## Contents

Preface ........................................................................................................................................................................... xiii
  Audience ....................................................................................................................................................................... xiii
  Documentation Accessibility ........................................................................................................................................ xiv
  Finding Information and Patches on My Oracle Support ............................................................................................ xiv
  Finding Oracle Documentation .................................................................................................................................... xvi
  Related Documents .................................................................................................................................................... xvi
  Conventions .............................................................................................................................................................. xvii

Part I  Configuration Tasks

1  Setting Up User Accounts

  Creating an Administrator User Account .......................................................................................................................... 1-1
  Setting Up Required Accounts and Directories ................................................................................................................ 1-2
    Setting Up a Report Server Log Directory ..................................................................................................................... 1-2
    Creating an Operating System Account ......................................................................................................................... 1-3
    Creating a PSUB Log Directory ...................................................................................................................................... 1-3
    Ensuring PSUB Execution Permission (UNIX Only) ......................................................................................................... 1-4
    Modifying the RXCPRD Account’s Profile ....................................................................................................................... 1-4
  Running the Add User Script ........................................................................................................................................ 1-5
    About the Add User Script ............................................................................................................................................ 1-5
    Running the Add User Script in UNIX ............................................................................................................................ 1-6
    Running the Add User Script in Windows ....................................................................................................................... 1-6
    Required Parameters ....................................................................................................................................................... 1-7
    Optional Parameters ....................................................................................................................................................... 1-8
  Maintaining Oracle User and Group User Accounts .................................................................................................... 1-9
    Adding a User to a Group User Account ......................................................................................................................... 1-11
  Granting Data Access to User and Group User Accounts .............................................................................................. 1-11
    Granting Data Access to Programs and Projects ............................................................................................................ 1-12
    Granting Data Access to a Study ................................................................................................................................... 1-12
    Superuser and Study Access Interaction ....................................................................................................................... 1-13
    Revoking User Access ................................................................................................................................................... 1-13
  Granting Data Access to RDC Users ................................................................................................................................ 1-13
    Granting Automatic Access in RDC to Studies Granted in Oracle Clinical ................................................................ 1-13
    Configuring Study and Site Security Privileges ........................................................................................................... 1-14
    Changing the Default Access to DCIs ............................................................................................................................ 1-14
Granting Additional Database Roles to User Accounts .......................................................... 1-15
  Additional Database Roles for RDC Users ........................................................................... 1-16
  Additional Database Roles for Oracle Clinical Users ............................................................ 1-16
Setting Up Data Extract Users ........................................................................................... 1-16
  Creating an Operating System Account on the SAS Server .............................................. 1-17
  Adding User to the OCLSASCR User Group (UNIX only) ............................................... 1-17
  Creating a Queries Table for a Data Extract User .............................................................. 1-17
Setting Up Power Users ....................................................................................................... 1-17
  UNIX .................................................................................................................................. 1-18
  Windows ............................................................................................................................ 1-18
Setting Up Passwords ........................................................................................................... 1-18
  Changing the Password for a Schema or Role ................................................................. 1-18
  Changing the Password for a User ..................................................................................... 1-19
  Enforcing Password Security .............................................................................................. 1-20
  Auditing Passwords ........................................................................................................... 1-20
  Operating System Passwords ............................................................................................. 1-20

2 Oracle Clinical Menu-Based Security
  Predefined Database Roles ................................................................................................. 2-3
  Creating and Modifying Database Roles ........................................................................... 2-3
    Viewing Menu-Role Associations .................................................................................. 2-3
    Modifying Menu-Role Associations ............................................................................. 2-6
  Creating Custom Database Roles ..................................................................................... 2-6
  Associating Roles with Menus .......................................................................................... 2-7
  Adding a Custom Role to OPA_MENU_ROLES ................................................................... 2-8
  Granting a Custom Role Access to a Custom Module ....................................................... 2-8
  Adding Menu Items to Oracle Clinical ............................................................................ 2-8

