedule a script for a date and time in the past. Scripts cannot be scheduled to start in the past.	Reschedule the script for a date and time that occur after the current date and time.
QSA-20318 Script is already scheduled	An attempt was made to schedule a script, which is already scheduled.	Check the status of the script.

QSA Server Agent Installation Errors (20360 – 20389)

Error	Cause	Action
QSA-20360 Oracle error during installer SQL	An Oracle error occurred during processing of one of the SQL scripts used to install, upgrade, or uninstall QSA. The cause may be conditions in the database or problems with the script. The Oracle error displayed along the QSA error helps identify the cause.	If the cause is lack of database resources, correct the problem and then try to install, upgrade, or uninstall QSA again. If the cause appears to be problems with the script, contact Quest Software Support.

Error	Cause	Action
QSA-20361 Can't parse QSAVersion.txt file	The install script for QSA could not correctly parse the QSAVersion.txt file.	Contact Quest Software Support.
QSA-20362 Oracle version not supported	An attempt was made to install, upgrade, or uninstall QSA against an Oracle database version that is not supported.	If the Oracle version is one that is supported, contact Quest Software Support.
QSA-20363 Product not installed	An attempt was made to uninstall QSA when it is not installed.	If QSA is installed, contact Quest Software Support.
QSA-20364 Can only upgrade or uninstall as installing UNIX user	An attempt was made to upgrade or uninstall QSA as a UNIX user who is not the installing UNIX user or root.	Upgrade or uninstall QSA as the user who installed it or as root.
QSA-20365 Can only upgrade or uninstall as installing Oracle user	An attempt was made to upgrade or uninstall QSA as an Oracle user who is not the installing Oracle user.	To identify the installing Oracle user, check the owner of the QUEST_EXEC_PARAMETER table. Then upgrade or uninstall QSA as that user.
QSA-20370 Required files are missing	Required files such as text, script, and shell script files were not found on the Space Manager client PC from which installation of QSA is being performed.	Reinstall the Space Manager client.
QSA-20371 This user does not have DBA status	The Oracle account used for installing QSA does not have DBA privileges. This account used must be a DBA account with ALTER, CREATE, DROP, and EXECUTE privileges on all objects in the database QSA is being installed for. It must also have the UNLIMITED TABLESPACE system privilege.	Log in as a DBA user with the privileges listed above.
QSA-20372 Required components not found	The Quest common objects are not found on the database server.	Reinstall Space Manager server components.
QSA-20373 Required Quest Script version not found	The Quest Script package is not installed on the database server or is an earlier version than the one required by QSA.	Reinstall Space Manager server components and QSA.
QSA-20374 Cannot stop the daemon	An attempt to stop QSA failed because a live reorganization script is being executed.	Wait until the script completes. Then try to stop QSA.

Error	Cause	Action
QSA-20375 Cannot start the daemon	An attempt to start QSA failed during installation of QSA.	Cancel the installation and check with a system administrator to determine if there are system problems. Once any problems are resolved, try to install QSA again.
QSA-20376 Login ID not valid	The login name of the UNIX account used for installing QSA is not valid.	Enter a valid UNIX login name.
QSA-20377 This Operating System is not supported	An attempt was made to install QSA on an operating system that is not supported. HP-UX, IBM AIX, Sun Solaris, and Compaq Tru64 UNIX, Windows NT, and Windows 2000 are the platforms are supported.	Cancel the installation.
QSA-20378 Root configuration failed	An attempt by root to install QSA failed.	Contact Quest Software Support.
QSA-20379 Root cleanup failed	An attempt by root to uninstall QSA failed.	Contact Quest Software Support.
QSA-20380 Path invocation error	The installation script for QSA was not invoked as ./SCRIPT_NAME. It must be invoked as ./SCRIPT_NAME.	If this error occurs more than once, contact Quest Software Software Support.
QSA-20381 Invalid arguments	An attempt was made to start the QSA installation script with incorrect arguments.	If this error occurs more than once, contact Quest Software Support.
QSA-20382 Unknown architecture type	An attempt was made to install QSA on an operating system that is not supported. HP-UX, IBM AIX, Sun Solaris, and Compaq Tru64 UNIX, Windows NT, and Windows 2000 are the platforms are supported.	If this error occurs more than once, contact Quest Software Support.
QSA-20383 Could not create configuration file	The QEXECD.CONF file could not be created. This file stores configuration information for QSA. The problem could be due to lack of disk space or permission.	If this error occurs more than once, contact Quest Software Support.
QSA-20384 Must be root	An attempt was made to invoke the ROOT.SH installation script or ROOT_UNINSTALL.SH uninstallation script by an account other than root. These scripts must only be invoked by the root user. This is because they require root privileges to install files into the /ETC and /SBIN directories.	Log in as root (or su) and invoke the script again.

Error	Cause	Action
QSA-20385 Config file doesn't exist	The QEXECD.CONF file could not be found. This file stores configuration information for QSA.	If this error occurs more than once, contact Quest Software Support.
QSA-20386 Invalid ORACLE_HOME	An invalid ORACLE_HOME was specified during installation of QSA. This is the default base directory for QSA. When an invalid directory is specified, QSA cannot find the LIBCLNTSH library with which it is dynamically linked. The LIBCLNTSH library is supplied by Oracle and should reside in \$ORACLE_HOME/lib.	Check the value of ORACLE_HOME. If installation is being performed using the installation wizard for QSA, check the value of ORACLE_HOME in your /ETC/ORATAB or /VAR/OPT/ORACLE/ORATAB file.
QSA-20387 Do not invoke this script	An attempt was made to execute a shell script that is intended for internal use by QSA.	Contact Quest Software Support.

Miscellaneous Errors (20390 – 20399)

Error	Cause	Action
QSA-20390 An underlying OS error occurred	An operating system error occurred. The cause may be a temporary lack of resources on the server host.	Execute the script again. If the problem occurs more than once, contact Quest Software Support.
QSA-20390 Can't copy <src_file> to <dest_file>	During manual installation of QSA on UNIX, the install script was not able to copy a file to the right location.	Make sure you are logged in as the user who initially installed QSA. Make sure the installing user has permission to write to all directories in the installation path. Check to see if the disk is full.
QSA-20391 ORA - <nnnnn> during internal SQL	An Oracle error occurred while internal SQL was being processed. The cause can fall into either of two categories. These are conditions in the database and problems experienced by QSA. The Oracle error number displayed in the message helps identify the cause of the error.	Check the script execution log generated by QSA to see a description of the Oracle error. Then take action based on information in the description.
QSA-20391 Cannot identify Oracle version	The version of QSA being installed is not compatible with the version of Oracle installed.	Contact Quest Software Support.
QSA-20392 Internal error in script engine	A function based index is the most selective index available for a table and was chosen by Space Manager as the posting index during a live reorganization. However, function based indexes cannot be used as posting indexes.	Reorganize the table using standard reorganization. If you want to perform a live reorganization for the table, give it a more selective index than the function based index.

Error	Cause	Action
QSA-20392 Internal error: Invalid install type	During a manual installation on UNIX, the install script could not determine whether the install option requested was Install, Update, or Uninstall.	Contact Quest Software Support.
QSA-20392 Internal error (ORACLE_SID not defined)	The ROOT.SH install script could not determine the ORACLE_SID from the QEXECD.CONF file.	Contact Quest Software Support.
QSA-20393 Must be run under QSA Server Agent	An attempt was made to run a script from the client or other environment when it should be run by QSA. Live reorganization scripts and scripts that use FastCopy or parallel QSA processes must be run by QSA.	Store and schedule the script for execution by QSA. You can work from the Script/Job Monitor to do this.
QSA-20394 Could not create temporary QSA object in <LW_ TABLESPACE>	The tablespace used for LiveReorg objects does not have enough free space. The tablespace is specified with QSA's LW_TABLESPACE parameter. QSA creates and uses LiveReorg objects during a live reorganization and drops them after the switch.	Provide the space needed by increasing datafile size for the tablespace, adding a datafile to the tablespace, or setting the Autoextend property to On. All can be done from Tablespace Properties. An alternative action is to specify a tablespace with more free space for the LW_TABLESPACE parameter. This is done from the QSA Installer. After taking any of these actions, restart the script from the Script/Job Monitor.
QSA-20395 Could not parse script	A character in an object or owner name could not be converted to UNICODE. During script execution in databases that use multibyte character sets, QSA converts QUEST_EXEC commands to UNICODE. Characters in names are also converted. They are converted back to their database character set before reorganization is complete. There are a few characters that QSA cannot convert to UNICODE. After a script fails due to a conversion problem, the character that could not be converted is displayed in the script execution log or in QSA's log (QEXECD.LOG).	Contact Quest Software Support.

Script Syntax Errors (20400 – 20499)

Error	Cause	Action
QSA-20400 WHENEVER without WARNING or ERROR keyword	The syntax of a WHENEVER command is incorrect. The WHENEVER command must specify either SQLERROR, ERROR, SQLWARNING, or WARNING.	If the script was automatically generated, contact Quest Software Support.
QSA-20401 BEGIN TASK occurred outside of parallel block	A BEGIN TASK command occurred outside of a parallel block. A task block is delimited by BEGIN_TASK and END_TASK. A task block can only occur inside a parallel block.	If the script was automatically generated, contact Quest Software Support.
QSA-20402 END TASK without BEGIN TASK	An END TASK command occurred outside a parallel block. Along with BEGIN_TASK, this command defines a task block. Task blocks can only occur inside a parallel block (delimited by BEGIN_PARALLEL and END_PARALLEL). They contain a group of SQL statements to be executed in parallel.	If the script was automatically generated, contact Quest Software Support.
QSA-20403 END PARALLEL without BEGIN PARALLEL	An END_PARALLEL command occurred without a corresponding BEGIN_PARALLEL command. Together, the two commands define a parallel block, used to support parallel script execution.	If the script was automatically generated, contact Quest Software Support.
QSA-20404 Too many tasks	The number of tasks within a parallel block exceeds the maximum allowed. The limit is 400 tasks.	Modify the script so that there are no more than 400 tasks within a parallel block. If you require more than 400 tasks, contact Quest Software Support.
QSA-20405 Parallel factor is out of range	The number of parallel operations specified is outside the range allowed. The range is from 1 to 20 inclusive.	Modify script options so that the number of parallel operations specified is from 1 to 20. Then regenerate the script. If you require more than 20 parallel operations, contact Quest Software Support.
QSA-20406 Internal error in task engine	This error should not occur under normal circumstances.	Contact Quest Software Support.
QSA-20407 END_COPY command without BEGIN_COPY	The END_COPY command occurred without the BEGIN_COPY command.	If the script was automatically generated, contact Quest Software Support.
QSA-20408 Invalid parameter name	The SET_PARAMETER() function was called with an invalid parameter name.	Correct the script with a valid parameter name.

Error	Cause	Action
QSA-20409 Invalid parameter value	The SET_PARAMETER() function was called with an invalid value for a parameter.	Correct the script with a valid parameter value.
QSA-20411 Statement numbering is incorrect	The script has been stored improperly.	Generate and store a replacement script (with the same options, if possible) under a different name. If the new script results in the same error, contact Quest Software Support.
QSA-20412 Script terminates inside task block	QSA detected that a script ended prematurely inside a BEGIN_TASK/END_TASK block.	Generate and store a replacement script (with the same options, if possible) under a different name. If the new script results in the same error, contact Quest Software Support.
QSA-20413 Script terminates inside parallel block	QSA detected that a script ended prematurely inside a BEGIN_PARALLEL / END_PARALLEL block.	Generate and store a replacement script (with the same options if possible) under a different name. If the new script results in the same error, contact Quest Software Support.

Script Execution Errors (20500 – 20599)

Error	Cause	Action
QSA-20500 Script/Engine version incompatible	An attempt was made to run a script that is not compatible with the version of QSA currently installed. This error occurs for scripts that were generated for a version of QSA that is not compatible with the current version.	Regenerate the script.
QSA-20501 Approval type not recognized	An invalid approval type was specified for the switch from original table to reorganized table.	Contact Quest Software Support.
QSA-20502 Target completion date/time expired	A script did not reach the COMPLETE_POSTING stage of live reorganization within the time window specified for the switch from original table to reorganized table.	Reschedule the script and adjust times for the window, if necessary. Allow enough time between the start of execution and the end of the window for the table to be reorganized, for indexes to be created, and for all live updates to be posted to the reorganized table. These tasks could take several hours for a large table.
QSA-20504 Table contains large data type without a primary key or non-nullable unique key	This error may occur during live reorganizations of tables that do not have a PRIMARY KEY or non-nullable UNIQUE index, but contain a BFILE or NESTED TABLE datatype. A PRIMARY KEY or unique index is required as the posting index. The error should not	Contact Quest Software Support.

Error	Cause	Action
	<p>occur for tables with a LONG, LONG RAW, CLOB, BLOB, REF, VARRAY, OBJECT, or XMLTYPE datatype. All of these datatypes are supported and can be reorganized with a ROWID posting index or the “all columns” posting method. See Posting Indexes on page 214 for more information.</p>	
<p>QSA-20506 Malformed script</p>	<p>This error occurs when a live reorganization script is missing an LW_START_REORG command, a BEGIN_COPY command, or an END_COPY command. Missing commands can result from problems in script generation or script edits by users. This error also occurs when you try to run a live reorganization script generated with a pre-5.6 version of Space Manager.</p>	<p>Contact Quest Software Support.</p>
<p>QSA-20522 Trigger lock is not possible</p>	<p>QSA could not place a trigger lock on a table undergoing a live reorganization. The problem can occur when a table has multiple triggers for the same event or when there is a conflict with other activity against the database.</p>	<p>Check the table definition for multiple triggers for the same event. If these do not exist, re-run the reorganization script. If the problem occurs again, contact Quest Software Support.</p>
<p>QSA-20524 Cannot reorg while snapshot replication is enabled</p>	<p>A live reorganization should not be performed for a table which is being replicated via snapshot replication.</p>	<p>If you want to perform a live reorganization for a table with snapshot replication, please contact Quest Software Support.</p>
<p>QSA-20525 Cannot reorg table with security policies enabled</p>	<p>Security policies interfere with the T-locks (trigger locks) used during the switch stage of a live reorganization and can prevent the switch from original table to reorganized copy table from completing.</p>	<p>If you want to perform a live reorganization for a table with security policies, please contact Quest Software Support.</p>
<p>QSA-20527 Possibly unsafe to reorg this table</p>	<p>An attempt was made to reorganize a table owned by SYS or SYSTEM, or a table whose name starts with QUEST_. Generally, these tables do not need to be reorganized. Reorganizing them might result in unpredictable behavior by Oracle or QSA.</p>	<p>Determine whether the table truly needs to be reorganized. If it does, contact Quest Software Support for assistance in completing the reorganization.</p>
<p>QSA-20531 Invalid script, no prior switch</p>	<p>A SWITCH_COMPLETE command is missing from the live reorganization script.</p>	<p>Regenerate the live reorganization script and run it. Do not delete any SWITCH_COMPLETE commands. If the problem occurs again, contact Quest Software Support.</p>
<p>QSA-20532 Attempt to restart script failed</p>	<p>An attempt to restart a script failed. Typically, this happens at the beginning of a restart. It can occur for any of the following reasons: a LiveReorg object is</p>	<p>Contact Quest Software Support.</p>

Error	Cause	Action
	missing; QSA cannot find the script; the restart point is not valid; there is an inconsistency in the persistent storage data used by QSA; an object ID is different from what is expected. (An ID can be different if an object was reorganized with another script while the failed script was waiting for restart.)	

FastCopy Errors (20600 – 20699)

Error	Cause	Action
FC-20601 An underlying Oracle error occurred	An Oracle error occurred.	Review Oracle's error message for the specific cause. Then take Oracle's recommended action and restart the reorganization script using the Restart function.
FC-20602 An underlying SQL*Loader error occurred	A SQL*Loader error occurred.	Review SQL*Loader's error message for the specific cause. Then take Oracle's recommended action and restart the reorganization script using the Restart function.
FC-20603 An underlying OS error occurred	An operating system error occurred.	Review the operating system error message. Then take the recommended action and restart the reorganization script using the Restart function. If no corrective action is recommended, write down the complete error message and contact Quest Software Support.
FC-20605 Invalid function arguments	Invalid function arguments were used.	Contact Quest Software Support.
FC-20606 Out of memory	The operating system is out of memory.	Reduce the number of running processes, increase operating system swap space, and/or increase ULIMIT resources. Then restart the script using the Restart function.
FC-20607 Invalid parameter identifier	An invalid identifier was used for a FastCopy parameter.	Contact Quest Software Support.
FC-20608 Invalid parameter value	An invalid value was entered for a FastCopy parameter.	Contact Quest Software Support.
FC-20609 Functionality not yet implemented	An attempt was made to use a QSA Server Agent function that has not been implemented.	Contact Quest Software Support.

Error	Cause	Action
FC-20611 Write failure on stream	The file system is full or a communication failure occurred during a data move.	Correct the file system error and/or assign additional OVERFLOW directories. Then restart the reorganization script using the Restart function. If the error persists, contact Quest Software Support.
FC-20612 File not found	The export file specified for reading during import of data was not found.	Contact Quest Software Support.
FC-20613 Disk space is low	The limit on the disk space reserved for FastCopy files in WORKDIR and OVERFLOW has been reached.	Assign additional OVERFLOW directories or decrease the value for the DISKSPACE_RESERVE parameter. The directories and DISKSPACE_RESERVE are specified with QSA Server Agent parameters. Restart the script using the Restart function. If the error persists, contact Quest Software Support.
FC-20614 No export workdir found	No WORKDIR directory was found.	Specify a directory for QSA's WORKDIR parameter.
FC-20615 Unexpected end of file	An unexpected end of file was detected while reading an export file.	Contact Quest Software Support.
FC-20616 Can't link direct export function	The direct export function was not found. This error is usually due to an installation problem or a conflict in Oracle version support. For example, this error will occur if the Oracle DLL in the ORACLE_HOME/bin directory is not the DLL for the version of Oracle currently installed. This directory may contain the wrong DLL if Oracle has been upgraded. The DLL is not automatically updated during upgrades to some versions of Oracle.	Reinstall your application or contact Quest Software Support. If the ORACLE_HOME/bin directory contains the wrong Oracle DLL, copy the right DLL to that directory for the version of Oracle you are running FastCopy against.
FC-20617 Table descriptions are incompatible	Column count or column attributes do not match for the original table and copy table.	Make sure the table definitions are identical and retry.
FC-20618 Table checksum comparison don't match	Checksums do not match for the original table and copy table.	Contact Quest Software Support.
FC-20619 Destination table must be empty	The destination table is not empty. It must be empty for table insert mode.	Contact Quest Software Support.

Error	Cause	Action
FC-20622 No table objects found	The tables specified for export were not found in the data dictionary. If partitions or subpartitions were specified, they have been found but contain no data.	Make sure the owner and table names specified are correct. Contact Quest Software Support.
FC-20623 Direct load row buffer overflow	The direct load import row buffer exceeds the maximum size allowed.	Set QSA's IMPORT_METHOD parameter to CONVENTIONAL. Then restart the reorganization script using the Restart function.
FC-20624 Direct load data conversion error	A conversion error occurred during a direct load import.	Contact Quest Software Support.
FC-20625 Direct load rows loaded different from direct export rows processed	The SQL*Loader dropped row data.	Set QSA's IMPORT_METHOD parameter to CONVENTIONAL. Then restart the reorganization script using the Restart function.
FC-20626 Direct load log file parsing error	An error occurred during parsing of the SQL*Loader log file. Either the log file was not found or the log file is incomplete.	Set QSA's IMPORT_METHOD parameter to CONVENTIONAL. Then restart the reorganization script using the Restart function.
FC-20627 Direct load SQL*Loader not running	The SQL*Loader terminated prematurely.	Set QSA's IMPORT_METHOD parameter to CONVENTIONAL. Then restart the reorganization script using the Restart function.
FC-20628 SQL*Loader terminated abnormally	The SQL*Loader terminated with an error.	Set QSA's IMPORT_METHOD parameter to CONVENTIONAL. Then restart the reorganization script using the Restart function.
FC-20629 Direct load unsupported data type	A direct load was attempted for an unsupported datatype.	Set QSA's IMPORT_METHOD parameter to CONVENTIONAL. Then restart the reorganization script using the Restart function.
FC-20630 Direct load all null columns detected	Rows containing all null columns were detected. SQL*Loader cannot insert these rows.	Set QSA's IMPORT_METHOD parameter to CONVENTIONAL. Then restart the reorganization script using the Restart function.
FC-20632 Column 'NAME' datatype is not supported	The current version of FastCopy does not support user-defined datatypes or the CLOB, BLOB, BFILE, CFILE, and MSLABEL datatypes.	Do not use FastCopy for unsupported types.
FC-20633 Direct path export fail	An Oracle bug causes direct path export to fail.	Contact Quest Software Support.
FC-20636 Empty SQL select list	An internal error occurred.	Contact Quest Software Support.

Error	Cause	Action
FC-20642 FastCopy export file exists	FastCopy has encountered an existing export file and will not overwrite it. The file may contain exported data from a table that was dropped during a previous reorganization that failed.	Contact Quest Software Support.

Administration Functions

License Key

After you first install the Space Manager client and try to connect to a database, you are prompted to enter a license key. Once you enter a permanent key, you do not need to enter it again. It is preserved during upgrades. You only need to change or re-enter a license key when:

- Your trial key expires.
- You switch from a trial key to a permanent key.
- You change your licensing agreement with Quest Software.
- You uninstall and then reinstall the Space Manager client.

To update a license key

1. Select **Tools | Admin | License Key**.
2. Enter your permanent or trial license key.

Install or Upgrade Server Objects

Install server components after installing the Space Manager client. Install server components for each database you want to manage with Space Manager.

Upgrade the server components after upgrading the Space Manager client.

Use the same DBA account to install server components and the Server Agent. See [Space Manager User Requirements](#) on page 16 for more information.

To install (or upgrade) server objects

1. Launch the Space Manager client.
2. When the Connect dialog displays, connect to the database using the Oracle DBA account chosen to install the server components and the Server Agent.
 - i Note:** For more information about creating a connection, see [Connect to Databases](#).
3. (Upgrade only) If this is an upgrade, one or more of the following messages display:
 - **Upgrade required**—This is the default message. Click **Yes**.
 - **Running scripts**—The install program has found scripts that the Server Agent is executing. Click **OK** to cancel the upgrade and perform it later.
 - **Scheduled scripts**—The install program has found scripts that are scheduled for execution by the Server Agent. If you are sure the upgrade will complete before the first script executes, click **Yes**. Otherwise, click **No** to cancel the upgrade. You will need to upgrade later.

- When the Create Space Manager with LiveReorg Objects dialog opens, specify an owner and tablespace for the components. See the following for more information.

Field	Description
Owner	If this is a first-time installation, specify the installation DBA. By default, the field displays the connecting user. To use a different DBA, enter the user name in the Owner field. After you click OK , a password prompt displays.
Tablespace	Select a tablespace for Space Manager objects. The tablespace must have a standard name and free space equal to or greater than the amount for Expected Allocation. For an upgrade, it must also have enough free space for copies of existing Space Manager tables (created for recovery purposes and dropped when an upgrade is complete). Objects are upgraded in their current tablespace. If you select a different tablespace, objects are upgraded in that tablespace and dropped from the current tablespace.
Recommended (upgrade only)	Verify the recommended space allocation for Space Manager objects. The recommendations allow for one year of growth in statistics tables. <ul style="list-style-type: none"> • Small is for databases with less than 1,000 non-SYS segments • Medium is for databases with 1,000 to 9,999 non-SYS segments • Large is for databases with 10,000 to 24,999 non-SYS segments • Huge is for databases with more than 25,000 non-SYS segments <p>i Note: Use the following guidelines for determining space allocation: small for up to 100 GB, medium for up to 500 GB, etc.</p>
Allocation to use	Select a space allocation for Space Manager objects. The recommended allocation is entered by default.
Expected allocation	Verify the expected space allocation for Space Manager objects. This corresponds to the Allocation to use value.
Install Space Manager Demonstration Data (install only)	If you want to install demonstration data after server components are installed, select this checkbox. Demonstration data requires 20 MB or 24 MB of disk space, depending on your database block size. i Note: If the Recycle Bin for an Oracle 10g database contains many objects, the Create Space Manager with LiveReorg Objects window might require a long time to display. To avoid this problem, empty the recycle bin before you begin the installation.

- Click **OK**. The installation progress displays in the Executing Create dialog. To cancel installation, click **Cancel**.

i Note: In some cases, Space Manager may try to allocate more than the expected allocation. It does this when a tablespace has a large minimum extent size or is locally managed and uses uniform extent allocation. In either case, installation may fail. If it does, click **OK** to clear the installation error. Begin installation again and select a larger tablespace for Space Manager objects.
- You will encounter a prompt if QUEST_COM_PRODUCTS_% tables are already installed for the target database but the owner is not the installation DBA for Space Manager.
 - To continue, click **Yes**. The Space Manager objects will be owned by the installation DBA. And QUEST_COM_PRODUCTS_% tables will continue to be owned by their current owner. (Public synonyms and certain views will be owned by SYS.)
 - To cancel installation, click **No**. Then, when you start the installation again, use the QUEST_COM_PRODUCTS_% table owner as the installation DBA. Space Manager objects and QUEST_

COM_PRODUCTS_% tables will be owned by this user. (Public synonyms and certain views will be owned by SYS.)

7. If O7_DICTIONARY_ACCESSIBILITY is FALSE for the target database and the installation DBA does not have EXECUTE on SYS.DBMS_SYS_SQL, you are allowed to run scripts to grant the privilege.
 - When prompted, enter the user name and password for a SYSDBA account. The first script prompts you to enter the name of the installation DBA.
8. When server component installation is finished, one of the following displays:
 - A “success” message displays if you are not installing demonstration data. Click **OK**. Then the Server Agent Installer displays so that you can install the agent. See [Install or Upgrade Server Agent on UNIX or Linux](#) on page 196 for more information.
 - The Install/Refresh Demonstration Data window displays if you are installing demonstration data. See the next step.
9. To install demonstration data, click **Install** in the Install/Refresh Demonstration Data window. It is recommended that you use the default demonstration user name, tablespace names, and datafile names. A password is not required. If you specify different directories for demonstration datafiles, the directories must already exist.
10. When demonstration data installation is finished, two “success” messages display. Click **OK**. The Server Agent Installer displays so that you can install (or upgrade) QSA. See [Install or Upgrade Server Agent on UNIX or Linux](#) on page 196 for more information.

i Note: During installation of server components in Oracle 9.2.0.7, if you encounter ORA-600, “ORA-01001. Invalid cursor” or “ORA-01921” Role name ‘QUEST_SPC_APP_PRIV’ conflicts with another role or user name,” contact Quest Software Support for a workaround or use the solution provided in Oracle note 339070.1.

Reinstall Server Objects

The **Reinstall Space Manager Server Objects** function can be used to install or upgrade Space Manager server components for a database whenever needed. This function is designed for use when you need to rebuild existing components. It is also designed for use after you upgrade to a new version of Oracle. Reinstalling server components updates them for the new version of Oracle.

i Note: The **Reinstall Space Manager Server Objects** function updates server components without affecting data stored in the Space Manager repository. All statistics and stored scripts are preserved.

To reinstall server objects

1. Select **Tools | Admin | Reinstall Space Manager Server Objects**. As installation of server components proceeds, its progress and status are displayed in the SQL Exec dialog. When installation of server components is finished, a completion message displays.
2. When component installation is finished, the Server Agent Installer displays so that you can update the Server Agent.

Uninstall (Remove) Server Objects

The **Remove Space Manager Server Objects** function can be used to uninstall server components from a database whenever needed. This function drops all Space Manager objects from a database, including statistics tables and script tables. It also drops the demonstration data.

i Note: Datafiles for demonstration data tablespaces must be removed from the server manually.

Server components must be uninstalled by the DBA that installed them. They must be removed individually from each database in which they are installed.

Normally, demonstration data is automatically removed along with server components. However, the data is not removed when demonstration tablespaces contain objects that are not owned by the demonstration user. To ensure that demonstration data is removed, relocate non-demonstration objects before you uninstall the server components.

After demonstration data is uninstalled, manually delete the demonstration datafiles, SPCDEMO01_nn.dbf and SPCDEMO02_nn.dbf.

To uninstall (remove) server objects and demonstration data

1. Run the Space Manager client and connect to the target database as the DBA that installed server components.
2. To uninstall server components, select **Tools | Admin | Remove Space Manager Server Objects**. If demonstration data is installed, it is removed along with server components.
 - i Note:** After server components and QSA are uninstalled, manually delete demonstration datafiles, SPCDEMO01_nn.dbf and SPCDEMO02_nn.dbf.
3. If the uninstall program detects scripts that are being executed by QSA, you must cancel the uninstall. You can finish the uninstall after script execution is complete.
4. If the uninstall program determines that one or more scripts are scheduled to run, a message displays the date and time for the first script and asks if you want to continue the uninstall:
 - If you want to continue and are sure that the uninstall process will finish before the first script runs, click **Yes**.
 - If you want to cancel the uninstall, click **No**.
5. Once verifications are complete, a confirmation message explains that all Space Manager objects will be removed from the server. Click **Yes**. The SQL Exec dialog displays the uninstall progress and status.
6. After server components are uninstalled, the QSA Installer displays so that you can remove the agent. See [Uninstall the Server Agent](#) on page 199 for more information.

About QSA Server Agent

The QSA Server Agent is a script execution engine that Space Manager uses to run scripts, such as reorganization, repair, and statistics-collection scripts, from the database server on a scheduled basis. The agent must be used to run live reorganization scripts and scripts that use FastCopy, FSCopy, or parallel agent processes. Scripts are automatically run by the agent when they are scheduled and stored in the Space Manager repository.

You can stop and start QSA from the Space Manager Tools menu or the Script/Job Monitor window. See [Start or Stop QSA Server Agent](#).

To view information about the currently installed QSA Server Agent, select a database connection in the Explorer tree-list. Then click **About QSA** in the QSA Status panel.

- i Note:** For detailed information about installing or uninstalling QSA (using the QSA Installer or the manual method), specifying QSA user privileges, or specifying QSA parameters, see the ***Space Manager with LiveReorg Installation Guide***.

QSA Server Agent Installer

You can use the QSA Server Agent Installer to install, upgrade, and uninstall the QSA Server Agent for Oracle® databases on UNIX®, Linux®, and Windows® servers.

You can also use the QSA Installer to specify values for QSA parameters. Values are first specified when you install QSA. However, you can modify parameter values, if necessary, after installation by opening the QSA Installer.

To open the QSA Server Agent Installer

1. Select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
2. In the Select Installation Type window, select from the following:
 - i Note:** If the QSA Server Agent is not installed, the QSA Installer skips this window and displays the pages necessary for installation.

QSA Installer Option	Description
Upgrade QSA Server Agent	Select to upgrade to a new version of the Quest Server Agent.
Remove Quest Server Agent	Select to uninstall the Quest Server Agent.
Configure Quest Server Agent	Select to specify Quest Server Agent parameter values.

3. Click **Next** after completing each QSA Installer page. Click **Finish** to save your entries and close the installer.

For More Information

For more information about the QSA Installer, see the following:

- [Install or Upgrade Server Agent on UNIX or Linux](#)
- [Uninstall the Server Agent](#)
- [Specify Parameters for the QSA Server Agent](#)
- [Oracle Upgrade](#)

i Note: You can also run the QSA Installer from the Space Manager application folder. To run the QSA Installer, double click **QSAInstall.exe**.

Start or Stop QSA Server Agent

You can start or stop the Quest Server Agent (QSA) using administration functions from the **Tools | Admin | Quest Server Agent** menu.

i Note: If you are running Space Manager in an active-active OPS or RAC environment, you must connect to the instance where the Quest Server Agent is installed before you can start or stop the agent.

i Tip: You can also start or stop the Quest Server Agent from the Scripts/Job Monitor window. See [Manage and Monitor QSA](#) on page 143 for more information.

Start QSA Using the Admin Function

The Start Quest Server Agent function can be used, if necessary, to start the Quest Server Agent. In most cases, the agent starts automatically when it is installed. It then runs continuously, waiting to execute scripts at their scheduled times. Normally, the agent only needs to be restarted after it is stopped using the Stop Quest Server Agent function. When the agent is stopped, *Not running* is displayed in the Quest Server Agent Status panel in the Scripts/Job Monitor window.

The Start Quest Server Agent function can be used to start the agent on local Windows servers (those that reside on your Space Manager client computer) and on UNIX and Linux servers to which you have both SSH and SFTP access. To start QSA with other configurations, see the section "Start QSA Under Special Circumstances" in this topic.

To start QSA using the admin function

1. Select **Tools | Admin | Quest Server Agent | Start Quest Server Agent**.
2. If you are starting the agent on a UNIX or Linux server, the server login dialog displays. Enter the login name and password for the UNIX or Linux account that is being used to run the agent.
3. When the *Quest Server Agent Started* message displays, click **OK** to clear it.
4. Check the status of the agent from the Scripts/Job Monitor. *Running* is displayed in the Quest Server Agent Status panel when the agent is started.

i Note: If QSA is not running when a script's scheduled execution time occurs, you have 10 minutes in which to start it. The script is executed when QSA starts. If QSA starts after the 10-minute time window ends, the script aborts with this error: "QSA-20312: Scheduled launch time expired". To run a script whose launch time has expired, execute it on demand or reschedule it from the Scripts/Job Monitor.

Restart QSA Using the Admin Function

Use the Restart Quest Server Agent function when you do not have the permission level necessary to start the Quest Server Agent. The Restart Quest Server Agent function automatically stops and restarts the agent. This only requires the same permission level as that for stopping the agent.

To restart QSA using the admin function

- » Select **Tools | Admin | Quest Server Agent | Restart Quest Server Agent**.

After the agent restarts, a *Quest Server Agent restarted* message displays.

Stop QSA Using the Admin Function

The Stop Quest Server Agent function can be used, if necessary, to stop the Quest Server Agent. The agent should **not** be stopped when scheduled scripts are executing or are scheduled to execute.

The Stop Quest Server Agent function can be used to stop the agent on local Windows servers (those that reside on your Space Manager client computer) and on UNIX and Linux servers to which you have both SSH and SFTP access. To stop QSA with other configurations, see the section "Stop QSA Under Special Circumstances" in this topic.

To stop QSA using the admin function

1. Select **Tools | Admin | Quest Server Agent | Stop Quest Server Agent**.
2. When the *Quest Server Agent Stopped* message displays, click **OK** to clear it.
3. Check the status of the agent from the Scripts/Job Monitor. *Not running* is displayed in the Quest Server Agent Status panel when the agent is stopped.

Start QSA Under Special Circumstances

In most cases, you can start the Quest Server Agent using client administrative functions. However, you need to use one of the following procedures if the agent is installed on:

- A remote Windows server. This is a server that does not reside on your Space Manager client computer.
- A UNIX or Linux server to which the Space Manager client computer does not have both SSH and SFTP access.

To start QSA on a remote Windows server

1. Log in to the server as a member of the Windows Administrators Group.
2. Start the Quest Server Agent using one of the following methods:
 - From the Windows **Services** window, right-click **Quest Server Agent** and select **Start**.
 - From the \bin subdirectory of the agent install directory, run this command: `QSASstart.cmd`

To start QSA on a UNIX or Linux server

1. Log in to the server under the UNIX or Linux account that is being used to run the agent.
2. From the \bin subdirectory of the agent install directory, run this script: `QSASstart.sh`

Stop QSA Under Special Circumstances

In most cases, you can stop the Quest Server Agent using client administrative functions. However, you need to use one of the following procedures if the agent is installed on:

- A remote Windows server. This is a server that does not reside on your Space Manager client computer.
- A UNIX or Linux server to which the Space Manager client computer does not have both SSH and SFTP access.

To stop QSA on a remote Windows server

1. Log in to the server as a member of the Windows Administrators Group.
2. Stop the Quest Server Agent using one of the following methods:
 - From the Windows Services window, right-click **Quest Server Agent** and select **Stop**.
 - From the \bin subdirectory of the agent install directory on the server, run this command: `QSASstop.cmd`

To stop QSA on a UNIX or Linux server

1. Log in to the server under the UNIX or Linux account that is being used to run the agent.
2. From the \bin subdirectory of the agent install directory, run this script: `QSASstop.sh`

Specify Parameters for the QSA Server Agent

Parameters for the Quest Server Agent can be specified when the agent is installed or upgraded for a database. They can also be modified, when necessary, using the following procedures. In most cases, parameters can be specified using the QSA Installer (through the Space Manager client). However, they must be specified manually for databases on UNIX or Linux servers to which the Space Manager client computer does not have both SSH and SFTP access.

You can use QSA parameters to control the following activities:

- To determine how long execution logs and script histories are kept for Space Manager scripts.
- To control various aspects of live reorganizations and reorganizations with FastCopy.