3 Configuring Discrepancy Management
  Mapping Database Roles to User Roles ........................................................................... 3-1
  Specifying User Roles for the Oracle Clinical Discrepancy Database ............................... 3-2
    Assigning Function Privileges and Layouts ..................................................................... 3-3
  Customizing Layout Definitions ....................................................................................... 3-4
  Customizing Profiles ........................................................................................................ 3-5
  Filtering Profile Views ...................................................................................................... 3-5
  Customizing Flexfields ..................................................................................................... 3-9
    DISC_FLEX1 and DISC_FLEX2 ...................................................................................... 3-10
    DISC_FLEX1_VALUES and DISC_FLEX2_VALUES ...................................................... 3-10
    Example ......................................................................................................................... 3-11
  Defining Reason Codes for Discrepancies ....................................................................... 3-11
    Reason Codes and Descriptions for Manual Discrepancies ........................................ 3-11
    Reason Codes and Descriptions for Univariate Discrepancies ..................................... 3-12
    Reason Codes and Descriptions for Multivariate Discrepancies ................................... 3-14
  Defining the Possible Review Statuses for Discrepancies ................................................ 3-14
    Default Entries for the DISCREPANCY REV STATUS CODE Codelist ....................... 3-15
    Rules for the DISCREPANCY REV STATUS CODE Codelist ....................................... 3-15
  Defining Resolution Reasons for Discrepancies ............................................................. 3-16
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring Data Entry and User Preferences</td>
<td>5-1</td>
</tr>
<tr>
<td>Define Data Entry Configuration Settings</td>
<td>5-1</td>
</tr>
<tr>
<td>Configuring Additional Data Entry User Preferences</td>
<td>5-5</td>
</tr>
<tr>
<td>Configuring Privileged Update</td>
<td>5-7</td>
</tr>
<tr>
<td>Customizing the Oracle Clinical Log-in Window Layout</td>
<td>5-8</td>
</tr>
<tr>
<td>Using the Log-in Layout Editor</td>
<td>5-8</td>
</tr>
<tr>
<td>Modifying the Received DCI window</td>
<td>5-8</td>
</tr>
<tr>
<td>Modifying the Received DCM Window</td>
<td>5-9</td>
</tr>
<tr>
<td>Modifying the Smart Received DCM Window</td>
<td>5-9</td>
</tr>
<tr>
<td>Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report</td>
<td>5-10</td>
</tr>
<tr>
<td>Setting and Enforcing Values</td>
<td>5-12</td>
</tr>
<tr>
<td>Settings</td>
<td>5-12</td>
</tr>
<tr>
<td>Customizing Flex Fields for DCI Forms</td>
<td>5-20</td>
</tr>
<tr>
<td>How Flex Fields Work</td>
<td>5-20</td>
</tr>
<tr>
<td>Flex Field Components</td>
<td>5-21</td>
</tr>
<tr>
<td>Customizing CRF Column Headers in the RDC Classic Surround</td>
<td>5-22</td>
</tr>
<tr>
<td>Default Behavior</td>
<td>5-23</td>
</tr>
<tr>
<td>Functions Used to Modify CRF Column Headers</td>
<td>5-23</td>
</tr>
<tr>
<td>Functions</td>
<td>5-23</td>
</tr>
<tr>
<td>Customizing Online Help</td>
<td>5-24</td>
</tr>
<tr>
<td>Modifying Calls to Online Help Topic Files</td>
<td>5-25</td>
</tr>
<tr>
<td>Copying Xhelp Topic Files</td>
<td>5-26</td>
</tr>
<tr>
<td>Creating Custom Help Files</td>
<td>5-26</td>
</tr>
</tbody>
</table>
6 Configuring Data Extract

Configuring Default Installation Data Extract Settings .................................................. 6-1
Setting Values in Data Extract-Related Reference Codelists ........................................ 6-5
Creating Tablespaces for Data Extract Tables and Indexes ........................................ 6-5
  Creating Tables ........................................................................................................... 6-5
  Entering Tablespace Names in Reference Codelists .................................................... 6-5
  Creating Data Extract Access Accounts Using Local Tablespaces ................................ 6-6
Customizing Data Extract Views ................................................................................. 6-6
Generating Data Extract Views ..................................................................................... 6-7
  Running gen_views on UNIX Platforms .................................................................... 6-7
  Running gen_views on Windows ............................................................................... 6-8
Enabling the View Builder and Converting Views ....................................................... 6-8
  Converting Views ....................................................................................................... 6-8
  Enabling the View Builder in a Study ....................................................................... 6-8
Controlling Access to Data Extract Views ................................................................. 6-9