- To configure QSA to send email notifications about script progress or script results. See [Enable Email \(SMTP\) Notification](#) on page 136 for more information.

i Note: To specify parameters, you must log in using the same account that installed the Quest Server Agent.

To specify parameters using the QSA Installer

1. Run Space Manager and connect to the target database for the Quest Server Agent. Connect as the Oracle DBA that installed the agent.
2. Select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
3. Select **Configure Quest Server Agent** and click **Next**.
4. If you are specifying the parameters on a UNIX or Linux server, the server login dialog displays. Enter the login name and password of the operating-system account that installed the agent.
 - i Important:** If the login window does not display the name that must be used to connect to the database server host, enter the correct name in the **Host Name** field.
5. In the QSA Parameters window, modify values for agent parameters as needed. See [QSA Server Agent Parameters](#) on page 189 for more information.
6. Click **Next** and then **Finish** to save your changes and close the QSA Installer.

i Note: For detailed descriptions of the Quest Server Agent parameters, see the *Space Manager with LiveReorg Installation Guide*.

To specify parameters manually on UNIX or Linux

1. Log in to the server under the UNIX or Linux account that installed the agent.
2. Change to the inst subdirectory of the Quest Server Agent installation directory using the following command: `# cd/$ORACLE_HOME/Quest/QSA/$ORACLE_SID/inst`
3. Run the manual configuration script using the following command and cases:


```
# ./QSAConfig
```
4. From the Quest Server Agent Parameters window, modify values for the agent parameters as needed:
 - At the Do you wish to modify any of these parameters prompt, enter Y (for Yes).
 - At the Please enter the ID number of the parameter prompt, enter the ID number of the parameter you want to change.
 - At each parameter prompt that displays, enter a new value for the parameter. Repeat for each parameter value you want to change.
 - When you finish entering parameters, enter N (for No) at the Do you wish to modify any of these parameters prompt.
 - At the Do you wish to save the parameter changes you've made prompt, enter **Y** (for Yes) to save all new values (enter **N** to revert to previous values). The *Parameters Updated* message is displayed.

i Note: By modifying certain QSA parameters, you can enable and configure automatic email notifications to monitor script execution progress. See the following for more information:

- [Enable Email \(SMTP\) Notification](#)
- [Enable SNMP Traps Notification](#)
- [Configure Additional Notification Events](#)

QSA Server Agent Parameters

This section describes QSA parameters. All parameters except LW_TABLESPACE and WORKDIR have default values. The user must specify LW_TABLESPACE and WORKDIR parameters. You can change default values as needed.

The following table contains a list of Quest Server Agent parameters and descriptions.

Parameter	Description
LW_TABLESPACE	<p>LW_TABLESPACE specifies the tablespace for objects created by QSA during a live reorganization. One set of LiveReorg objects is created for each table in a live reorganization script. The objects are automatically dropped after a table is reorganized.</p> <p>LW_TABLESPACE must have a standard name and a datafile that is at least 50 MB in size. The datafile must have at least 20 MB of contiguous free space.</p> <p>Note: A tablespace must be selected for the LW_TABLESPACE parameter even if LiveReorg is not licensed.</p>
WORKDIR	<p>WORKDIR specifies the main directory that FastCopy (or DBMS_DataPump) should use for the data exported to the file system during a reorganization. If QSA is installed on a remote Windows server (separate from the Space Manager client computer), specify this directory as if it were on a local drive, for example, C:\TEMP. Enter the directory and path in ASCII characters.</p>
COMMIT_SIZE	<p>COMMIT_SIZE determines the amount of data FastCopy or FSCopy imports from the file system into the database before a COMMIT is performed. The parameter value can range from 1 MB to 1024 MB. The default is 10 MB.</p>
COUNT_STAR	<p>This parameter determines whether Quest Server Agent (QSA) performs a select count(*) on the orig_table and compares this to the number of rows copied. Select one of the following to indicate whether you want the agent to perform this extra check.</p> <p>Select 1 to perform the check. This is the default.</p> <p>Select 0 to NOT perform the check.</p>
DISKSPACE_RESERVE	<p>DISKSPACE_RESERVE determines how much space is reserved for applications in WORKDIR and OVERFLOW directories. Unreserved space can be used by FastCopy/DBMS_DataPump or FSCopy for data exported during a reorganization. Once FastCopy/DBMS_DataPump or FSCopy uses all unreserved space in the WORKDIR directory, it uses unreserved space in the first OVERFLOW directory. The parameter value can range from 1 MB to 1024 MB. The default is 10 MB.</p>
EXPORT_BUFFER_SIZE	<p>EXPORT_BUFFER_SIZE determines the maximum size of the export buffer used by FastCopy or FSCopy. Reorganizations are faster when export buffer size is at least the size of the largest row being reorganized. The parameter value can range from 32 KB to 1024 MB. The default is 1 MB.</p>
IMPORT_METHOD	<p>IMPORT_METHOD determines how FastCopy or FSCopy imports data from the file system into the database. There are three options:</p> <ul style="list-style-type: none"> AUTO—FastCopy or FSCopy uses several factors (unsupported data types, rows larger than 64 KB, and SQL*Loader errors) to determine whether to use the direct or conventional import method. FastCopy or FSCopy tries the direct method first. If unsuccessful, it automatically switches to the conventional method.

Parameter	Description
	<ul style="list-style-type: none"> • DIRECT—FastCopy or FSCopy uses SQL*Loader’s direct path load. The direct method is faster than the conventional because it uses the direct path API to move data and bypasses SQL processing. FastCopy or FSCopy aborts the reorganization, however, if it encounters an unsupported datatype, a row larger than 64 KB, or a SQL*Loader error. • CONVENTIONAL—FastCopy or FSCopy uses OCI array inserts to move data. Although slower than the direct method, the conventional method has fewer limitations and associated problems. This is the default.
IMPORT_BUFFER_SIZE	IMPORT_BUFFER_SIZE determines the maximum size of the import buffer that FastCopy or FSCopy uses during imports with the conventional method. The parameter value can range from 1 MB to 1024 MB. The default is 10 MB.
LW_NULL_COLUMNS	This parameter determines the maximum number of columns to check for a NULL value. The default is 32 and should not be changed.
MAX_FILE_SIZE	MAX_FILE_SIZE determines the maximum size of FastCopy or FSCopy export files, which are used for the data exported from the database to the file system. (FastCopy files are in a proprietary format.) When the first file reaches the maximum size, a second file is created automatically, and so on. The parameter value can range from 1 MB to 1024 MB. (It must be less than your system’s default maximum file size.) The default is 100 MB.
MAX_HISTORY_AGE	MAX_HISTORY_AGE determines the maximum number of days that records are kept in the QUEST_SCRIPT_HISTORY table. The table contains a history of script execution by QSA and is used to associate script logs with their scripts. The parameter value can range from 0 to 32765. (It should be greater than the value for MAX_LOG_AGE.) The default is 365.
MAX_LOG_AGE	MAX_LOG_AGE determines the maximum number of days that script execution logs and QSA log are kept. Script logs can be used to troubleshoot script execution by the agent. The agent log (QEXECD.LOG) can be used to troubleshoot all agent activity. You can view logs from Space Manager’s Script/Job Monitor. You can send them to Quest Software Support . The parameter value can range from 0 to 365. (0 retains logs until Space Manager is uninstalled.) The default is 30.
MULTIBLOCK_READ_COUNT	MULTIBLOCK_READ_COUNT determines the number of data blocks that Oracle reads at one time during a FastCopy reorganization. The number should be a multiple of the size of your operating system read buffer divided by your database block size. For example, if your operating system read buffer is 256 KB and your database block size is 8 KB, set the parameter to a multiple of 32 KB. The parameter value must be greater than or equal to 0. The default is 100.
OVERFLOW_DIR_01 thru DIR_05	<p>OVERFLOW_DIR specifies directories that can be used in addition to the WORKDIR directory for the data FastCopy/DBMS_DataPump exports. FastCopy/DBMS_DataPump uses overflow directories when WORKDIR is full. You can specify up to 5 directories. A new directory is used when the last one used is full. Directory and path names must be entered in ASCII characters.</p> <p>i Note: Each overflow directory must be on a different disk. The WORKDIR disk must not be used for an OVERFLOW_DIR.</p>

SENDMAIL Parameters

SENDMAIL parameters allow you to configure QSA to automatically send email notifications when scheduled script execution fails. Notifications identify the failure reason, script name, execution start and stop times, database server hostname, and database Oracle SID. The subject field displays "Script Failure Notice!"

SENDMAIL includes the following parameters (all except SENDMAIL_TO1 require values).

QSA Parameter	Description
SENDMAIL_DOMAIN	Specifies the domain name of the mail server or relay server. For example: yourcompany.com.
SENDMAIL_HOST	Specifies the address or fully-qualified name of the mail server or relay server host, for example: www.quest.com.
SENDMAIL_PORT	Specifies the port number on which the mail server or relay server is listening.
SENDMAIL_FROM	Specifies who sent the email notification. The text string you enter should be the email address of an individual or group, such as Administrator@yourcompany.com.
SENDMAIL_TO	Specifies the email recipient. Must be the email address of an individual or group, such as DBA@yourcompany.com. The address must be valid. Only one address may be entered.
SENDMAIL_TO1	Specifies a second recipient for email notifications. To specify three or more recipients, insert a row for each recipient in the QUEST_EXEC_PARAMETER table, where NAME is SENDMAIL_TO# and VALUE is a valid email address. Each SENDMAIL_TO# entry should increment by 1; for example, SENDMAIL_TO3, SENDMAIL_TO4. If there is a break in the number sequence, emails are not sent to recipients after the break. If you delete a recipient row, update the NAME rows that follow so the number sequence is preserved. You can specify a maximum of 99 recipients.

i Note: For QSA to send email, Oracle's UTL_SMTP package must be installed on the database server. The package must have access to an SMTP mail server or relay server.

SNMP Trap Parameters

The following parameters are used to configure Simple Network Management Protocol (SNMP) Trap notifications.

QSA Parameter	Description
SNMP_TO_HOST	Enter the host name for the SNMP monitor.
SNMP_PORT	Enter the port number that the SNMP monitor is listening on.
SNMP_COMMUNITY	Enter the name of your SNMP community (or any unique string value).

Additional Notification Parameters

After enabling SMTP email or SNMP trap notification methods, you can specify additional notification events by configuring the following parameters.

Parameter	Description
SEND_GROUP_STATUS	Send notification of a static change to the group status.
SEND_SCRIPT_START	Send notification when script starts (101).
SEND_SCRIPT_RESTART	Send notification when script is restarted (104).
SEND_SCRIPT_UNCHEDULED	Send notification when script is unscheduled (110).
SEND_SCRIPT_COMPLETE	Send notification when script completes (210).
SEND_SCRIPT_CANCELLED	Send notification when script is cancelled (310).
SEND_SCRIPT_ABORT	Send notification when script aborts (320).
SEND_SCRIPT_HALTED	Send notification when script is halted (370).
SEND_SCRIPT_SCHEDULED	Send notification when a script is scheduled (410).
SEND_LR_WAIT_WINDOW	Send notification when a LiveReorg is waiting for window (620).
SEND_LR_IN_WINDOW	Send notification when a LiveReorg is in the window (621).
SEND_LR_WAIT_APPROVAL	Send notification when a LiveReorg is waiting for approval (630).
SEND_LR_APPROVED	Send notification when a LiveReorg is approved (631).
SEND_LR_START	Send notification when a LiveReorg starts (710).
SEND_LR_RESTART	Send notification when a LiveReorg restarts (715).
SEND_LR_COPY_START	Send notification when a LiveReorg table copy starts (720).
SEND_LR_COPY_COMPLETE	Send notification when a LiveReorg table copy completes (730).
SEND_LR_SWITCH_START	Send notification when a LiveReorg starts to switch (740).
SEND_LR_SWITCH_COMPLETE	Send notification when a LiveReorg switch completes (750).
SEND_LR_COMPLETE	Send notification when a LiveReorg completes (760). Setting this parameter sends an email without a results report. To include a results report, use one of the SEND_LR_COMPLETE parameters listed below.

SEND_LR_COMPLETE Options

Use the following parameters to specify the type of results report to include when using the SEND_LR_COMPLETE parameter.

Parameter	Results Report Type	Description
SEND_LR_COMPLETE	No report	No report
SEND_LR_COMPLETE_DATA	Detail objects, summary by object	This report is the most detailed and includes the object details, detailed storage data, and a

Parameter	Results Report Type	Description
	type and total savings reports.	storage data summary by object.
SEND_LR_COMPLETE_SUM	Summary by object type and total savings reports.	This report delivers a summary of the storage data by object.
SEND_LR_COMPLETE_TOTAL	Total savings report.	This report lists the storage data totals only.

Locks Reporting Parameters

Use the following parameters to configure advanced options for locks reporting.

Parameter	Description
LOCKS_REPORTING	Select YES to enable locks reporting. <i>Default: NO</i> i Note: When you disable locks reporting, the values selected for the other parameters have no effect.
LOCKS_CANCELED_JOBS	Select YES to enable reporting of locks for canceled jobs. <i>Default: YES</i> i Note: Space Manager reports locks held by Oracle sessions for canceled jobs to the qexecd.log file before canceling the job.
LOCKS_SCRIPT_LOG	Select YES to enable reporting of locks when a table lock or a T-Lock fails with a resource busy error (ORA-0054). <i>Default: YES</i> i Note: Space Manager reports these locks to the script execution log that you can view using the Script/Job Monitor.
LOCKS_SYSTEM_FILE_LOGS	Select YES to enable reporting of locks to a system file. <i>Default: NO</i>
LOCKS_SYSTEM_FILE_APPEND	Select YES to append the system file created for locks reporting. <i>Default: YES</i> i Note: If you disable this parameter, the system file only contains information for the most recent lock reported.
LOCKS_MAXIMUM_REPORTS	Enter a value for the number of times to report locks for a specific table. <i>Range: 1 to 999</i> <i>Default: 3</i> i Tip: A single lock may be retired thousands of times during a single reorganization. Use the default value to minimize changes to the system file.
i Note: Space Manager creates system files in the QSA log file directory with the following naming convention: <owner>.<folder>.<script>.<jobnumber>.lok	

QSA Server Agent Locks Reporting Parameters

You can use parameters for the Quest Server Agent (QSA) to configure advanced options for locks reporting. Space Manager uses locks reporting when an Oracle database has sessions holding locks that prevent LiveReorgs from starting, syncing, or switching. Held locks occur from table locks, row inserts, deletes, or updates. When you enable locks reporting, Space Manager reports Oracle session information for sessions with locks. The majority of the session information reported comes from the Oracle View v\$lock file. You can configure the locks reporting parameters to report locks for a specific table when a triggering event occurs. The values chosen in the QUEST_EXEC_PARAMETER table determine how QSA reports locks.

i Notes:

- Parameters must be set by the account that installed the Quest Server Agent.
- This topic focuses on information that may be unfamiliar to you. It does not include all step and field descriptions.

To set locks reporting parameters

1. Start Space Manager and connect to the target database for the Quest Server Agent.
2. Select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
i **Note:** You can also set parameters manually. See [Specify Parameters for the QSA Server Agent](#) on page 187 for more information.
3. In the Quest Server Agent dialog, select **Configure Quest Server Agent** and click **Next**.
4. If you are configuring parameters on a Unix or Linux server, a log-in window displays. Complete the log-on information and click **Next**.
i **Note:** To configure parameters, you must use the user login and password of the operating system account that installed the agent.
5. Configure the parameters for locks reporting. Review the following for additional information:

Parameter	Description
LOCKS_REPORTING	Select YES to enable locks reporting. <i>Default: NO</i> i Note: When you disable locks reporting, the values selected for the other parameters have no effect.
LOCKS_CANCELED_JOBS	Select YES to enable reporting of locks for canceled jobs. <i>Default: YES</i> i Note: Space Manager reports locks held by Oracle sessions for canceled jobs to the qexecd.log file before canceling the job.
LOCKS_SCRIPT_LOG	Select YES to enable reporting of locks when a table lock or a T-Lock fails with a resource busy error (ORA-0054). <i>Default: YES</i> i Note: Space Manager reports these locks to the script execution log that you can view using the Script/Job Monitor.

Parameter	Description
LOCKS_SYSTEM_FILE_LOGS	Select YES to enable reporting of locks to a system file. <i>Default: NO</i>
LOCKS_SYSTEM_FILE_APPEND	Select YES to append the system file created for locks reporting. <i>Default: YES</i> i Note: If you disable this parameter, the system file only contains information for the most recent lock reported.
LOCKS_MAXIMUM_REPORTS	Enter a value for the number of times to report locks for a specific table. <i>Range: 1 to 999</i> <i>Default: 3</i> i Tip: A single lock may be retired thousands of times during a single reorganization. Use the default value to minimize changes to the system file.
i Note: Space Manager creates system files in the QSA log file directory with the following naming convention: <owner>.<folder>.<script>.<jobnumber>.lok	

QSA Server Agent Locking Strategies

You can use parameters for the Quest Server Agent (QSA) to configure advanced options for locking strategies. The following locking strategies are available for a LiveReorg.

Wait Mode

Wait mode is the default locking strategy used for a LiveReorg. This locking strategy assumes the success of the LiveReorg is the primary objective. When Space Manager issues a table lock using this strategy, the lock type is a blocking exclusive table lock. Blocking occurs within Oracle and causes the lock to wait in a queue until the process obtains the lock.

Wait for No Locks

The Wait for No Locks strategy forces the LiveReorg to wait until a table has no locks before attempting to lock the table. Lock attempts for a table are indicated by an asterisk (*) before the keyword Locking in the status message of the Script/Job Monitor. The process scans the table for locks. If the process finds a lock, it sleeps for a second and rescans the table locks. Once the process does not find a lock on table, it attempts the lock.

No-Wait Mode

You can use No-wait mode as the locking strategy for the initial table copy and switch when the success of the LiveReorg is not the primary objective. This strategy should not be used in most environments. The advantage of this strategy is that the exclusive table lock never blocks and prevents other locks from completing.

- i Note:** You can configure table copy synchronization or online switch locking to use No-Wait mode. However, all QSA parameters to enable No-wait mode are hidden by default. Contact Quest Software technical support for instructions to enable No-Wait mode.

Install or Upgrade Server Agent on UNIX or Linux

Use the following procedure to install (or upgrade) the Server Agent in an Oracle database on a UNIX or Linux server. You can use the Server Agent Installer if you have both SSH and SFTP access to your database server. If you do not have both, you must install the agent manually. See "Manually Install Server Agent on UNIX or Linux" in the *Space Manager Installation Guide* for more information.

Install the Server Agent using the same DBA account used to install the server components. For information about installation DBA account requirements, see [Space Manager User Requirements](#).

i Notes:

- If the target database is in an OPS or RAC system that uses the active-active configuration, see "Install Server Agent on OPS or RAC" in the *Space Manager Installation Guide* for more information.
- If upgrading the server agent, perform the upgrade when no scheduled scripts are executing or about to execute.

To install (or upgrade) the Server Agent on UNIX or Linux

1. Run the Space Manager client and connect to the target database as the DBA that installed the Space Manager server components.
 - i** **Note:** If the Server Agent Installer automatically opens after installing the server components, it opens to the Log on to Server page. Skip ahead to step 4.
2. To run the Server Agent Installer, select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
3. (Upgrade only) If you are upgrading a currently installed instance of QSA, select **Upgrade Quest Server Agent** on the Select Installation Type page and click **Next**.
4. In the Log on to Server window, enter the user name and password of the UNIX or Linux account. Enter the host server name if it is not displayed. Click **Next**.
5. In the Oracle Connection Info window, review the information. If necessary, complete or modify any fields. Click **Next**. If you are upgrading, you can only edit the staging directory path in this window.
6. In the Install Quest Server Agent window, click **Install** to install QSA. In the Agent Installation Complete window, click **Next**.
7. In the QSA Parameters window, specify values for QSA parameters or use the default values.

i **Note:** Deselect Show only required fields with missing value to see all parameters.

Values apply to the target database. You must enter values for **WORKDIR** (used for FastCopy/DBMS_DataPump) and **LW_TABLESPACE** (used for live reorganizations and required even if LiveReorg is not licensed). Click **Next** when finished. See [QSA Server Agent Parameters](#) on page 189 for more information.

8. In the Installation Finished window, click **Finish** to exit the Server Agent Installer.

i **Note:** Before the Server Agent is installed, you may encounter one of the following messages:

- **Files not current**—The agent installation files are either not available or not current. Exit the Server Agent Installer, install the most recent Space Manager client. New agent files are copied to Server Agent folder on the client computer. You can then install the agent.
- **Environment not supported / cannot proceed**—Both your Oracle version and operating system version are not supported. The agent cannot be installed. You must upgrade to a supported Oracle version and a supported operating system. See the *Space Manager Release Notes* for a list of supported Oracle Database and Operating System versions.

- **Environment not supported / can proceed**—Either your Oracle version or your operating system version is not supported. In most cases, a supported version of Oracle is running on an unsupported operating system version. Although you can proceed with the installation, you may have problems.

To instruct the Server Agent to restart automatically after a server restart

If you want the Server Agent to start automatically each time the server is started, perform the following steps:

1. Log in to UNIX or Linux as **root**.
2. Change to the agent installation directory using the following command:

```
# cd /$ORACLE_HOME/Quest/QSA/SID/inst
```

3. Run the following root.sh script:

```
# ./root.sh
```

Install Server Agent on Windows

This section describes how to install the Server Agent on a local or a remote Windows server:

- **Local server**—A server is local if it resides on your Space Manager client computer. When a server is local, you must work from the client computer to install the Server Agent. It must be installed on a local drive.
- **Remote server**—A server is remote if it does not reside on your Space Manager client computer. When a server is remote, you should work from the server to install the Server Agent.

Install the Server Agent using the same DBA account used to install the server components. For information about installation DBA account requirements, see [Space Manager User Requirements](#).

Prerequisites

- For installation on either a local or a remote Windows server, the installation Oracle DBA must have all permissions for the agent installation folder.
- On a remote server, the folder must be shared or copied to the server.

i Note: If upgrading the server agent, perform the upgrade when no scheduled scripts are executing or about to execute.

Install the Agent on a Local Server

Use the following procedure to install QSA in a database on a local Windows server.

To install (or upgrade) the Server Agent on a local Windows server

1. Run the Space Manager client and connect to the target database using the DBA account that installed the Space Manager server components.
 - i Note:** If you installed (or upgraded) the server components, the Server Agent Installer opens automatically after component installation is finished. Skip ahead to step 4.
2. To run the QSA Installer, select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
3. (Upgrade only) If you are upgrading QSA, the Select Installation Type window displays. Select **Upgrade Quest Server Agent** and click **Next**.

4. In the Oracle Connection Info window, review the information. If necessary, complete or modify any fields. Click **Next**. If you are upgrading, you can only edit the staging directory path in this window.
5. In the Install Quest Server Agent window, click **Install** to install QSA. In the Agent Installation Complete window, click **Next**.
6. In the QSA Parameters window, specify values for QSA parameters or use the default values. Values apply to the target database. You must enter values for **WORKDIR** (used for FastCopy/DBMS_DataPump) and **LW_TABLESPACE** (used for live reorganizations and required even if LiveReorg is not licensed). Click **Next** when finished. See [QSA Server Agent Parameters](#) on page 189 for more information.
7. In the Installation Finished window, click **Finish** to exit the QSA Installer.

i Note: Before the Server Agent is installed, you may encounter one of the following messages:

- **Files not current**—The agent installation files are either not available or not current. Exit the Server Agent Installer, install the most recent Space Manager client. New agent files are copied to Server Agent folder on the client computer. You can then install the agent.
- **Environment not supported / cannot proceed**—Both your Oracle version and operating system version are not supported. The agent cannot be installed. You must upgrade to a supported Oracle version and a supported operating system. See the *Space Manager Release Notes* for a list of supported Oracle Database and Operating System versions.
- **Environment not supported / can proceed**—Either your Oracle version or your operating system version is not supported. In most cases, a supported version of Oracle is running on an unsupported operating system version. Although you can proceed with the installation, you may have problems.

Install the Agent on a Remote Server

To install QSA on a remote Windows server, you must copy the QSA installer from the Space Manager application directory to the Windows server.

To manually install QSA on Windows, see "Manually Install and Uninstall the Server Agent on Windows" in the *Space Manager Installation Guide*.

i Note: Best practice is to install locally. During remote installation, if you encounter problems or error messages such as "QSA-20375: Cannot install the daemon." or "Network path not found" use the local install procedure.

To copy the QSA Installer to remote Windows server

1. Navigate to the Space Manager client application directory:
 - In Windows Explorer, navigate to: C:\Program Files\Quest\Space Manager <version number>
2. Open the QuestServerAgent folder and copy QSA_<version>_build-QSASetup.exe to the Windows server.

To install the Server Agent on a remote Windows server

1. After copying the Server Agent installer (QSA_<version>_build-QSASetup.exe) to the Windows server, double-click QSA_<version>_build-QSASetup.exe to run it.
2. Extract the files to a temporary directory.
 - i Note:** Make sure the temporary directory is empty before you extract the files.
3. After the files are extracted, the GUI installer automatically opens.
4. In the Connect to Oracle Database window, enter the service name of the target database and connect using the DBA account that installed the Space Manager server components.
5. Follow the instructions in the previous section, "Install the Agent on a Local Server," beginning with step 3.

6. After installation, from the Windows Services window, verify that QSA is running. If **Started** does not display as the QSA status, start the agent from the Services window or run the QSASStart.cmd from the \bin folder under the agent installation folder (<installation path>\Quest Server Agent\<ORACLE_SID>\bin).

i Tip: If the agent is installed for more than one database on the server, look for the “Quest Server Agent” name that includes the target database SID.

Uninstall the Server Agent

The Server Agent must be uninstalled individually from each database for which it is installed. It must be uninstalled by the user that installed it.

i Note: If you used the QSA Installer or the manual install method to install the agent, use the same method to uninstall it.

Uninstall QSA from UNIX or Linux

This section describes how to uninstall QSA from an Oracle database on a UNIX or Linux server with the QSA Installer. You can use the QSA Installer if you have both SSH and SFTP access to the database server. If you do not have both, you must use a manual procedure. See "Manually Uninstall Server Agent from UNIX or Linux" in the *Space Manager Installation Guide* for more information.

To uninstall the agent from UNIX or Linux

1. Run the Space Manager client and connect to the target database as the DBA that installed QSA.
i Note: After uninstalling the server components, the Server Agent Installer automatically opens to the Log on page. Skip ahead to step 4.
2. To run the QSA Installer, select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent**.
3. In the Select Installation Type window, select **Remove Quest Server Agent**, and then click **Next**.
4. In the Log on to Server window, enter the user name and password for the UNIX or Linux account that installed the agent. Enter the host server name if it is not displayed. Click **Next**.
5. In the Uninstall Quest Server Agent window, click **Remove**.
6. In the Installation Finished window, click **Finish** to exit the QSA Installer.
7. If you did not uninstall QSA as root, log into the server as root and run the ROOT_UNINSTALL.SH script as follows. Use SID for the target database.

```
# /tmp/root_uninstall.sh SID
```

Uninstall QSA from Windows

This section describes how to uninstall the Server Agent from a local or remote Windows server.

i Note: The agent must be running when you uninstall it from a Windows server. You can start the agent from the Windows Services window, or run QSASStart.cmd in the \bin folder under the agent install folder (<installation path>\Quest Server Agent\<ORACLE_SID>\bin).

To manually uninstall the Server Agent from Windows, see "Manually Uninstall Server Agent from Windows" in the *Space Manager Installation Guide*.

Local Windows Server

To uninstall QSA from a local Windows server, you can run the Server Agent Installer from the Space Manager application.

To uninstall QSA from a local Windows server

1. Run the Space Manager client and connect to the target database as the DBA that installed QSA.
 - i Note:** After uninstalling server components, the Server Agent Installer automatically opens. Skip ahead to step 3.
2. To run the QSA Installer, select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent**.
3. In the Select Installation Type window, select **Remove Quest Server Agent**, and then click **Next**.
4. In the Uninstall Quest Server Agent window, click **Remove**.
5. In the Installation Finished window, click **Finish** to exit the QSA Installer.

Remote Windows Server

To uninstall QSA from a remote Windows server, you must copy the Server Agent Installer from the Space Manager application directory to the Windows server. (Installing QSA on the remote Windows server also required copying the Server Agent Installer to the Windows server.)

To copy the QSA Installer to remote Windows server

1. Navigate to the Space Manager application directory:
 - In Windows Explorer, navigate to: C:\Program Files\Quest\Space Manager <version number>
2. Open the QuestServerAgent folder and copy QSA_<version_build>-QSASetup.exe to the Windows server.

To uninstall QSA from a remote Windows server

1. Log in to the remote Windows server as a member of the Windows Administrators Group.
2. After copying the Server Agent installer (QSA_<version_build>-QSASetup.exe) to the Windows server, double-click QSA_<version_build>-QSASetup.exe to run it.
3. Follow the instructions in the previous section, "To uninstall QSA from a local Windows server" beginning with step 3.

Oracle Upgrade

If you plan to upgrade from one version of Oracle Database to another, uninstall QSA first. After the new Oracle database is installed, reinstall the server and the QSA agent.

When upgrading Oracle

1. Uninstall the Server Agent.
 - a. From the Space Manager main menu, select **Tools | Admin | Quest Server Agent | Manage Quest Server Agent Installation**.
 - b. Select **Remove Quest Server Agent** and complete the Server Agent Installer wizard. See [Uninstall the Server Agent](#) on page 199 for more information.
2. Upgrade Oracle Database.

3. Reinstall server components.
 - a. Return to Space Manager and from the main menu select **Tools | Admin | Reinstall Space Manager Server Objects**. This recreates all the Space Manager views and triggers the installation of the Server Agent.
 - b. When installation of the server components is finished, the Server Agent Installer opens and prompts you to install the agent.
4. Install the Server Agent. This ensures that the correct Oracle libraries are used when you upgrade Space Manager. See [Install or Upgrade Server Agent on UNIX or Linux](#) on page 196 for more information.

Change Repository Update Schedule

The Space Manager Repository is used to store object and tablespace lists and statistics. This information is used in reports, in reorg need calculations, and to display in the Explorer. Space Manager updates the Repository daily, by default. However, you can modify the schedule at which the Repository Update job runs.

i Tip: The Repository Update job is listed in the Script/Job Monitor. You can access the script execution log from the Job Monitor window.

To Set or Modify Repository Update Schedule

1. Select **Tools | Admin | Change Repository Update Schedule**.
2. In the Schedule Task dialog, select the frequency with which you want to update the Repository: daily, weekly or monthly. Click **Next**.

i Note: The default is *Daily*.

3. Select the time of day and other details for scheduling the update. Click **Finish** when done.

i Tips:

- You can immediately initiate the update process for a Repository by setting the next scheduled update to the current day and time. This starts the update process, which can run for some time.
- You can view the status of a database's Repository in the Explorer. Select a database in the Explorer tree-list to display the date of the last update and whether an update is scheduled.
- You can specify the method for Space Manager to use when gathering object and tablespace lists displayed in the Explorer by selecting **Tools | Options | Explorer Object Lists**. See [Explorer Object Lists](#) on page 156 for more information.
- You can specify the number of segments to update in each Repository Update job. See [Repository Schedule Options](#) on page 156 for more information.

Install or Uninstall Demonstration Data

Demonstration data consists of objects and tablespaces that suffer from the space-use problems found in a typical database. You can use the data to learn how Space Manager detects and resolves these problems.

After you resolve problems, you can refresh demonstration data. This restores its original condition.

i Note Demonstration data functions are only enabled for DBAs. They are disabled for users who do not have the DBA role.

Install Demonstration Data

You can install or refresh demonstration data at any time. The data requires 20 MB or 24 MB of disk space, depending on the database block size. The demonstration user is created when you install demonstration data. This user is the owner of all demonstration objects.

! **CAUTION:** When demonstration data is uninstalled or refreshed, all objects owned by the demonstration user are dropped. To avoid losing objects, do not create objects under the demonstration user's name.

i **Note:** When you install server components, you have the option to install demonstration data at the time.

To install or refresh demonstration data

1. Run Space Manager and connect to the target database as a DBA.
2. Select **Tools | Admin | Demonstration Data**.
3. Use default entries for demonstration username, password, tablespace names, and datafile names. A password is not needed. If you specify different directories for demonstration datafiles, the directories must already exist.
4. Click **Install** if you are installing demo data for the first time. Click **Refresh** if you are restoring demo data to its original condition.

Uninstall Demonstration Data

Uninstalling demo data removes demonstration tablespaces and objects, and the demonstration user. Datafiles for the two demonstration tablespaces must be removed manually from the database server.

! **CAUTION:** When demonstration data is uninstalled or refreshed, all objects owned by the demonstration user are dropped. To avoid losing objects, do not create objects under the demonstration user's name.

If demonstration tablespaces contain objects that are not owned by the demonstration user, relocate these objects before uninstalling demonstration data. Demonstration tablespaces and objects are not removed if demo tablespaces contain objects that are owned by a user other than the demo user.

i **Note** When you uninstll server components, demonstration data is automatically uninstalled along with the components.

To uninstall demo data

1. Run Space Manager and connect to the target database as a DBA.
2. Select **Tools | Admin | Demonstration Data**.
3. Click **Uninstall**. Then click **Yes** to clear the confirmation message that displays.
4. When the uninstall is complete, click **OK** to clear the message reminding you to delete tablespace datafiles.
5. From the database server host, manually delete datafiles for the two demonstration tablespaces. The names of the datafiles are SPCDEMO01_nn.dbf and SPCDEMO02_nn.dbf.

Install Analysis Packages

The **Install Analysis Option** function can be used to install analysis packages if they were not installed along with server components. When analysis packages are missing, you are prompted to install them by statistics-

collection functions.

To install analysis packages

- **Select Tools | Admin | Install Analysis Option.**

Use the Find Long LONGs Package

Space Manager with LiveReorg includes the Find Long LONGs package. This allows users and reorg tools to easily identify tables with “long LONG” columns. These are LONG or LONG RAW columns that are larger than 32,760 bytes in size.

The size of LONG and LONG RAW columns determines which data movement method can be used to reorganize their tables. Tables that contain LONG columns in which some rows contain data larger than 32,760 bytes must be reorganized with FastCopy (or DBMS_DataPump). Tables that contain LONG columns where all rows are 32,760 bytes or less can be reorganized with either FastCopy/DBMS_DataPump or SQL (PL/SQL is used).

If a table is not supported for FastCopy (or DBMS_DataPump) and its LONG columns are 32,760 bytes or less, run the long LONGs procedure to provide this information to Space Manager. This information may enable Space Manager to reorganize the table. If Space Manager does not know the size of a LONG column, its table is excluded from reorganization.

Reorg Manager automatically checks the Find Long LONGs table to identify tables with long LONGs. You can query the table yourself to identify these tables. See [Query the Find Long LONGs table](#) on page 206 for more information.

The Find Long LONGs package (QUEST_SPC_FIND_LONG) is installed with Space Manager. (The package is installed via the SPC_PKG_FIND_LONG.SQL script, which is located in the INSTALL subdirectory of the Space Manager client directory.) The package includes the following:

- Find Long LONGs table—Called QUEST_SPC_LONG_TABLES, this contains information on the size of any LONG or LONG RAW columns contained in your tables. The Find Long LONGs table is created when you install Space Manager server components; it is located in the tablespace you specify for server components.
- Find Long LONGs procedure—Called QUEST_SPC_FIND_LONG.EXECUTE, this procedure populates the Find Long LONGs table with information on the size of any LONG or LONG RAW columns contained in your tables.

For more information about the Find Long LONGs procedure or to learn how to run the procedure, see [Populate the Find Long LONGs table](#).

i Important: Before you can use the Find Long LONGs package, Oracle’s UTL_RAW package must be installed and valid on your database server.

Populate the Find Long LONGs table

This section covers populating the Find Long LONGs table with data on the size of any LONG or LONG RAW columns contained in your tables.

You should populate the Find Long LONGs table before you begin using Reorg Manager. Both reorg tools check this table for the size of LONG and LONG RAW columns.

To keep the Find Long LONGs table up-to-date with the latest sizes of LONG columns, you should re-populate it from time to time. You should also re-populate it after you create tables with LONG columns.

The Find Long LONGs Procedure

The Find Long LONGs table is populated with the Find Long LONGs Execute procedure. This is displayed below:

```
begin
quest_spc_find_long.execute (schema=>'SCHEMA', include_sys=>'NO',
tablename=>'TABLENAME');
end;
/
```

The Find Long LONGs procedure can scan an entire database, a certain schema, or a certain table. You specify what to scan using the parameters described below. When no parameters are used, the procedure scans the entire database (with the exception of SYS and SYSTEM objects):

- **schema**—Use this parameter if you want to scan a certain schema.
- **include_sys** - [yes | no]—Set this parameter to yes if you want to scan SYS and SYSTEM objects. If you do not want to scan these objects, omit the parameter or set it to no.
- **tablename**—Use this parameter if you want to scan a table with a certain name. To scan all tables with a certain name, use the tablename parameter on its own, without the schema parameter. To scan a certain table in a certain schema, use the schema parameter along with the tablename parameter.