7 Reference Codelists

Overview ......................................................................................................................... 7-1
Adding a Value to a Reference Codelist ....................................................................... 7-2
  Modifying a Value in a Reference Codelist ............................................................... 7-3
  Running the Reference Codelists Report ................................................................... 7-3
  Maintain/Query Reference Codelists Windows .......................................................... 7-4
Local Reference Codelists .............................................................................................. 7-7
  BATCH QUEUE NAME .................................................................................................. 7-7
  DB_LINKS ..................................................................................................................... 7-8
  DCF COMMENT TEXT .................................................................................................. 7-8
  DCF DEFAULT FOOTERS ............................................................................................ 7-8
  DCF DEFAULT HEADERS ............................................................................................ 7-8
  DCF REPORT LABELS .................................................................................................. 7-8
  DISC COLS .................................................................................................................... 7-8
  DISC_FLEX1 and DISC_FLEX2 .................................................................................... 7-8
  DISC_FLEX1_VALUES and DISC_FLEX2_VALUES ..................................................... 7-8
  FLD RXCMCMCD RDCI DELETE ................................................................................ 7-8
  FLD RXCMCMCD RDCI KEY ..................................................................................... 7-9
  FLD RXCMCMCD RDCM KEY .................................................................................... 7-9
  FLD RXCMCMCD RESPONSE ................................................................................. 7-9
  IND DISC COLS .......................................................................................................... 7-9
  MAN DISC COLS ......................................................................................................... 7-9
  MANHD DISC COLS ..................................................................................................... 7-9
  MC CDS SORT ORDER .................................................................................................. 7-9
  MC COLUMNS .............................................................................................................. 7-9
  MULTI DISC COLS ...................................................................................................... 7-9
  NLS_CONFIG ............................................................................................................... 7-9
  OCL MC PREFS ............................................................................................................ 7-9
  OCL DE_CONFIG .......................................................................................................... 7-10
  OCL DE_PREFS .......................................................................................................... 7-12
  OCL JOB_PREF ............................................................................................................. 7-13
Installation Reference Codelists

APPLICATION AREA CODE ................................................................. 7-19
APPLICATION SYSTEM NAME ............................................................ 7-19
APPROVE VERIFY RETAIN CODE ....................................................... 7-19
APPROVE VERIFY REVERSE CODE .................................................... 7-19
BLIND TYPE CODE ........................................................................... 7-20
CLINICAL PHASE ............................................................................. 7-20
COUNTRIES ..................................................................................... 7-20
CRF PAGE NUMBERING SCHEME ..................................................... 7-20
CRF PAGE STATUS CODES ............................................................... 7-20
CRF PAGE STATUS QUERY ............................................................... 7-20
DATA CHANGE REASON TYPE CODE .............................................. 7-20
DATA CHANGE REASON2 TYPE CODE ............................................ 7-20
DCF LOCK CONDITIONS ................................................................. 7-21
DCF_OPTIONAL_STATUS_CODES ..................................................... 7-21
DCF_STATUS_CODES ....................................................................... 7-21
DCF_CHECKBOX_SIZE ...................................................................... 7-22
DCF_FONT_TYPESIZE ....................................................................... 7-22
DCF_PAGE_DEFINITION ..................................................................... 7-22
DCM_DCI_QG_TYPE_CODE ............................................................... 7-22
DISCREPANCY ACTIONS ROLE Codelists ......................................... 7-22
DISCREPANCY MESSAGES ............................................................... 7-23
DISCREPANCY RESOLU TYPE CODE .............................................. 7-23
DISCREPANCY REV STATUS CODE .................................................. 7-23
DISCREPANCY STATUS ROLE Codelists ........................................... 7-23
DISCREPANCY NO OTHER UPDATE ................................................ 7-23
DISCRETE_VAL_GRP_TYPE_CODE ................................................... 7-23
DISC_STDST_VALUES .................................................................... 7-23
DOSE FORM TYPE CODE ............................................................... 7-23
DOSE FREQUENCY TYPE CODE ...................................................... 7-23
DX_CONFIG ................................................................................... 7-23
DX_EXTENDED_ATTRIBUTES ......................................................... 7-24
DX_INDEX_TABLESPACE ................................................................. 7-25
DX_KEY_NAME ............................................................................... 7-25
DX_ROLES ..................................................................................... 7-25
DX_VIEW_TABLESPACE ................................................................. 7-26
EXECUTION_SCHED_CODE ............................................................. 7-26
Part II  Oracle Clinical Administration Tasks

8 Managing Batch Jobs

User Account Requirements for Batch Jobs .......................................................... 8-1
How PSUB Handles a Request .................................................................................. 8-2
  Asynchronous PSUB Requests ........................................................................... 8-2
  Blocking and Nonblocking Jobs ....................................................................... 8-2
  Checking a Nonblocking Batch Job ................................................................. 8-3
9 Partitioning and Indexing

9-1 Introduction
9-2 Timing Considerations and Deferral of Index or Partition Upgrade
9-4 About Partitioning

9-5 Planning for Partitioning Upgrade
9-6 Pre-upgrade Steps
9-10 Partitioning the Responses Table
9-10 Step 1. Back Up the Database
9-10 Step 2. Export the Responses Table
9-11 Step 3. Prepare the Database
9-12 Step 4. Import Responses Data
9-12 Step 5. Compute Statistics on the Responses Table
9-12 Step 6. Create Indexes on Partitioned Responses Table
9-13 Step 7. Compile Invalid Objects and, Optionally, Restore Database Trigger
9-13 Step 8. Back Up the Database
9-13 Step 9. Regenerate Procedures and Data Extract Views
9-13 Step 10. Back Up the Database
9-13 Partition Maintenance

9-17 Upgrading Indexes
9-18 Index Upgrade Process
9-18 Resizing the Storage Clause for Indexes
9-18 Response Index Upgrade
9-19 ResponsesT (Test Database) Index Upgrade
9-19 RECEIVED_DCI and RECEIVED_DCM Index Upgrades

9-20 Query Tuning Guidelines

10 Utilities

10-1 Computing the Validation Status of All Responses
10-1 Running cnvstatus on UNIX Platforms
10-2 Running cnvstatus on Windows
10-2 Generating Validation Procedures
10-3 Running gen_procs on UNIX Systems
10-3 Running gen_procs on Windows Systems
Deleting Inactive Procedures .............................................................................................................. 10-4

11  Setting Up File Viewing

File Viewing Setup Overview ............................................................................................................. 11-1
Editing formsweb.cfg to Change File Viewing Protocol .............................................................. 11-2
Setting Directory Mappings .......................................................................................................... 11-2
Setting Up Image Viewing for DCI and DCM Form Layouts HTML Preview Feature .......... 11-3

12  Enabling Image Viewing

Setting Up a Central Image Repository .......................................................................................... 12-1
Setting Up a Network User to Run the Oracle Process Manager Service ................................ 12-2
Setting Up Image Viewing During Data Entry .............................................................................. 12-2
Copying Image Files to All Installations ...................................................................................... 12-3

Part III  Appendixes

A  Environment Variables and Registry Settings

Summary of the Oracle Clinical Setup ........................................................................................... A-1
Editing opa_settings.bat ............................................................................................................... A-2
Setting Up UNIX Environments .................................................................................................. A-3
Changing Configuration Settings on UNIX Database Servers .................................................. A-4
Changing Environment Variables on Windows .......................................................................... A-7
Settings that Affect Back-End Job Execution .......................................................................... A-7

Windows Registry Settings ......................................................................................................... A-10
OPA Front End .............................................................................................................................. A-11
Online Help ................................................................................................................................ A-11
Oracle Clinical Front End .......................................................................................................... A-12
RDC Front End ............................................................................................................................ A-12
OCN Front End ........................................................................................................................... A-13
Report Server .............................................................................................................................. A-14
Registry Keys ............................................................................................................................... A-14

B  SAS_VIEW Directory Tree

C  Troubleshooting

Managing High Sequence Numbers .............................................................................................. C-1
Assessing Sequence Sizes ............................................................................................................. C-1
Reseeding Sequences .................................................................................................................. C-3

Error Messages ............................................................................................................................ C-5
Message: ORA-12223 ................................................................................................................. C-5
Message: ORA-04020 ................................................................................................................. C-5
Message: Unable to Change Mode ............................................................................................. C-6

System Malfunction: GPF Occurs During Data Entry ................................................................. C-6
Situation: $ulimit unlimited........................................................................................................ C-6
PSUB Jobs .................................................................................................................................................. C-7
  Check the Failure Text in the Submitted Batch Jobs Window .............................................................. C-7
  Check the PSUB Log Files .................................................................................................................. C-8
  If Batch Jobs Hang and the Batch Queue Is Full .................................................................................... C-9
  Determining if PSUB is Running for a Database ................................................................................ C-10
  Troubleshooting PSUB Based on the Batch Job’s Execution Status ................................................... C-11
  Handling PSUB Failures that Return ”Fatal two-task communication protocol” Error .................. C-14
  Handling PSUB Failure that Returns ”Illegal use of PSLAUNCH...” Error ........................................ C-14
  Tracking Previous PSUB Process Connections ............................................................................... C-15

Database Trace ........................................................................................................................................ C-15

D Oracle Clinical Tablespaces

E Routine Server Administration
  Recreating Symbolic Links—UNIX Only ............................................................................................... E-1
  Relinking Server Code—UNIX Only .................................................................................................... E-1
  Relocating Oracle Clinical .................................................................................................................. E-2
  Updating Oracle Clinical Seed Data ................................................................................................... E-2
  Collecting Statistics for Optimization ............................................................................................... E-3

Oracle Clinical Administrator Glossary

Index
This Oracle Clinical Administrator’s Guide describes activities that may be required of a system or database administrator to maintain an Oracle Clinical site. For installation and one-time tasks, see the Oracle Clinical Installation Guide. For information on configuring Remote Data Capture Onsite (RDC Onsite), see the Oracle Clinical Remote Data Capture Onsite Administrator’s Guide.