To populate the Find Long LONGs table

1. Select **File | New Script** to open the SQL Editor.
2. Run the Find Long LONGs Execute procedure for all tables in a database, for a certain schema, or for a certain table using parameters as follows:
 - To scan all tables in a database, run the procedure using no parameters or all parameters (if you don't use any parameters, SYS and SYSTEM tables are excluded by default):

```
begin
quest_spc_find_long.execute;
end;
/
```

or

```
begin
quest_spc_find_long.execute (schema=>'ALL', include_sys=>'NO',
tablename=>'ALL');
end;
/
```

- To scan all tables in a certain schema, run the procedure using the schema parameter:

```
begin
quest_spc_find_long.execute (schema => 'SCHEMA');
end;
/
```

- To scan all tables with a certain name, run the procedure using only the tablename parameter:

```
begin
quest_spc_find_long.execute (tablename=> 'TABLENAME');
end;
/
```

- To scan a certain table in a certain schema, run the procedure using the schema and tablename

parameters:

```
begin
quest_spc_find_long.execute (schema=>'SCHEMA', tablename=> 'TABLENAME');
end;
/
```

i Tip: To include SYS and SYSTEM tables in a scan, run the procedure with the `include_sys` parameter set to **YES** as follows: `include_sys=>'YES'`

Messages for the Find Long LONGs Package

To provide you with feedback on the results of a scan, the Find Long LONGs procedure includes several messages. When a messages applies, it is displayed at the bottom of the SQL Editor after a scan is complete.

When a scan finds one or more tables with long LONGs, the following message alerts you to this and recommends that you check the Find Long LONGs table to identify these tables:

“ORA-20120: Longs that cannot be reorganized with PL/SQL found. Check QUEST_SPC_LONG_TABLES to see which tables cannot be reorganized using PL/SQL. These tables will have a YES value in the MUST_USE_FASTCOPY column. Fastcopy (or DBMS_DataPump) needs to be used.”

When a scan does not find any tables with long LONGs, the following message is displayed:

“ORA-20121: No tables with long LONG columns found”

When one or more tables is not accessible during a scan, the following message alerts you to this (a table is not accessible if its tablespace is offline):

“ORA-20122: Table(s) not accessible to scan”

Query the Find Long LONGs table

You can query the Find Long LONGs table to identify tables that contain long LONGs. This information is displayed in the MUST_USE_FASTCOPY results column in the Results tab of the SQL Editor:

- YES is displayed for tables with LONG or LONG RAW columns that are larger than 32,760 bytes (long LONGs). These tables must be reorganized with FastCopy (or DBMS_DataPump). (YES is also displayed for empty tables.)
- NO is displayed for tables with LONG or LONG RAW columns that are 32,760 bytes or less in size. These tables can be reorganized with either FastCopy/DBMS_DataPump or SQL (PL/SQL is used).

Please note that the Find Long LONGs table only contains data on tables that have been scanned with the Find Long LONGs procedure.

i Note: To identify tables with long LONGs from reorg wizards, check the **Long LONG Columns** field in Customize Object Allocations windows. This field displays **Yes** for tables that contain long LONGs, **No** for tables that do not contain long LONGs, and **Unknown** for tables that contain LONGs but have not been scanned with the Find Long LONGs procedure. The information displayed is based on data in the Find Long LONGs table.

How to query the Find Long LONGs table

1. Select **File | New Script** to open the SQL Editor.
2. Query the QUEST_SPC_LONG_TABLES table with a SELECT statement as follows:

- To find all records in the Find Long LONGs table, run:

```
Select * from quest_spc_long_tables;
```

- To find records only for tables that contain LONGs > 32K, run:

```
Select * from quest_spc_long_tables where MUST_USE_FASTCOPY = 'YES';
```
 - To find records only for tables that contain LONGS < 32K, run:

```
Select * from quest_spc_long_tables where MUST_USE_FASTCOPY = 'NO';
```
3. View results of the query in the Result tab of the SQL Editor. The MUST_USE_FASTCOPY column identifies which tables contain long LONGs:
- YES is displayed for tables with LONG or LONG RAW columns that are larger than 32,760 bytes (long LONGs). These tables must be reorganized with FastCopy (or DBMS_DataPump). (YES is also displayed for empty tables.)
 - NO is displayed for tables with LONG or LONG RAW columns that are 32,760 bytes or less in size. These tables can be reorganized with either FastCopy/DBMS_DataPump or SQL (PL/SQL is used).

Defragment Tablespaces and Coalesce Free Space

Defragment Tablespaces

Options for defragmenting a tablespace allow you to merge free extents that are scattered throughout a tablespace. These include free extents that are next to each other and free extents that are separated by data extents. After a tablespace is defragmented, all free space is contiguous and coalesced into larger extents.

i Tip: To check the degree of fragmentation in a tablespace, use a Datafile Map. See [Monitor Datafile Storage](#) on page 29 for more information.

The sections that follow cover how to defragment a tablespace during a standard reorganization with Reorg Manager. The basic steps are to include all tablespace objects and select an interim tablespace as the target tablespace. Please note that a defragment option is not displayed in the Reorg Manager interface.

Selecting all objects and the interim option causes a tablespace to be defragmented as follows:

1. All data is moved out of the tablespace on a temporary basis.
2. All objects in the tablespace are dropped.
3. All free space in the tablespace is coalesced.
4. All objects in the tablespace are recreated and repopulated with data.

i Note: Be aware that if objects remain in a tablespace during reorganization, the tablespace is not completely defragmented. Also note that it should not be dropped and recreated. See [Empty a Tablespace Before You Defragment](#) on page 208 for more information.

Defragment with an Interim Tablespace

The interim target tablespace option provides an alternative to reorganizing tables in their current tablespace. It allows you to reorganize them in a different tablespace, something that is useful when the current tablespace is low on free space. During a standard reorganization, this option defragments a tablespace when all objects are included.

First, all table data are copied from the original tablespace to copy tables in the interim tablespace or to space in the filesystem (depending on the data movement option in use). Then all original objects are dropped from the original tablespace at the same time. While data are stored in the interim tablespace or filesystem and the

original tablespace is empty, all free space in the original is coalesced. Afterward, objects are recreated in the original tablespace and data is copied back to that tablespace. (Indexes are automatically dropped and recreated when an interim tablespace is used for their tables. However, the interim tablespace is not used for the indexes.)

To defragment with an interim tablespace

1. In the Object Explorer, select all objects in a tablespace for reorganization. (Hint: Use **SHIFT+click** to select all objects in the Segments grid for the selected tablespace.)
2. With the objects selected, right-click and select **Reorg Manager | Standard Reorg**.
3. In the Target Tablespaces section, select the interim tablespace option for tables. To do this, select **Leave in Original Tablespace | Use Interim Tablespace**. Also select the tablespace to use on an interim basis. Check the free space displayed for each tablespace. The one you select must have enough concurrent free space for all copy tables it will contain.
4. Under Indexes, select either target tablespace option for indexes. (Indexes are automatically dropped and recreated when an interim tablespace is used for their tables. However, the interim tablespace is not used for the indexes.)
5. Click **Next** to proceed to the Scripting Options windows and select options that control various aspects of a reorganization.
6. Proceed through remaining Reorg Manager windows. If the list of objects to reorganize includes individual partitions or subpartitions, a message asks if you want to drop and recreate the entire partitioned objects. Click **Yes** in response. If index partitions or subpartitions are reorganized individually, they remain in the tablespace during reorganization. As a result, it is not completely defragmented.
7. From the Schedule Script Execution window, select an execution option for the reorganization script. If the script uses only SQL, it can be run interactively from Space Manager's SQL Editor or on a scheduled basis by the QSA Server Agent. If the script uses both SQL and FastCopy or only FastCopy, it must be run on a scheduled basis.

Empty a Tablespace Before You Defragment

In most cases, a tablespace is temporarily emptied during reorganization when all of its objects are included and the right combination of data movement and target tablespace options is selected. It is then defragmented.

However, under certain circumstances, objects may remain in a tablespace during a reorganization even when they are included on the object list. This happens when:

- The objects are cluster indexes.
- Index partitions and subpartitions are reorganized individually.
- The objects are not supported for reorganization with either FastCopy/DBMS_DataPump or SQL. An example of such an object is one that is not supported for FastCopy and also contains a LONG column that is too large for SQL.

When objects remain in a tablespace during reorganization, it cannot be completely defragmented.

There are workarounds for moving cluster indexes and index partitions and subpartitions out of a tablespace in order to defragment it. These are described in the sections that follow. However, there are no workarounds for objects that are not supported for either FastCopy/DBMS_DataPump or SQL. See [What is Not Supported for Standard Reorganization](#) on page 55 for more information.

Workarounds for Cluster Indexes

If you want to defragment a tablespace that contains cluster indexes, the workaround for moving them out of the tablespace is to relocate them during reorganization. To do this, select a new tablespace for each cluster index in Reorg Manager's Customize Object Allocations window.

Workarounds for Index Partitions and Subpartitions

If you want to defragment a tablespace that contains individual index partitions or subpartitions, the main workaround is to reorganize the entire partitioned index. The entire partitioned index can be reorganized as a single object no matter where its individual partitions or subpartitions are located in a database.

Normally, index partitions and subpartitions are not moved out of a tablespace when they are reorganized individually. However, two additional workarounds make it possible to reorganize index partitions and subpartitions individually and still defragment a tablespace:

- **Workaround for local and global index (sub)partitions**—For local and global index (sub)partitions that are being reorganized along with their table segments, one workaround is to relocate the index segments during reorg. To do this, first select the Exchange partition reorg method for the table segments in Reorg Manager's Select Partition Reorganization Method window. Then select a new tablespace for each index partition or subpartition in the Customize Object Allocations window. You can move the index segments back to their current tablespace with a second reorg, if needed.
- **Workaround for global index partitions**—For global index partitions that are being reorganized without their table segments, an additional workaround is to use the Move/Rebuild partition reorg method. To do this, select the Move/Rebuild method for the global index partitions in Reorg Manager's Select Partition Reorganization Method window. Then select a new or interim tablespace for each global index partition in the Customize Object Allocations window.

How Live Reorganizations Work

Live reorganizations are designed to restore tables to peak capacity with minimum impact on their availability. When tables are reorganized on a live basis, applications can remain online and users can continue activity against the tables. Data remains available for Data Manipulation Language (DML) statements, including INSERT, UPDATE, and DELETE statements.

i Note: Before you can perform live reorganizations with Space Manager, the LiveReorg option must be licensed. In addition, the QSA Server Agent must be installed, current, and running.

Each live reorganization is performed in a series of stages. During these stages, a copy of the table being reorganized is created. Data from the original table is reorganized as it is inserted into the copy table.

As the copy table takes shape, applications and users can make changes to data in the original table. All changes are captured by QSA, Space Manager's script execution agent. The agent continually posts (copies) these changes to the copy table. Once both tables are completely in sync, the "switch" is made. At this point, the original table is replaced with its reorganized copy table, which can then be accessed as usual.

When the online switch is used, tables remain available throughout an entire live reorganization. When the T-Lock switch is used, the table is unavailable during the switch. Any application trying to access the table during the switch will get a database error. Be sure that your application will handle these errors appropriately. See [Switch Mode](#) on page 62 for more information.

When multiple tables are included in a live reorganization script, if they are not partitioned, Space Manager uses a parallel query to reorganize tables. If the tables are partitioned, there is one switch per table, and reorganization of the next table in the script does not begin until all tasks have been performed for the table ahead of it.

! CAUTION: Reorganization commands must not be modified or removed from a script. If they are changed in any way, a reorganization can fail or unpredictable results can occur. Most commands begin with `QUEST_EXEC`.

Stages of a Live Reorganization

This section describes the stages of a live reorganization. To monitor these stages as they are performed, use the Live Reorg Detail pane of the Scripts/Job Monitor. See [Monitor Live Reorganizations](#) on page 133 for more

information.

Stage 1 — Create copy table

In the first stage of a live reorganization, a structural copy of the original table is created and all existing data is copied from the original table to the copy table:

- If an entire table is being reorganized, the copy table is a structural copy of the original table. All of the original table's data is copied.
- If an individual partition or subpartition is being reorganized, the copy table is a structural copy of the parent-partitioned table, but only data for the original partition or subpartition is copied.

Data is reorganized as it is inserted into the copy table. The copy table uses the storage attributes specified in Reorg Manager or the Partitioning Wizard.

As the copy table is created and existing data are copied, live DML activity can continue against the original table. QSA collects information on live transactions using collection triggers. One trigger is used for most transactions. A second is used for direct UPDATES to LOBs.

The information collected by triggers is stored in a collection table. The table is automatically created in the tablespace being used for LiveReorg objects. This is specified with QSA's LW_TABLESPACE parameter and is identified in documentation as the "LiveReorg tablespace". See [LiveReorg Objects and Their Tablespace](#) on page 221 for more information.

In the third stage of a live reorganization, QSA posts (copies) live transactions to the reorganized copy table based on information in the collection table.

Stage 2 — Create Posting Index

Once the reorganized copy table is in place, the first index is recreated. This index serves as the posting index. The posting index is used when live DML transactions are posted to the reorganized copy table. It can be used for one or two purposes, depending on whether it is a selective or nonselective index:

- Selective posting indexes are used to ensure that rows are uniquely identified so that live transactions are posted to the correct rows. A selective index is one where every index entry points to a single location in table data. The PRIMARY KEY and unique indexes are selective indexes. A nonselective index is one where each index entry points to multiple locations in table data.
- All posting indexes are used to speed up the posting process by increasing the efficiency of searches on the copy table.

What type of index is used as the posting index depends on what is available for a table. Space Manager tries to use the most selective index. How the posting index is chosen is covered under "Posting Indexes".

As the posting index is created on the reorganized copy table, QSA continues collecting live DML transactions.

Stage 3 — Post Live Transactions

Once the posting index and its constraints are in place on the reorganized copy table, QSA begins posting live transactions to the copy table. Posting continues throughout Stage 3.

Additional tasks for this stage consist of recreating all remaining indexes. They can also include recreating constraints, privileges, and comments. Which of these tasks are performed depends on object type and switch style.

How Posting Progresses

1. First, QSA posts the live DML transactions collected during Stages 1 and 2. This allows the copy table to "catch up" with changes made to the original table in these stages.

2. Next, the agent continually collects and posts the live transactions made against the original table as Stage 3 proceeds.
3. Finally, after all tasks for Stage 3 have been performed, the agent posts any last-minute transactions that remain in the collection table. Once the collection table is empty and there are no remaining transactions, the original table and reorganized copy table are in sync and ready for the switch. Up until the switch is made, the agent continues collecting and posting live transactions, keeping the original and copy tables in sync.

How Transactions Are Posted

QSA posts live transactions using a PL/SQL stored procedure and Oracle Call Interface (OCI) calls. The OCI calls are used for data in LONG columns. The procedure and calls read transactions in chronological order, oldest transactions first. See [The Posting Procedure and OCI Calls](#) on page 221 for more information.

Stage 4 — Make the Switch

In the switch stage of a live reorganization, the original table and its indexes are replaced with the reorganized copy table and its indexes. When the online switch is used, the switch is the only task performed during the switch stage. When the T-Lock switch is used, additional tasks are always performed. These depend on whether a table or partition is being reorganized. See [Estimate the Time Needed for a Live Reorganization](#) on page 213 for more information.

Additional Tasks for a T-Lock Switch for Tables

When a table is reorganized with the T-Lock switch, additional tasks for the switch stage include:

- Recreating constraints, triggers and synonyms.
- Recompiling dependencies when the **While the table is protected by T-Lock** option is selected. (Dependencies are recompiled after the switch stage when this option is not selected. This can significantly shorten the switch stage for tables with hundreds of dependencies.)

Additional Tasks for a T-Lock Switch for Partitions

When a partition or subpartition is reorganized with the T-Lock switch, additional tasks for the switch stage include:

- Recreating or rebuilding global indexes on the parent-partitioned table.
- Recreating FOREIGN KEYs that reference the parent-partitioned table.
- Recompiling materialized views.
- Recompiling dependencies when indexes are recreated (instead of being rebuilt) and the **While the table is protected by T-Lock** option is selected. (Dependencies are compiled after the switch when this option is not selected.)
- Dropping a table's temporary ROWID column, which causes its ROWID index to be dropped. This task is performed when a ROWID index is used as the posting index.
 - i Note** When a (sub)partition is reorganized live, triggers, synonyms, and most dependencies are left in place on the parent table. As a result, they do not need to be recreated or recompiled. However, dependencies must be recompiled when indexes are recreated instead of being rebuilt.

How the Switch Begins

How the switch begins depends on the switch option selected for a live reorganization script:

- Automatic switch option—When this option is selected, the switch stage begins as soon as the copy table is ready for the switch.

- User-approval switch option—When this option is selected, the switch stage begins when approval is given.
- Time-window switch option—When this option is selected, the switch stage begins when the time window begins.

If QSA must wait for user approval or a time window, it continues collecting and posting live transactions, keeping the original table and copy table in sync. Once approval is given or a time window begins, the agent posts any live transactions that remain in the collection table and then makes the switch.

How the Switch Is Made

How the switch is made depends on the switch style being used:

- Online switch—The online switch is made by exchanging the original table and copy table. The copy table takes the place of the original. The original takes the place of the copy. Triggers, synonyms, and dependencies, which were left in place on the original, now apply to the copy.
- T-Lock switch—For a table, the T-Lock switch is made by renaming both the original and reorganized copy tables. The reorganized copy is given the name of the original table and takes its place as a result. For a partition or subpartition, the T-Lock switch is made by exchanging the original partition or subpartition with its copy table. The copy table takes the place of the original partition or subpartition in the parent-partitioned table. It thus becomes a partition or subpartition. The original partition or subpartition takes the place of the copy table. (If the original is kept, it is kept as a table.)

Table Availability During the Switch

Table availability during the switch depends on the switch style being used:

- Online switch—When the online switch style is used, the table being reorganized remains available throughout the switch stage.
 - T-Lock switch—When the T-Lock switch style is used, original and copy tables are locked with trigger locks throughout the switch stage. Although write access is prevented, read-only access is available for most of this stage. Normally, the duration of the stage is less than a minute.
- i Note** Just before making the switch, QSA acquires an exclusive lock on the original table. This ensures that there are no open transactions against the table. The lock does not affect the online switch.