This preface contains the following topics:

- **Audience** on page xiii
- **Documentation Accessibility** on page xiv
- **Finding Information and Patches on My Oracle Support** on page xiv
- **Finding Oracle Documentation** on page xvi
- **Related Documents** on page xvi
- **Conventions** on page xvii

**Audience**

To administer an Oracle Clinical installation you need to be able to carry out the tasks listed below. If you lack the necessary skills, one alternative is to engage Oracle Consulting.

**Oracle Database Administrators**

To perform Oracle Clinical database tasks, you should have a level of knowledge equivalent to what is taught in Oracle’s DBA Architecture and Administration course. You must be able to read, edit, and run SQL scripts and review log files. For ongoing administration, additional DBA training is essential.

**System Administrators**

A general understanding of the operating system and networking is required, including:

- For UNIX:
  - Creating and managing user accounts and groups
  - Installing Oracle RDBMS software and patches
  - Identifying space on a file system for Oracle database tablespaces
  - Setting and using environment variables
For Microsoft Windows:

- Creating and managing user accounts and groups
- Creating and managing services
- Installing Oracle software
- Managing settings through the Control Panel applets
- Adding network printers

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Finding Information and Patches on My Oracle Support

Your source for the latest information about Oracle Clinical is Oracle Support's self-service Web site My Oracle Support (formerly MetaLink).

Before you install and use Oracle Clinical, always visit the My Oracle Support Web site for the latest information, including alerts, White Papers, installation verification (smoke tests), bulletins, and patches.

Creating a My Oracle Support Account

You must register at My Oracle Support to obtain a user name and password account before you can enter the Web site.

To register for My Oracle Support:

1. Open a Web browser to https://support.oracle.com.
2. Click the Register link to create a My Oracle Support account.
3. Follow the instructions on the registration page.

Signing In to My Oracle Support

To sign in to My Oracle Support:

1. Open a Web browser to https://support.oracle.com.
2. Click Sign In.
3. Enter your user name and password.
4. Click Go to open the My Oracle Support home page.

Finding Information on My Oracle Support

There are many ways to find information on My Oracle Support.
Searching by Article ID
The fastest way to search for information, including alerts, White Papers, installation verification (smoke tests), and bulletins is by the article ID number, if you know it.

To search by article ID:
2. Locate the Search box in the upper right corner of the My Oracle Support page.
3. Click the sources icon to the left of the search box, and then select Article ID from the list.
4. Enter the article ID number in the text box.
5. Click the magnifying glass icon to the right of the search box (or press the Enter key) to execute your search.

The Knowledge page displays the results of your search. If the article is found, click the link to view the abstract, text, attachments, and related products.

Searching by Product and Topic
You can use the following My Oracle Support tools to browse and search the knowledge base:

- Product Focus — On the Knowledge page under Select Product, type part of the product name and the system immediately filters the product list by the letters you have typed. (You do not need to type “Oracle.”) Select the product you want from the filtered list and then use other search or browse tools to find the information you need.
- Advanced Search — You can specify one or more search criteria, such as source, exact phrase, and related product, to find information. This option is available from the Advanced link on almost all pages.

Finding Patches on My Oracle Support
Be sure to check My Oracle Support for the latest patches, if any, for your product. You can search for patches by patch ID or number, or by product or family.

To locate and download a patch:
2. Click the Patches & Updates tab. The Patches & Updates page opens and displays the Patch Search region. You have the following options:
   - In the Patch Name or Number field, enter the number of the patch you want. (This number is the same as the primary bug number fixed by the patch.) This option is useful if you already know the patch number.
   - To find a patch by product name, release, and platform, click the Product or Family link to enter one or more search criteria.
3. Click Search to execute your query. The Patch Search Results page opens.
4. Click the patch ID number. The system displays details about the patch. In addition, you can view the Read Me file before downloading the patch.
5. Click Download. Follow the instructions on the screen to download, save, and install the patch files.
Finding Oracle Documentation

The Oracle Web site contains links to all Oracle user and reference documentation. You can view or download a single document or an entire product library.