Stage 5 — Finish Reorganization

After the switch stage completes, the final tasks for a live reorganization are performed. These tasks always include dropping the temporary LiveReorg objects created by QSA. They can also include dropping the original table. This is dropped when the **Drop original table** option is selected.

Additional tasks may be performed depending on switch style and on how scripting options are set. Reorganization of the next table in a script does not begin until all tasks have been performed for the table ahead of it. If the original table is being dropped, this is the last task for a live reorganization.

Additional Tasks After an Online Switch

After an online switch for a table with a LOB column, additional tasks include dropping the table's temporary materialized view log and AFTER INSERT trigger. These are dropped along with other LiveReorg objects. They are created by QSA for use in collecting direct UPDATES to the LOB column. (When the T-Lock switch is used, the log and trigger are dropped right before the switch.)

Additional Tasks After a T-Lock Switch

After a T-Lock switch, additional tasks can include recompiling dependencies. Dependencies are recompiled after the switch stage when the **While the table is protected by T-Lock** option is not selected. (They are recompiled during the switch stage when this option is selected.)

Additional Tasks After Either Switch

After either type of switch, additional tasks can include validating data against constraints. Post-switch validation occurs when the **Revalidate originally validated constraints** option is selected and the constraints were ENABLE VALIDATE prior to reorganization.

Post-switch validation also occurs for PRIMARY KEYs that serve as the posting index. However, it does not occur when an entire table is reorganized with a T-Lock switch. In this case, data is validated against the PRIMARY KEY before the switch.

- i Note** Reorganization of the next table in a script does not begin until validation for the last is complete. DML activity can continue against a table as constraints are validated.

Estimate the Time Needed for a Live Reorganization

This section provides information that can help you estimate the time required for a live reorganization. As a live reorganization is performed, you can monitor its progress from the Live Reorg Detail pane of the Scripts/Job Monitor. See [Monitor Live Reorganizations](#) on page 133 for more information.

Time from Start of Reorganization to the Switch

The time required from the start of a live reorganization to the start of the switch depends on the size of the original table, the number and size of its indexes, and the level of activity against the table. It also depends on whether statistics are being collected. Recreating a large table and its indexes could require a minimum of several hours.

Time for the Switch

The time required for the switch is very brief. Normally, it takes less than a minute to replace an original table and its indexes with a reorganized copy table and its indexes. When the T-Lock switch is used, time is needed for additional tasks. These tasks are covered under “Stage 4—Make the switch”. The following list identifies factors that determine the amount of time required for additional tasks:

- Handling constraints and dependencies—The time required to recreate constraints and recompile dependencies is determined by how many must be processed. Time requirements can range from a matter of seconds to a matter of hours.
- Recreating global indexes—The time required to recreate global indexes is determined by their number and size.
- Dropping a ROWID index—If a ROWID index is used as the posting index, its size determines the time required to perform the tasks involved in dropping that index.

Time to Finish a Live Reorganization

After the switch stage completes, time is needed to drop the original table, if it is not being kept. Time is also needed to drop the LiveReorg objects created by QSA. Depending on how scripting options are set, more time may be needed for additional tasks. These tasks are covered under “Stage 5—Finish reorganization”.

The following list identifies factors that determine the amount of time required for final tasks:

- Recompiling dependencies—The time required to recompile dependencies is determined by how many must be processed. Time requirements can range from a matter of seconds to a matter of hours.
- Validating constraints—The time required to validate data against constraints depends on table size. Very large tables might need several hours or more. Reorganization of the next table in a script does not begin until validation for the last is complete. DML activity can continue against a table as constraints are validated.

- Dropping tables—Normally, only a matter of seconds is needed to drop an original table and LiveReorg objects. However, it might take minutes or hours if a table contains a large number of extents and is located in a data dictionary managed tablespace.

What to Do When Posting Falls Behind

Under rare circumstances, posting can fall behind or become “stuck” in a repetitive loop so that the switch cannot be made. The first problem can occur due to extremely heavy live activity. The second problem can occur due to system problems such as extremely high CPU usage.

In both cases, the main symptom is that posting sessions are longer than the “catchup” interval set for these sessions. The default is 60 seconds. When posting sessions are longer than the catchup interval, QSA treats this as a sign that there are more live transactions to post. As a result, it does not consider the copy table to be caught up with the original table and does not make the switch.

To determine whether posting has fallen behind or is stuck, check the Scripts/Job Monitor, the monitor’s Live Reorg Detail pane, and the script execution log:

- Scripts/Job Monitor—For both problems, “Posting live modifications” is displayed in the monitor’s Status Message column.
- Live Reorg Detail—For both problems, a value of greater than 60 seconds is displayed in the detail pane’s second Elapsed Time field for Stage 3. When posting is stuck, the value for Post Count continues to increase while the value for Current stays the same.
- Script log—For both problems, a series of session-time entries like the one in the following example are displayed in the script log:

```
session time nnn > 60
```

To resolve either problem, increase the catchup interval for posting sessions. This is set in the CP_OVER_RIDE column of the QUEST_QSA_PDATA table. The value should be set in the row for the current reorganization. Check the REORG_LABEL column to find the name of the table being reorganized. Check the START_TIME column to identify the current reorganization. CP_OVER_RIDE displays 0 for the default value of 60 seconds. The new value should be greater than the last session time displayed in the script log.

Once the catchup interval is changed, it goes into effect for the next posting session. The change is only used for the table currently being reorganized. The default of 60 seconds is used for the next table in a live reorganization script.

i Note Changes to the catchup interval are reported in the execution log for a live reorganization script. The entry for a change reads: “Posting time limit over-ride changed to nnn seconds”.

Posting Indexes

A posting index is used by QSA as it copies live transactions to a reorganized copy table. The main purpose of the index is to guarantee that rows are uniquely identified. Another purpose is to speed up the posting process by increasing the efficiency of searches on the copy table. Sometimes these functions are split between a posting index and QSA’s “all columns” posting method. This happens when a table does not have a selective index and an agent-created ROWID index is not used as the posting index.

i Note: A selective index is one where every index entry points to a single location in table data. The PRIMARY KEY and unique indexes are selective indexes. A nonselective index is one where each index entry points to multiple locations in table data.

Comparison Chart

Feature	All columns	ROWID indexes
Online switch supported	Yes	No

Feature	All columns	ROWID indexes
T-Lock switch supported	Yes	Yes
FastCopy/DBMS_DataPump supported	Yes	No
Partitions and subpartitions supported	Yes	Not by default
Relative posting speed	Slower	Faster
Guaranteed row uniqueness for tables with LOBs or LONGs	No	Yes
Requires DDL during reorg	No	Yes

How Space Manager Chooses a Posting Index

In most cases, Space Manager automatically chooses the index to use as the posting index. It looks at all indexes available for a table and selects the index that ranks highest on the list under “Posting index types and their rankings”.

Index types are ranked in order of most selective to least selective. Space Manager’s first choice among selective indexes is the PRIMARY KEY. Its second choice is a unique index. Its third choice is a non-unique index with a unique constraint.

If a selective index is not available, Space Manager compensates by using one of two special methods for guaranteeing uniqueness of rows. These are the “all columns” posting method and the ROWID posting index:

- All columns—The “all columns” method determines row uniqueness by using all columns in an original table to identify unique rows in its reorganized copy. Space Manager’s default behavior is to use “all columns” for tables that do not contain LOBs, LONGs, XMLTypes, or user-defined datatypes (object types, REFS, and VARRAYs). This method is used in conjunction with a non-selective index or no posting index. The non-selective index is used to speed up posting. First choice is a unique index with all columns nullable. Second choice is non-unique index without a unique constraint.
- ROWID index —A ROWID posting index determines row uniqueness by using the ROWIDs of all rows in an original table to identify unique rows in its reorganized copy. Space Manager’s default behavior is to use a ROWID index for tables that contain LOBs, LONGs, XMLTypes, or user-defined datatypes (object types, REFS, and VARRAYs).

If you want to change Space Manager’s default use of “all columns” and ROWID indexes, you can do this by setting a parameter for QSA. See [Comparison Chart](#) on page 214 for more information.

i Tip To see what type of posting index will be used for a table, check its LW_START_REORG command in a live reorganization or partitioning script.

Important Considerations for Posting Indexes

- When a unique index is used as the posting index, be aware that posting may be slow if the leading column of the UNIQUE KEY constraint is nullable.
- Indexes with disabled constraints are not considered in choosing a posting index.
- Function-based indexes and indexes that contain an object type attribute cannot be used as a posting index. Object type attributes are a feature of user-defined datatypes.
- If a table contains a LONG column larger than 32,760 bytes and does not have a PRIMARY KEY or non-nullable unique index, it can only be reorganized on a live basis if the “all columns” posting method and T-Lock switch are used. The table and all of its datatypes must be supported for FastCopy (or DBMS_DataPump).

Posting Index Types and Their Rankings

1. First choice — PRIMARY KEY

First choice for the posting index is a PRIMARY KEY in the ENABLE VALIDATE state. If a PRIMARY KEY is in the ENABLE NOVALIDATE state and a table has no other suitable posting indexes, a message prompts you to validate the constraint in Reorg Manager or the Partitioning Wizard. The PRIMARY KEY must be in the validate state to guarantee that rows are uniquely identified during posting. See [Prompts to Validate Constraints](#) on page 217 for more information.

PK or PKNV = identifier in LW_START_REORG

PKNV is displayed when a PRIMARY KEY will be temporarily set to NOVALIDATE to speed up the switch. It is reset to VALIDATE after the switch is made. This happens when the online switch is used or when partitions and subpartitions are reorganized individually. (PKNV posting indexes are not used for partitioning scripts.)

2. Second choice — Unique index with NOT NULL column

Second choice for the posting index is a unique index with at least one NOT NULL column. Its NOT NULL constraint must be in the ENABLE VALIDATE state. If the index has a unique constraint, this does not need to be validated. If a table has multiple unique indexes that meet requirements, Space Manager uses the first unique index returned by the query it runs.

UK = identifier in LW_START_REORG

3. Third choice — Non-unique index with a unique constraint

Third choice for the posting index is a non-unique index with a unique constraint that is in the ENABLE VALIDATE state and has at least one NOT NULL column. These requirements are listed below:

- Non-unique index
- Unique constraint that is in ENABLE VALIDATE state and is enforced by the index
- Unique constraint has a NOT NULL column in the ENABLE VALIDATE state

If the UNIQUE KEY constraint is in the ENABLE NOVALIDATE state and a table has no other suitable posting indexes, a message prompts you to validate the constraint in Reorg Manager or the Partitioning Wizard. See [Prompts to Validate Constraints](#) on page 217 for more information. The constraint must be in the validate state in order to guarantee that rows are uniquely identified during posting of live transactions.

UC or UCNV = identifier in LW_START_REORG

UCNV is displayed when a unique constraint will temporarily be set to NOVALIDATE to speed up the switch. It is reset to VALIDATE after the switch is made. This happens when the online switch is used or when partitions and subpartitions are reorganized individually. (UCNV posting indexes are not used for partitioning scripts.)

4. Fourth choice — Unique index with all columns nullable

Fourth choice for the posting index is a unique index in which all of the columns allow null values. This type of index is nonselective and may not contain all of its table's rows. Normally, that would allow the table to contain duplicate rows. However, when this type of index is used as the posting index, the "all columns" posting method is also to guarantee uniqueness of rows during posting.

UI = identifier in LW_START_REORG

5. Fifth choice — Non-unique index without a unique constraint

Fifth choice for a posting index can be either:

A non-unique index without a unique constraint.

or

A non-unique index with a unique constraint that does not have at least one NOT NULL column.

This type of index is nonselective. To guarantee uniqueness of rows, the “all columns” posting method is used along with it.

IX = identifier in LW_START_REORG

6. Sixth choice — No index

Sixth choice is to post without an index. This choice is made for tables that do not have any indexes. To guarantee uniqueness of rows, the “all columns” posting method is used when a posting index is not available.

NONE = identifier in LW_START_REORG

7. Last choice — ROWID index

Last choice is a ROWID index. This uses the ROWIDs of all rows in the original table to guarantee uniqueness of rows. A ROWID index is created during script execution for temporary use. It is dropped during or after the switch, depending on whether a table or (sub)partition is being reorganized. See [ROWID Posting Indexes](#) on page 218 for more information. The reason a ROWID index is the last choice is because of the DDL it requires.

RID = identifier in LW_START_REORG

Prompts to Validate Constraints

Normally, Space Manager automatically chooses the posting index to use for a table. However, in some cases, the Reorg Manager or the Partitioning Wizard prompts you to make decisions regarding the posting index. This happens when the only candidate for the posting index has a PRIMARY KEY constraint or a UNIQUE KEY constraint in the ENABLE NOVALIDATE state (choices one and three under “Rankings for posting index types”).

The prompt occurs in a message raised by the Reorg Manager or the Partitioning Wizard. The message prompts you to validate a constraint so that it can be used for posting. If you do not want to validate a constraint, you can exclude its table from a script, use the “all columns” posting method, or use a ROWID posting index. See [Respond to Checks for a Posting Index](#) on page 114 for more information.

All Columns Posting Method

The “all columns” posting method is automatically used for tables that do not have a selective index and do not contain LOBs, LONGs, XMLTypes, or user-defined datatypes (object types, REFs, and VARRAYs). It is used along with a non-selective posting index (or no posting index) as follows:

- “All columns” is used to ensure that rows are uniquely identified during posting of live transactions.
- The non-selective posting index is used to speed up the posting process. If a table has multiple non-selective indexes, one that ranks highest on Space Manager’s priority list is used. Nonselective indexes are ranked from highest to lowest as follows: 1) unique index with all columns nullable, 2) non-unique index without a unique constraint, and 3) no index.

How the All Columns Method Works

The “all columns” method determines row uniqueness by using all columns in an original table to compare live transactions to rows in its reorganized copy.

The only columns this method does not use in comparing rows are columns with LOBs, LONGs, XMLTypes, and user-defined datatypes (object types, REFs, and VARRAYs).

If the only difference between two rows occurs in one of these columns, the “all columns” method does not detect the difference. Because this could result in duplicate rows in the reorganized table, Space Manager’s default behavior is to prevent use of “all columns” for tables that contain these columns and do not have a selective index. ROWID indexes are used instead.

If you want to allow use of “all columns” for tables with LOBs, LONGs, XMLTypes, and user-defined datatypes (object types, REFs, and VARRAYs), you can do this by setting a Quest Server Agent parameter. You should

only do this if you are certain that rows can be identified uniquely based on columns that do not contain these datatypes. See [Set the Posting Parameter](#) on page 219 for more information.

ROWID Posting Indexes

ROWID indexes are automatically used for tables that do not have a selective index and contain LOBs, LONGs, XMLTypes, and user-defined datatypes (object types, REFs, and VARRAYs). This type of posting index is only used for other tables if you select it in a message prompting you to validate constraints. See [Prompts to Validate Constraints](#) on page 217 for more information.

How ROWID Indexes Work

ROWID indexes determine row uniqueness by using the ROWIDs of all rows in an original table to compare live transactions to rows in its reorganized copy.

The ROWID posting method is faster than the “all columns” method and never allows duplicate rows for tables with LOBs, LONGs, XMLTypes, and user-defined datatypes (object types, REFs, and VARRAYs). However, it does not support FastCopy/DBMS_DataPump or the online switch and it requires that a column and index be created and dropped during reorganization. This DDL can affect reorganization performance in various ways. See [Comparison Chart](#) on page 214 for more information.

If you want to use the “all columns” method instead of ROWID posting indexes for tables with LOBs, LONGs, XMLTypes, and user-defined datatypes (object types, REFs, and VARRAYs), you can do this by setting a Quest Server Agent parameter. You should only do this if you are certain that rows can be identified uniquely based on columns that do not contain these datatypes. See [Set the Posting Parameter](#) on page 219 for more information.

How ROWID Indexes Are Created and Dropped

When a ROWID index will be used for posting, it is automatically created during a live reorganization as follows:

- A ROWID column is added to the copy table as the copy is created. The column is populated with the ROWIDs of all rows in the original table.
- A ROWID index and UNIQUE constraint are created on the ROWID column. The constraint is used to guarantee uniqueness of rows.

ROWID indexes are not permanent. They are created in the LiveReorg tablespace and dropped as follows:

- For tables—When a ROWID is used for a table, the ROWID index is dropped during the switch. Its ROWID column is set to UNUSED at that point and dropped at the end of live reorganization or partitioning script, after all tables have been reorganized.
- For partitions and subpartitions—When a ROWID index is used for a partition or subpartition, both the ROWID index and its ROWID column are dropped during the switch.

After the ROWID column and index are dropped, dependencies may be invalid. (However, they are automatically recompiled the first time they are referenced.) Under rare circumstances, dropping the column may take a long time and affect application performance. Also, dropping the column can sometimes fail with a “Snapshot too old” error.

i Note The LiveReorg tablespace is where temporary LiveReorg objects are created. It is specified with QSA's LW_TABLESPACE parameter.

ROWID Indexes and Partitions

By default, ROWID indexes are only used for entire partitioned tables and nonpartitioned tables. They are not used for individual partitions or subpartitions. This is because when they are used for the latter objects, they require more time during the switch. The additional time is due to how ROWID indexes are dropped for these objects.

If you want to be able to use ROWID indexes for partitions and subpartitions, you need to set a Space Manager parameter. To do this, insert a row in the QUEST_SPC_PARAMETER table, where:

NAME = Reorganization\Method\UseRowidForPartitions

VALUE = 1

Set the Posting Parameter

A QSA parameter controls whether the “all columns” posting method or ROWID posting indexes are used for tables without selective indexes that contain LOBs, LONGs, XMLTypes, or user-defined datatypes (object types, REFS, and VARRAYs).

The parameter is called **ALLOW_LONG_WO_UNIQUE** and is specified in the VALUE column of the QUEST_EXEC_PARAMETER table:

- When the parameter is set to NO, a ROWID posting index is used for tables that contain LOBs, LONGs, XMLTypes, and user-defined datatypes (object types, REFS, and VARRAYs).
- When the parameter is set to YES, the “all columns” posting method is used for tables that contain LOBs, LONGs, XMLTypes, and user-defined datatypes (object types, REFS, and VARRAYs). It is used in conjunction with a non-selective posting index or no index. The nonselective index can be a unique index with all columns nullable or a non-unique index without a unique constraint.

By default, ALLOW_LONG_WO_UNIQUE parameter is set to NO. It can be changed while the agent is running and the database is online. It should only be changed to YES if you are certain that rows can be identified uniquely based on columns that do not contain LOBs, LONGs, XMLTypes, and user-defined datatypes (object types, REFS, and VARRAYs).