Finding Oracle Health Sciences Documentation

To get user documentation for Oracle Health Sciences applications, go to the Oracle Health Sciences documentation page at:

http://www.oracle.com/technetwork/documentation/hsgbu-154445.html

Note: Always check the Oracle Health Sciences Documentation page to ensure you have the latest updates to the documentation.

Finding Other Oracle Documentation

To get user documentation for other Oracle products:

1. Go to the following Web page:
   http://www.oracle.com/technology/documentation/index.html
   Alternatively, you can go to http://www.oracle.com, point to the Support tab, and then click Documentation.

2. Scroll to the product you need and click the link.

3. Click the link for the documentation you need.

Related Documents

This section lists the documents in the Oracle Clinical documentation set, followed by their part number. The most recent version of each guide is posted on the Oracle Web site; see “Finding Oracle Health Sciences Documentation” on page xvi.

- Oracle Clinical Installation Guide (Part A83779)
- Oracle Clinical Administrator’s Guide (Part A83791)
- Oracle Clinical Getting Started (Part B12308)
- Oracle Clinical Creating a Study (Part A85200)
- Oracle Clinical Conducting a Study (Part A85201)
- Oracle Clinical Interfacing from Oracle Clinical (Part A83793)
- Oracle Clinical Remote Data Capture Onsite Administrator’s Guide (Part E11064)
- Oracle Clinical Remote Data Capture Onsite User’s Guide (Part B31158)
- Oracle Clinical Remote Data Capture Classic Data Entry User’s Guide (Part B13921)

The release notes and the release content document are also posted in the Oracle Health Sciences documentation library.

In addition, Oracle Clinical customers can request a copy of the Oracle Clinical Stable Interface Technical Reference Manual from Oracle Support.
Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
This section includes tasks that you perform when you install Oracle Clinical or the Remote Data Capture option.

- Chapter 1, "Setting Up User Accounts"
- Chapter 2, "Oracle Clinical Menu-Based Security"
- Chapter 3, "Configuring Discrepancy Management"
- Chapter 4, "Configuring the Mass Changes Utility"
- Chapter 5, "Configuring Data Entry and User Preferences"
- Chapter 6, "Configuring Data Extract"
- Chapter 7, "Reference Codelists"
Creating an Administrator User Account

To create an administrator user account with which you can perform all the functions under the Oracle Clinical Admin menu:

1. Log on to SQL*Plus as SYSTEM and run the Add User script; see "Running the Add User Script" on page 1-5.

2. Add one of the following database roles; see "Granting Additional Database Roles to User Accounts" on page 1-15. By default, these roles have access to menus as follows:
   - RXC_ADMIN: Provides access to all Admin menu items.
   - RXC_SUPER: Provides access to all menu items.
   - RXC_SUPER_NOGL: Provides access to all menu items except the Global Library.
   - RXC_DES: Provides access to the RDC Administration tool as well as Oracle Clinical Design menu items
   - RXC_DMGR: Provides access to the RDC Administration tool as well as Oracle Clinical Definition and Conduct menu items

3. Add privileges, if required; see "Setting Up Power Users" on page 1-17.
Setting Up Required Accounts and Directories

To add a user account to Oracle Clinical, you run the Add User script. In that script you are prompted to supply information about the servers, queues, and directories the user will use. You must either set these up in advance and enter the correct information as you run the script, or enter the information in the script and do the required setup afterward.

- **Set up the Reports Server**, see the Oracle Clinical Installation Guide for instructions.
- **Setting Up a Report Server Log Directory** on page 1-2
- **Determine which database roles** the user needs; see Chapter 2, "Oracle Clinical Menu-Based Security"
- **Parameterized Submission (PSUB) user requirements**. The following are required only for users who need to use the PSUB utility; see "Who Needs PSUB?" on page 1-2:
  - Creating an Operating System Account on page 1-3.
  - Creating a PSUB Log Directory on page 1-3
  - Ensuring PSUB Execution Permission (UNIX Only) on page 1-4
  - Modifying the RXCPRD Account’s Profile on page 1-4

**Who Needs PSUB?** Some of the user setup tasks are required only if the user needs to be able to run Oracle Clinical's Parameterized Submission (PSUB) batch utility. PSUB is required to:

- Run batch data load
- Run batch data delete
- Run batch validation
- Generate a default, character-based DCM layout
- Run the following reports: Randomization Report by Treatment, Patient DCI/DCM Matrix, Generate Study Report, Missing and Overdue DCMs, Investigator Corrs & Missing Pgs, Study/Investigator DCM Summary Matrix, Display Treatment Assignments
- Run Validate Study or Validate Site in RDC—the only RDC users who need PSUB

PSUB is not required to:

- Perform basic data entry in either Oracle Clinical or RDC
- Generate graphic layouts and DCI forms
- Run most reports, including the Patient Data and Audit History reports

PSUB initiates batch jobs on the operating system of the database server through the account of a dedicated user, RXCPRD. This account places the logs and output of the PSUB job into a directory that is accessible to the user who submits the job. Oracle Clinical users must therefore have operating system accounts on the database server.

**Setting Up a Report Server Log Directory**

During Oracle Clinical installation you create a Report Server root directory (see the Oracle Clinical Installation Guide for instructions). You can either use the root directory for all users’ report output, or you can create a separate subdirectory for each user under the root directory. In either case, a user’s access is restricted to the reports.
generated by that user in Oracle Clinical or RDC—regardless of whether or not there is a user-specific subdirectory.

In order for users to view reports through Oracle Clinical and RDC, enter the full path to the root directory or user-specific subdirectory when you execute the Add User script or in the Oracle Accounts window.

---

**Note:** The full path cannot exceed 35 characters.

---

**Creating an Operating System Account**

Users who need to run PSUB (see "Who Needs PSUB?" on page 1-2) need an operating system account on the database server. Users who want to run Data Extract jobs and SAS must also have an account on the server that runs SAS, if it is a different machine from the PSUB server.

For setting the password for a user, see "Changing the Password for a User" on page 1-19.

**Accounts on UNIX Systems**

When you create a user account, ensure that the path to the user's login directory does not contain uppercase characters. PSUB changes all path specifications to lowercase on UNIX platforms.

If you must have uppercase characters in the path, you can provide lowercase and uppercase versions of the paths by using symbolic links, as needed.

For example, if the standard path to user bsmith's account is:

```
/usr1/home/Clinical/bsmith
```

You can create this link:

```
% cd /usr1/home
% ln -s Clinical clinical
```

**Accounts on Windows Systems**

For security reasons, when you set up local accounts for users on the Windows database server, do not use the same password as the user's domain account.

**Creating a PSUB Log Directory**

Create a PSUB root directory on the database server and user-specific log subdirectories for users who need PSUB. When a PSUB batch job runs, the system writes the log and output files associated with the job to the log directory of the user who ran the job.

**UNIX**  On UNIX servers, Oracle suggests you create a directory named `log` as a subdirectory of each user's home directory. For example, for user bsmith

```
/u01/home/bsmith/log
```

**Windows**  On Windows servers, Oracle suggests you create a directory named `oc_users`, and beneath that create a subdirectory for each user, which is named for that user. In each user-specific directory, create a log subdirectory.

You must give each user read/write access to their directory.
For example, for user bsmith:

d:\oc_users\bsmith\log

**Making the PSUB Root Directory Accessible**

Each user has a PSUB log directory under the PSUB root directory. Therefore the root directory must be accessible to all users. In the previous examples, the shared PSUB root directories are "/u01/home" (UNIX) and "d:\users" (Windows). In addition, the root directory must be accessible to the Oracle Clinical middle tier. You do this differently, depending upon whether you choose the UNC or HTTP protocol, as described below.

**FTP Protocol**   If you chose FTP as the file viewing protocol, you can skip this step.

**UNC Protocol**   If you chose UNC as the file viewing protocol, you must create a UNC for the PSUB root directory and make the PSUB root directory readable by the network domain account that is used to start the application server.

**HTTP Protocol**  If you chose HTTP as the file viewing protocol, you must set up the PSUB root directory as a virtual directory on the Web Server. Generally, you would setup that Web Server on the same computer as the PSUB root directory. Oracle Database ships with an Oracle HTTP Server that you can use as a Web Server.

---

**Note:** Refer to "Setting Up File Viewing" on page 11-1 for additional information.

---

**Ensuring PSUB Execution Permission (UNIX Only)**

On UNIX database servers, the PSUB utility works by having one operating system account, RXCPROD, use rsh (remsh on HP-UX) to submit batch jobs on behalf of the actual user's operating system account. To submit these jobs on the user's behalf, RXCPROD must have permission to access the user's account. You can grant this permission either: through an entry in /etc/hosts.equiv, which grants RXCPROD the permission for all users, including new users as they come to be created, or through an entry in each account's .rhosts file.