Collection Triggers and the Collection Table

To collect information on live transactions, QSA uses triggers. These are designed to capture the INSERTs, UPDATEs, and DELETEs that are made against the original table as a live reorganization proceeds. Information on live transactions is stored in a collection table. During the posting stage, the agent uses this information to post live transactions to the reorganized copy table.

The collection table is called QUEST_QSA_LWDATA<obj#>. The last part of the table name is the object ID of the original table. The collection table is created in the LiveReorg tablespace. After the switch from original table to copy table is made, the collection table is automatically dropped along with other LiveReorg objects.

Collection Trigger for Most Transactions

To collect information on most live transactions, the agent uses an AFTER trigger called QUEST_QSA_LWRT<obj#>. The trigger fires after each INSERT, UPDATE, and DELETE against the original table. The last part of the trigger name is the table’s object ID.

The trigger is created when a read consistency point is established for the original table. Read consistency is established just before original data is copied to the newly created copy table. The agent uses the read-consistency point to distinguish between original data and live transactions.

Collection Trigger for Direct UPDATES to LOBs

To collect information on direct UPDATES to LOB columns, the agent uses an AFTER INSERT trigger on a materialized view log. The trigger and log work together as follows:

1. As live transactions are made against the table being reorganized, any rows for LOB updates are inserted into the log. They are committed before they are inserted.
2. When a direct UPDATE to a LOB is recorded in the log, the trigger fires and collects information on the UPDATE from the original table.
3. The trigger stores information on the UPDATE in the collection table.

The trigger and materialized view log are created when a LOB is detected in the table being reorganized. The log is created on the table. Both the trigger and log are stored in the LiveReorg tablespace. The owner of the table needs quota for this tablespace. See [Space for LiveReorg Objects](#) on page 223 for more information.

When the online switch is used, the trigger and log are dropped along with other LiveReorg objects after the switch. When the T-Lock switch is used, the trigger and log are dropped right before the switch.

The trigger used for direct UPDATES to LOBs is named QUEST_QSA_LWRMT<obj#>. The last part of the trigger name is the object ID of the table being reorganized. The materialized view log is named MLOG\$_TABLENAME. The last part of the log name is the name of the table being reorganized.

i Important When a table with a LOB has a materialized view log, the agent drops the log and replaces it with its own materialized view log. If a (sub)partition is being reorganized, the log is dropped from the parent table. You need to recreate the table's log after reorganization is complete.

What to Do If a Trigger Error Occurs

Errors can sometimes occur during execution of a collection trigger. If a trigger error occurs, it is recorded in the QUEST_QSA_ERR table. A live reorganization then fails during posting. At that point, this error is displayed in the Scripts/Job Monitor: "Aborted QSA-20391 ORA-20777 during internal SQL."

The error is also displayed in the script execution log along with an Oracle trigger error, a Quest Server Agent error, or both. For example, if a collection trigger could not insert a row in the collection table, the following is displayed in the script log:

```
Collection trigger has failed (20777)
Trigger Oracle error is - 103
Error msg: Failed LW collector trigger
```

To recover from a trigger error, resolve the problem that caused the error. Then restart execution of the live reorganization script using the Restart function in the Scripts/Job Monitor. See [Restart Scheduled Scripts](#) on page 124 for more information.

What to Do If a Materialized View Log Error Occurs

An error sometimes occurs when a row cannot be inserted in the materialized view log used for collecting direct UPDATES to LOBs. The error is: "ORA-12096: Error in materialized view log on 'string'.string'." Oracle returns this error to the application that is accessing the table being reorganized. The error is not displayed in Space Manager's Script/Jobs Monitor and does not cause a reorganization to fail.

The error can occur when the LiveReorg tablespace does not have enough space for the materialized view log to grow. It can also occur during large, long-running open transactions where thousands of rows have been inserted, updated, or deleted without a COMMIT statement being issued. (To keep the log from running out of space, QSA periodically deletes all rows from the log. See [Space for Materialized View Logs](#) on page 223 for more information.)

To identify the cause of a materialized view log error, check additional error messages in the stack. If a space shortage caused the error, you can recover by increasing the amount of free space in the LiveReorg tablespace. (You can use Tablespace Properties to do this.) Transactions can then continue against the table being reorganized. QSA automatically continues the live reorganization from the point at which it was interrupted.

If you need to discontinue a live reorganization after a materialized view log error occurs, use the following steps to recover from the error; you can then perform the reorganization at another time:

1. Cancel the live reorganization script from the Script/Jobs Monitor.
2. Drop the materialized view log on the table that was being reorganized with the following statement (this also drops the trigger on the log); TABLENAME is the name of the original table:

Drop materialized view log on owner.table_name;
3. Drop LiveReorg objects and the collection trigger for the table that was being reorganized. Objects must be dropped in the following order; the trigger must be dropped first (obj# is the object ID of the original table):

- a. Drop trigger QUEST_QSA_LWRT<obj#>;
 - b. Drop table QUEST_QSA_LWDATA <obj#>;
 - c. Drop procedure QUEST_QSA_LWRP<obj#>;
 - d. Drop sequence QUEST_QSA_LWSES<obj#>;
 - e. Drop sequence QUEST_QSA_LWTNUMSEQ;
 - f. Drop table QUEST_QSA_LWSES<obj#>;
 - g. Drop table QUEST_QSA_LWLONG<obj#>; (this only exists if table being reorganized contains a LONG column)
4. When you are ready to retry the live reorganization, generate a new script for the table that was being reorganized when the materialized view log error occurred. If you want to reorganize any tables that came after that table in the old script, include those tables in the new script.
 5. If a lack of free space caused the materialized view log error, increase the amount of free space in the LiveReorg tablespace or select a tablespace with more free space to serve as the LiveReorg tablespace.
 6. Run the new live reorganization script.

The Posting Procedure and OCI Calls

To post live transactions to reorganized copy tables, QSA uses a PL/SQL stored procedure and Oracle Call Interface (OCI) calls:

- **Stored procedure**—This is used to post live transactions for all columns except LONG columns. The procedure is called QUEST_QSA_LWRP<obj#>. The last part of the procedure name is the object ID of the original table. The stored procedure retrieves information on transactions from the QUEST_QSA_LWDATA<obj#> collection table.
- **OCI calls**—These are used to post live transactions for LONG and LONG RAW columns. OCI calls retrieve information on transactions from the QUEST_QSA_LWLONG<obj#> table.

When a table contains a LONG column, changes to non-LONG columns are posted first with the stored procedure. Any changes needed for the LONG column are posted afterward with OCI calls.

Each execution of the stored procedure is called a “posting session”. If OCI calls are executed after the stored procedure is run, they are considered to be part of the posting session.

Both the stored procedure and OCI calls retrieve transactions in chronological order (oldest transactions first).

- i Note** The reason OCI calls are used for LONG columns is to ensure that data of any size is successfully posted. LONG values larger than 32,760 bytes are not supported for PL/SQL. If a PL/SQL stored procedure were used to post a non-supported LONG value, data would be truncated.

LiveReorg Objects and Their Tablespace

QSA creates several objects for temporary use during a live reorganization. These include the collection table used for storing live transactions and the table used for storing posting statistics. One set of objects is created for each table in a live reorganization script. A table’s object ID is appended to the names of most LiveReorg objects.

Temporary LiveReorg objects are used in addition to QSA’s permanent objects. The names of temporary objects begin with QUEST_QSA_LW%. The names of permanent objects begin with QUEST_QSA_% or QUEST_EXEC_%.

All temporary LiveReorg objects are created in the LiveReorg tablespace. This is specified with QSA’s LW_TABLESPACE parameter. (If a ROWID index is used as the posting index, it is also created in this tablespace.)

QSA's permanent objects are stored in another tablespace. This is specified during first-time installation of the agent.

LiveReorg objects are created at the beginning of a live reorganization. Most are dropped from the LiveReorg tablespace after the switch from original table to reorganized copy table. If the T-Lock switch is used for a table with a LOB, the materialized view log and trigger created for that table are dropped right before the switch.

i Important In most cases, you should not drop or change objects used by QSA. If a live reorganization script fails, leave LiveReorg objects in place while you resolve the problem that caused the failure. The tables will be used when the script is restarted.

A Look at LiveReorg Objects

This section provides a list of the temporary LiveReorg objects that are created by QSA. It also identifies two permanent tables that are used by the agent during live reorganizations:

- **QUEST_QSA_LWDATA<obj#>**—This table is used for storing information on live transactions. The last part of its name is the object ID of the table being reorganized. (This is a temporary LiveReorg object.)
- **QUEST_QSA_LWLONG<obj#>**—This table is used for storing information on posted live transactions that require changes to a LONG column. The last part of its name is the object ID of the table being reorganized. The LWLONG table is only created for tables with a LONG column. (This is a temporary LiveReorg object.)
- **QUEST_QSA_LWIX<obj#>**—This index is used for the collection table, QUEST_QSA_LWDATA. It is created on the transaction number column, Q_TN. The last part of the index name is the object ID of the table being reorganized. (This is a temporary LiveReorg object.)
- **QUEST_QSA_LWSES<obj#>**—This table is used for storing statistics for each posting session that occurs during a live reorganization. The last part of the table name is the object ID of the table being reorganized. (This is a temporary LiveReorg object.)
- **QUEST_QSA_LWRT<obj#>**—This AFTER trigger is used to collect most information on most live transactions. The last part of the trigger name is the object ID of the table being reorganized.
- **QUEST_QSA_LWRMT<obj#>**—This AFTER INSERT trigger is used to collect information on direct UPDATES to LOB columns. It is used in conjunction with the materialized view log, QUEST_QSA_LWRMT<obj#>. The last part of the trigger name is the object ID of the table being reorganized.
- **MLOG\$_TABLENAME**—This materialized view log is used in conjunction with the QUEST_QSA_LWRMT<obj#> trigger to detect and collect direct UPDATES to LOB columns. The last part of the log name is the name of the table being reorganized.
- **QUEST_QSA_MVPKG**—This package is used for inter-process messaging during live reorganizations of tables with LOBs.
- **QUEST_QSA_PDATA**—This table contains persistent information for each live reorganization performed in a database. If a table undergoes multiple live reorganizations, QUEST_QSA_PDATA contains a record for each one. A record is created as soon as reorganization begins. It is preserved whether or not reorganization completes. Information in this table includes key statistics such as posting rates for INSERTs and DELETEs. (This is a permanent QSA object.)
- **QUEST_QSA_ERR**—This table is used for recording errors that occur during execution of collection triggers. (This is a permanent QSA object.)

i Note The “LW” displayed in the names of LiveReorg objects stands for “lightweight”. The collection and posting processes associated with these objects are lightweight. All are designed to have low overhead and minimal impact.

A Look at the LiveReorg Tablespace

The tablespace used for LiveReorg objects is specified with QSA's LW_TABLESPACE parameter. The LiveReorg tablespace must be specified when you install the agent. It can be changed at any time from the QSA

Installer (through the Space Manager client) or from the QUEST_EXEC_PARAMETER table. See [QSA Server Agent Parameters](#) on page 189 for more information.

If you try to exit from the QSA Installer without specifying a LiveReorg tablespace, a “Required parameter is not set” message displays. If you delete the parameter or its value from the QUEST_EXEC_PARAMETER table, live reorganizations fail with this error: “QSA-20391: ORA-NNNNN during internal SQL”.

It is recommended that you create or reserve a tablespace for exclusive use by LiveReorg objects. Dedicating a tablespace to LiveReorg objects ensures that the space they need will not be used by other objects. If you create a LiveReorg tablespace before installing the agent, that tablespace will be available for selection during the installation process.

i Note LiveReorg tablespace must have a standard name, one that does not require double quotes.

To learn how to modify the LW_TABLESPACE parameter, see [Specify Parameters for the QSA Server Agent](#).

Space for LiveReorg Objects

The tablespace used for LiveReorg objects must have sufficient space for creating these objects. It must also have sufficient space for the objects to grow during a live reorganization.

At a minimum, the LiveReorg tablespace should have a datafile that is at least 50 megabytes in size. The datafile should contain at least 20 megabytes of contiguous free space.

How much additional space is needed as a live reorganization proceeds depends on row length for the collection table, total size of the columns in the posting index, and number of live transactions made against the original table during creation of the copy table. If additional space is needed by LiveReorg objects, it is allocated in five-megabyte extents.

i Tip To see how much free space is available in the LiveReorg tablespace, select it in Space Manager’s Explorer. Then check **Freespace** fields at the top of the Explorer.

Space for Materialized View Logs

If the table being reorganized contains a LOB column, the LiveReorg tablespace needs additional space for the materialized view log that is created for the table. The log doubles the amount of free space required.

The materialized view log is used in conjunction with an AFTER INSERT trigger to collect direct updates to the LOB. As live transactions are made against the original table, rows for LOB updates are committed and then inserted into the log.

To keep the log from running out of space as rows are inserted, QSA periodically deletes all rows from the log. It does this using a process that runs every five seconds. If a row cannot be inserted in the log, Oracle returns the following error to your application: “ORA-12096: Error in materialized view log on “string”.”string”. See [What to Do If a Materialized View Log Error Occurs](#) on page 220 for more information.

Space Checks for the LiveReorg Tablespace

Prior to script generation, Space Manager checks to see whether the LiveReorg tablespace has enough space for creating LiveReorg objects. If a table has a LOB column, Space Manager also checks to see whether the tablespace has enough space for creating a materialized view log on the table. A space check of the LiveReorg tablespace is made along with space checks of target tablespaces. See [Review Space Usage](#) on page 91 for more information.

If the LiveReorg tablespace does not have sufficient space for creating LiveReorg objects, warnings are displayed in Reorg Manager’s Review Space Usage window or the Partitioning Wizard’s Space Usage Summary tab. In addition, a message may be raised. To provide the free space needed, you can use Tablespace Properties to add datafiles or increase datafile size. You can leave Reorg Manager or the Partitioning Wizard open while you do this.

If you do not provide the space needed for creating LiveReorg objects, a live reorganization will fail with this error: “QSA-20394: Could not create temporary QSA object in <LW_TABLESPACE>”. If the LiveReorg

tablespace does not have space for a materialized view log, your application will receive the ORA-12096 error. However, reorganization will not fail.

i Note Space checks consider the space needed for creating LiveReorg objects. They do not consider additional space that might be needed for object growth. If a live reorganization fails because a LiveReorg object cannot grow, a collection trigger error is displayed in the script log.

Quota Checks for the LiveReorg Tablespace

During script generation, Space Manager checks to see whether the owners of tables with LOBs have quota for the LiveReorg tablespace. This is needed for creating the materialized view log used in collecting direct UPDATES to LOBs.

If an owner does not have quota, Space Manager raises a warning right after generating a script. You can avoid the warning and automatically grant quota by leaving the Script Quota changes scripting option selected (it is selected by default). You can ensure that a script will not fail due to insufficient quota by selecting the Grant unlimited quota scripting option. Quota that did not exist prior to reorganization is revoked afterward.

Calculate Space for LiveReorg Objects

To estimate how much space will be needed to create and grow LiveReorg objects during a live reorganization, use the following steps:

1. Calculate “total live activity” to be made against the original table. The formula to use is: Total INSERTs + DELETEs + (UPDATES x 2).
2. Calculate “collection table size” for QUEST_QSA_LWDATA. The formula to use is: Row length of 55 bytes + total size of all columns in the posting index.
3. Calculate “collection table space usage”. The formula to use is: “Total live activity” x “collection table size”.
4. If the table to be reorganized contains a LONG column, calculate “LONG collection table size” for QUEST_QSA_LWLONG. The formula to use is: Total INSERTs + UPDATES x row length of 45 bytes.
5. Calculate “total space usage”. The formula to use is: “Collection table space usage” + “LONG collection table size” rounded up to the nearest 5 MB + minimum requirement of 5 MB. (Additional space is allocated in five-megabyte extents.)
6. If the table being reorganized contains a LOB column, double the calculation made in Step 5.

Storage Values for LiveReorg Objects and Their Tablespace

Some LiveReorg objects are sized with storage values specified by QSA. These objects are the collection table and the table that stores changes for LONG columns.

Other LiveReorg objects are sized with default storage values for the LiveReorg tablespace. These objects are the statistics table, the index on the collection table, and the materialized view log that is created for tables with LOB columns.

Default storage values for the LiveReorg tablespace should be set as follows:

- Storage for locally managed tablespace with UNIFORM allocation type:

INITIAL	1 MB
NEXT	1 MB
MAXEXTENTS	UNLIMITED
PCTINCREASE	0

- Storage for dictionary managed tablespace:

INITIAL	1 MB
NEXT	1 MB

MINIMUM EXTENT size	1 MB
MAXEXTENTS	UNLIMITED
PCTINCREASE	0

LiveReorg Commands

This section describes most of the commands that are unique to live reorganizations. There is one set of LiveReorg commands for each table included in a script.

Most LiveReorg commands begin with `QUEST_EXEC`. For example, the first LiveReorg command for a table looks like this:

```
QUEST_EXEC.LW_START_REORG
```

QSA processes LiveReorg commands and passes the SQL statements in a script to Oracle.

- ! **CAUTION: Reorganization commands must not be modified or removed from a script. If they are changed in any way, a reorganization can fail or unpredictable results can occur. Most commands begin with `QUEST_EXEC`.**

LW_START_REORG

This command is used at the start of a live reorganization to provide QSA with basic information. The first parameter in the command specifies a label that uniquely identifies the reorganization process for a certain table. The reorganization label consists of owner name, object name, and a random 10-digit number:

```
OWNER_NAME.OBJECT_NAME.NNNNNNNNNN
```

The `LW_START_REORG` command also identifies the name of the table being reorganized; its restart number (uniquely identifies the table and its commands during script restart); its object type (table, partition, or subpartition); the switch style being used; name and type of the posting index being used. Posting index types are identified as follows:

- **PK**—This is displayed when a PRIMARY KEY will be used as the posting index.
- **PKNV**—This is displayed when a PRIMARY KEY will be used as the posting index and temporarily set to NOVALIDATE to speed up the switch. It is reset to VALIDATE after the switch is made. This happens when the online switch is used or when partitions and subpartitions are reorganized individually. (PKNV posting indexes are not used for partitioning scripts.)
- **UK**—This is displayed when a UNIQUE index with at least one NOT NULL column will be used as the posting index.
- **UC**—This is displayed when the posting index will be a non-unique index with a unique constraint that is in the ENABLE VALIDATE state and has at least one NOT NULL column.
- **UCNV**—This is displayed when the posting index will be as described above and the unique constraint will temporarily be set to NOVALIDATE to speed up the switch. It is reset to VALIDATE after the switch is made. This happens when the online switch is used or when partitions and subpartitions are reorganized individually. (UCNV posting indexes are not used for partitioning scripts.)
- **UI**—This is displayed when a unique index with all columns nullable will be used as the posting index.
- **IX**—This is displayed when the posting index will be either a non-unique index without a unique constraint or a non-unique index with a unique constraint that does not have at least one NOT NULL column.
- **IOT**—This is displayed when the table being reorganized is an index organized table.
- **NONE**—This is displayed when posting will be done without an index. In this case, the “all columns” posting method is used to guarantee uniqueness of rows during posting.
- **RID**—This is displayed when a ROWID index will be used as the posting index.

LW_START_PQ_REORG

This command uses a parallel query to copy tables at the start of an *Online* live reorganization if the tables are not partitioned. This allows Space Manager to create a multi-threaded table copy.

BEGIN_COPY

This command tells QSA to begin copying all original data from the original table to the copy table. Data is reorganized as it is inserted into the copy.

The command used for a partition is `BEGIN_COPY_PARTITION`. The command used for a subpartition is `BEGIN_COPY_SUBPARTITION`.

END_COPY

This command tells QSA that copying of original data is complete.

BEGIN_POSTING

This command tells QSA to begin posting live transactions to the reorganized copy table. First, the agent posts the live transactions collected while the copy table and posting index were being created. Then, it continually posts the live transactions made as reorganization proceeds.

The command used for a partition is `BEGIN_POSTING_PARTITION`. The command used for a subpartition is `BEGIN_POSTING_SUBPARTITION`.

PREPARE_TO_SWITCH

This command tells QSA to get ready for a T-Lock switch from original table to reorganized copy table. Just before the switch is made, the agent applies a T-Lock to the original table. (The `CREATE_TLOCK` command applies a T-Lock to the reorganized copy table.)

The command used for a partition is `PREPARE_TO_SWITCH_PARTITION`. The command used for a subpartitions is `PREPARE_TO_SWITCH_SUBPARTITION`.

ONLINE_PREPARE

This command tells QSA to get ready for an online switch from original table to reorganized copy table.

ONLINE_SWITCH

This command tells QSA to make an online switch from original table to reorganized copy table.

ONLINE_COMPLETE

This command tells QSA that the switch is complete for a table being reorganized with the online switch.

SWITCH_COMPLETE

This command tells QSA that the switch is complete for a table being reorganized with the T-Lock switch.

CREATE_TLOCK

This command applies a T-Lock (trigger lock) to the reorganized copy table during a reorganization where the T-Lock switch is used. (A T-Lock is applied to the original table by the PREPARE_TO_SWITCH command.)

TLOCK_AUTOLIST_ADD

Adds a table to the list of tables that are T-Locked for live reorgs that use T-Locks. When Space Manager creates a T-Lock on the original table, these tables are also T-Locked. If the T-Lock on any of the tables fail, all T-Locks are rolled back and the script terminates with an error.

TLOCK_LIST_ADD

Adds a table to the list of tables that are T-Locked when the TLOCK_LIST_EXEC command executes.

TLOCK_LIST_EXEC

Applies a T-Lock to the list of tables that were added using TLOCK_LIST_ADD. If the T-Lock on any of the tables fail, all T-Locks are rolled back and the script terminates with an error.

TLOCK_DROP_LIST

Drops the list of tables that were added using TLOCK_LIST_ADD.

TLOCK_DROP_AUTOLIST

Drops the list of tables that were added using TLOCK_AUTOLIST_ADD.

DROP_TLOCK

This command removes the T-Lock (trigger lock) from the reorganized copy table during a reorganization where the T-Lock switch is used. (The T-Lock on the original table is dropped when the original table is dropped.)

LW_FINISH_REORG

This command tells QSA that a live reorganization is complete. It includes the reorganization label specified in the LW_START_REORG command.

Global Operations for Space Manager

Space Manager and the QSA Server Agent are globalized applications that support various locales, languages, and character sets. Supported locales include Japanese and Western European locales. Supported character sets include multibyte character sets and ASCII. (QSA is Space Manager's script execution engine.)

Space Manager and QSA must be run in an environment where the same or compatible locales are used for your Oracle database, your Oracle client, your database server operating system, and your client computer operating system. All must use the same character set or compatible character sets.

Please note that although Space Manager can be run for various locales, it is not localized for any particular locale. Also note that it supports single-language-at-a-time (SLAAT) operation as opposed to multi-locale operation.

This section identifies the ways in which Space Manager supports locale settings for a database environment:

- Oracle names—All modules display Oracle names in the characters for your client computer character set. For example, if your computer operating system uses a Japanese character set, names for databases, datafiles, tablespaces, objects, and owners are displayed in Japanese characters. The QSA Installer displays database names in the characters for your database character set. It allows you to specify path names in the characters for your database server character set.
 - Datafile and tablespace names—Tablespace Properties allows you to enter datafile and tablespace names in the characters for your client computer character set.
 - Script and folder names—All windows for storing and scheduling scripts allow you to enter folder, script, and script owner names in the characters for your client computer character set.
 - Script execution—All scripts can be generated and run for data encoded in your database character set. The scripts in question include reorganization scripts. QSA preserves data as it is encoded prior to script execution.
 - SQL Editor—The editor displays query results in the characters for your client computer character set. It also allows you to enter Oracle names in these characters. See [Use the SQL Editor for a Certain Locale](#) on page 232 for more information.
 - Scripts/Job Monitor—The monitor displays folder, script, and script owner names in the characters for your client computer character set. Status messages and script statements are displayed in U.S. English (US7ASCII) characters. Within script statements, object names, object owner names, and tablespace names are displayed in the characters for your client computer character set.
 - Date and time display—All modules display dates and times in the formats for the connecting user's locale. User locale settings are specified with Windows regional settings on the client computer.
 - Messages and field labels—All modules display messages and field labels in U.S. English characters (US7ASCII).
- i Note** Certain languages are not displayed correctly by Space Manager. These include bidirectional languages such as Hebrew and Arabic. They also include complex script languages such as Hindi and Thai.

How Space Manager Uses Locale Settings

Space Manager and QSA automatically use locale settings for:

- Your database
- The Oracle client being used by Space Manager
- The operating system on your Space Manager client computer

The sections that follow describe how these settings are used.

How Server-side Locales Are Used

On your database server, QSA uses locale settings for your database to communicate with the database during script execution. The settings in question are for language, territory, and character set. These are specified with the NLS_LANGUAGE, NLS_TERRITORY, and NLS_CHARACTERSET database parameters when a database is created.

QSA does not use locale settings for the database server operating system. However, it reports these settings in log files. See [How Space Manager Reports Locale Settings](#) on page 231 for more information.

Multibyte Character Sets and QUEST_EXEC Commands

When a database uses a multibyte character set such as JA16SJIS or JA16EUC, QSA converts its QUEST_EXEC commands to UNICODE during script execution. All names in the commands are also converted.

QSA does not convert commands for databases that use a single-byte or UNICODE character set such as AL24UTFFSS, UTF8 or AL32UTF8. QSA reports on UNICODE translation in log files. See [How Space Manager Reports Locale Settings](#) on page 231 for more information.

i Note If an object name or username contains a character that cannot be converted to UNICODE, script execution fails with the following error: “QSA-20395—Could not parse script”. For assistance in recovering from the error, contact Quest Software Support.

How Client-side Locales Are Used

On your client computer, Space Manager uses the character set for the Oracle client, the character set (ANSI code page) for the operating system, and the locale for the connecting user—

- Oracle client—Space Manager uses the character set for your Oracle client to transfer data between the client and your database. All data sent to or received from the client are encoded in this character set.
- Client operating system—Space Manager uses the character set for your client computer operating system to display names and data.
- Connecting user—Space Manager uses the locale for the connecting user to display numbers, currencies, times, and dates.

Requirements and Recommendations for Global Operation

Space Manager and QSA do not need to be configured for global operation. However, your database environment must be configured so that it meets requirements for Space Manager and QSA.

One requirement is that locale settings for your database environment must be compatible. All of the following must use compatible character sets:

- The database
- The database server operating system
- The Oracle client being used by Space Manager
- The operating system on your Space Manager client computer

Compatibility among character sets ensures that information is transferred correctly between Space Manager and your database. It also ensures that Space Manager and QSA function correctly.

Other requirements apply to installation of QSA. For example, in some cases, ASCII characters must be used for usernames, passwords, and file paths during agent installation.

Server-side Recommendations

On your database server, the operating system character set should be compatible with your database character set. Ideally, it should be the same as or a superset of your database character set. If the database character set contains characters that are not in the operating system character set, errors might occur, including errors that cause script execution to abort. When a script fails due to a mismatch in characters, the script execution log shows an error such as “invalid multibyte character” or “malformed script”.

Check Database and Operating System Character Sets

To determine which character set is being used by your database, check the value for the NLS_CHARACTERSET parameter. This is specified when a database is created.

To determine which locale and character set are being used by a UNIX or LINUX database server operating system, type LOCALE at the command-line prompt. To determine which locale and character set are being used by a Windows operating system, check Windows regional settings.

Client-side Requirements

On your client computer, requirements for global operation of Space Manager are as follows:

- The character set for your Oracle client and the character set for the operating system (Windows ANSI code page) must be compatible with your database character set. An example of a compatible combination is:

Oracle client character set = JA16SJIS
Client OS code page = 932
Database character set = JA16SJIS
 - The character set for your Oracle client and the character set for the operating system must be compatible. (When the operating system uses Japanese Windows, the Oracle client must use JA16SJIS as the character set.)
 - The character set for the operating system and the locale for the connecting user must be compatible. If they are not compatible, characters may not display correctly in the Explorer, the SQL Editor, and selection criteria grids.
- i Important** If you are running multiple Space Manager clients for the same database, all client character sets must be compatible with each other and with your database character set.

Check the Oracle Client Character Set

To determine which character set your Oracle client is using, check the value for its NLS_LANG parameter. This can be set with the client's HOME key in the Windows registry, with a Windows environment variable, or with both the HOME key and an environment variable:

- HOME key only—If NLS_LANG is set only in the client's HOME key in the Windows registry, check the HOME key to see which character set your Oracle client is using.
- Environment variable only—If NLS_LANG is set only as a Windows environment variable, check the Windows System Properties window to see which character set your Oracle client is using.
- Both—If NLS_LANG is set in both ways, check the Windows System Properties window to see which character set your Oracle client is using. The setting for the environment variable overrides the client HOME key.

The NLS_LANG parameter has three arguments—language, territory, and character set. These are specified in the following format:

```
NLS_LANG = language_territory.charset
```

The three arguments for NLS_LANG can be specified in different combinations. For a Japanese locale, parameter arguments might be specified as follows—

```
NLS_LANG = JAPANESE_JAPAN.JA16SJIS
```

For an American locale, parameter arguments might be specified as follows:

```
NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252
```

Check the Client Computer Character Sets

To determine which character sets are being used for your Space Manager client computer, check system language and user locale in Windows regional settings:

- System language—The language selected for the client computer operating system controls which character set (code page) is used. On Windows 2000 computers, system language is specified in the General tab of the Regional Options window. On Windows XP computers, it is specified on the Advanced tab of the Regional and Language Options window using the option for non-Unicode programs.

- User locale —The locale selected for the connecting user controls which formats are used for numbers, currencies, times, and dates. On Windows 2000 computers, user locale is specified on the General tab of the Regional Options window. On Windows XP computers, it is specified on the Regional Options tab of the Regional and Language Options window. User locale must be compatible with system language for characters to display correctly in the Explorer, the SQL Editor, and selection criteria grids.

Before you can use a character set, the Windows ANSI code page for that character set must be installed. When a code page is installed, its check box is selected in the **Code page conversion tables** list in regional settings windows.

How Space Manager Reports Locale Settings

Space Manager reports locale settings in reorganization scripts, script logs, and the QSA log. All can be sent to Quest Software Technical Support.

Information in scripts and log files can be used by Support to troubleshoot any locale-related problems that occur during script execution. Information in saved maps allows Support to configure its environment to match yours so that database, object, and owner names are correctly displayed.

Locale Information in Reorganization Scripts

Locale information for your Oracle client and database are reported in the headers of Space Manager reorganization scripts. The settings reported for both are language, territory, and character set. The settings displayed are those that were current at the time a script was generated.

Locale Information in Script and QSA Logs

Locale information for your database and database server operating system is reported in script execution logs and in QSA's log (qexecd.log). This information can help you determine whether locale settings for the two are compatible. Ideally, the operating system character set should be the same as or a superset of the database character set.

The information displayed in script logs and in QSA's log is as follows:

- NLS_LANG—This shows the database language, territory, and character set being used by QSA to connect to and communicate with your database. These values are specified with the NLS_LANGUAGE, NLS_TERRITORY, and NLS_CHARACTERSET parameters when a database is created. QSA records parameter values in its configuration file (QEXECD.CONF). It then retrieves them from this file during script execution.
- LC_ALL—This shows any setting for the LC_ALL environment variable on a UNIX or Linux database server. When LC_ALL is not set, the LC_ALL field is either blank or shows a value of NULL. A value for LC_ALL overrides any values for LC_CTYPE and LANG.
- LC_CTYPE—This shows any setting for the LC_CTYPE environment variable on a UNIX or Linux database server. When LC_CTYPE is not set, the LC_CTYPE field is either blank or shows a value of NULL. A value for LC_CTYPE overrides any value for LANG.
- LANG—This shows any setting for the LANG environment variable on a UNIX, Linux, or Windows database server. When LANG is not set, the LANG field is either blank or shows a value of NULL.
- Unicode translation—This indicates whether QUEST_EXEC commands are being converted to UNICODE during script execution. QSA converts commands for databases that use a multibyte character set such as JA16SJIS or JA16EUC. It does not convert commands for databases that use a single-byte or UNICODE character set such as AL24UTFSS, UTF8 or AL32UTF8. The message is displayed in one of two ways:
 - Unicode translation is language_territory.charset—This is displayed when QSA must convert QUEST_EXEC commands to UNICODE. The message identifies the character set into which commands are being converted. The message also identifies the language and territory being used for conversion.

- Unicode translation will not be used for this database—This is displayed when QSA is not converting QUEST_EXEC commands to UNICODE. In this case, the database character set is used for commands. The message identifies the character set, language, and territory for the database.
- Unicode conversion size—This shows the maximum size for the conversion buffer. The default is 4000 bytes, which should be sufficient in most cases. (Buffer size is displayed even QUEST_EXEC commands are not converted to UNICODE. However, a conversion buffer is not used.)
- Unicode database character set is—This shows language, territory, and character set for the database. If UNICODE translation is being performed, the character set displayed in this message is different from the character set displayed in the Unicode translation is language_territory.charset message.

Locale Information Only in Script Logs

The locale information reported only in script execution logs allows you to see the database NLS settings used for script execution. The information is as follows:

- NLS instance settings—This shows the language, territory, and character set specified for your database with NLS_LANGUAGE, NLS_TERRITORY, and NLS_CHARACTERSET database parameters.
- NLS config settings—This shows the language, territory, and character set from QSA's configuration file (QEXECD.CONF). QSA automatically records settings for the NLS_LANGUAGE, NLS_TERRITORY, and NLS_CHARACTERSET database parameters in its configuration file. It then retrieves them from this file during script execution. NLS settings in QEXECD.CONF must not be changed.
- NLS environment settings—This shows the language, territory, and character set that were actually used by QSA to connect to your database during script execution. The values for NLS environment settings should be the same as those for NLS config settings. However, they might not be the same if the operating system experienced problems or if the database ran out of space during script execution.

Use the SQL Editor for a Certain Locale

The SQL Editor supports locale settings as follows:

- Query results—When a query is run from the SQL Editor, query results are displayed in the characters for your client computer operating system character set.
- Oracle names and SQL—When you enter SQL statements in the SQL Editor, you can enter Oracle names in the characters for your client computer character set. However, you must use U.S. English (US7ASCII) as the input language for the command portion of each SQL statement.
- Scripts—When a Space Manager script is viewed in the SQL Editor, script statements are displayed in U.S. English characters (US7ASCII). Within script statements, object names, object owner names, and tablespace names are displayed in the characters for your client computer character set.

As you use the SQL Editor, you can easily switch between input languages using Windows options such as the Language Bar and key sequences. For more information, refer to Windows documentation on input locale.

- i Important** For characters to display correctly in the SQL Editor, the locale for the connecting user must be compatible with the language for the client computer operating system. See [Check the Client Computer Character Sets](#) on page 230 for more information.

Fonts Used in the SQL Editor

The SQL Editor defaults to a certain font based on the language setting for your client computer operating system. The defaults used are as follows—

- Font for Japanese—Microsoft's MS 明朝 font is the default font for Japanese.
- Font for Western languages—Microsoft's Courier New is the default font for Western languages such as French and English.

- Font for other languages—A font chosen by the operating system is the default for non-Japanese and non-Western languages such as Chinese and Korean. The default can vary by language.

Default fonts for Japanese and Western languages are fixed-width (monospace) fonts. Default fonts for other languages can be either a fixed-width font or a variable-width (proportional) font.

If you want to change the font used in the SQL Editor, you can do this by creating a **Font** key for the editor in the Windows Registry. You might want to switch to a fixed-width font if a variable-width font is the default. Fixed-width fonts are sometimes easier to read. Values for the SQL Editor's **Font** key can be changed whenever needed. Different fonts can be specified for different users.

To specify a font for the SQL Editor

1. Open the Windows registry on your Space Manager client computer.
 - i Caution:** Editing your Registry incorrectly can cause serious, system-wide problems. It is advised that you back up your Registry before modifying it.
2. Navigate to the SQL Editor key. The path to this key is:
HKEY_CURRENT_USER > SOFTWARE > QUEST SOFTWARE > Space Manager > SQL EDITOR
3. Right-click the SQL Editor key and select **New | Key**. Enter **Font** as the name of the key.
4. Specify the name of the font to use in the SQL Editor as follows:
 - a. Right-click the Font key and select **New | String Value**. Enter *Name* as the name of the value.
 - b. Double-click the Name value and enter a font name in the **Value** data field. For example, enter *Courier New* to use Microsoft's Courier New font.
5. Specify the size of the font to use in the SQL Editor as follows:
 - a. Right-click the Font key and select **New | DWORD Value**. Enter **Size** as the name of the value.
 - b. Double-click the Size value, select **Decimal**, and enter a font size in the **Value** data field. For example, enter *10* for a 10-point size.
6. Close the Windows registry.

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

Contact Quest

For sales or other inquiries, visit www.quest.com/company/contact-us.aspx or call 1-949-754-8000.

Technical Support Resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at <https://support.quest.com>.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product