- **For all users at once:** If the database server has an /etc/hosts.equiv file, add official_host_name rxcprod as a line in the file. This grants RXCPROD the permission for all users, including new user accounts as they are created.
- **For each user individually:** Create a file named rhosts in the user's login directory and include official_host_name rxcprod as a line in the file.

where official_host_name is the official name of the computer on which you are installing Oracle Clinical. You must use the official name — not an alias — for the server. The official name is the first listing after the IP address in the /etc/hosts file.

**Modifying the RXCPROD Account's Profile**

RXCPROD is the dedicated PSUB account. Enable RXCPROD to find the programs that PSUB runs by doing the following:

Open the RXCPROD account's .profile file and edit the PATH command:

PATH=$PATH:opapps_home/bin:oracle_home/bin
where *oracle_home* is the path of the Oracle home directory and *opapps_home* is the path of the Oracle Clinical home directory.

### Running the Add User Script

Create a database account for the user in each database instance to which the user connects by running the script `ocl_add_user.sql` in SQL*Plus.

This section contains the following topics:

- **About the Add User Script**
- **Running the Add User Script in UNIX**
- **Running the Add User Script in Windows**
- **Required Parameters**
- **Optional Parameters**

### About the Add User Script

The `ocl_add_user.sql` script performs the following tasks:

- creates an Oracle database account for the user, with the specified password
- sets the user's temporary tablespace to "temp" and default tablespace to "users"
- grants default Oracle Clinical database roles to the user; you can edit the script to assign additional database roles
- grants the RXC_SUPER role for data access to all studies, if specified
- grants RXC_RDC and RDC_ACCESS if access to RDC is required
- makes the RXCLIN_MOD role a non-default role
- creates a record in the Oracle Clinical table `ORACLE_ACCOUNTS`.

**Note:** If you use SQL*Plus to create a Oracle Clinical database account, do not use the IDENTIFIED EXTERNALLY clause; rather, assign an explicit password.

**Note:** You can copy and customize this script, or create different versions of it for different types of users. For example, you can change temporary and default tablespaces, provide default values for some parameters instead of entering a value for each user at a prompt. You can add a call to `crusrq.sql` to create the USER_QUERIES table (see "Creating a Queries Table for a Data Extract User" on page 1-17).

However, if you decide to customize the script, it is important not to modify the line: `alter user &&ops_id default role all except rxclin_mod;`

**Note:** You can give users additional database roles directly in SQL*Plus; see "Granting Additional Database Roles to User Accounts" on page 1-15.
Running the Add User Script in UNIX

1. Log on with your UNIX account.
2. Set the environment.
   For UNIX servers, C shell, enter the following:
   ```
   opa_setup database_instance_name code_environment_designation
   ```
   For example: `%opa_setup burlma9 22`
   For UNIX servers, Bourne shell, enter the following:
   ```
   p1 = database_instance_name
   p2 = code_environment_designation
   opa_setup
   ```
3. Change to the RXC_TOOLS directory:
   ```
   cd $RXC_TOOLS
   ```
4. Call SQL*Plus and log in as SYSTEM:
   ```
   sqlplus system
   ```
5. Run the Add User script:
   ```
   start ocl_add_user.sql
   ```
   See Table 1–1, "Add User Script Required Parameters" and Table 1–2, "Add User Script Optional Parameters" for a description of the prompts output by the Add User script, valid entries, and examples.

Running the Add User Script in Windows

1. Log on with your local account.
2. Open an MS-DOS window.
3. Set the server environment:
   ```
   set p1=db_name
   set p2=code_env
   opa_setup
   ```
   where `db_name` is a database instance name and `code_env` is a code environment designation.
4. Change to the RXC_TOOLS directory:
   ```
   cd %RXC_TOOLS%
   ```
5. Call SQL*Plus and log in as SYSTEM.
   ```
   sqlplus system
   ```
6. Run the Add User script:
   ```
   start ocl_add_user.sql
   ```
## Required Parameters

The following table describes the required Add User script parameters:

<table>
<thead>
<tr>
<th>Prompt/Equivalent in Oracle Accounts Window</th>
<th>Description</th>
<th>Rules and Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID, starting with OPS$ /Account Name</td>
<td>Defines the login name for the user. Names are not case sensitive. OPS$ is required at the beginning of the account name if the user needs to run PSUB processes in Oracle Clinical; see &quot;Who Needs PSUB?&quot; on page 1-2. The letters after OPS$ (if OPS$ is included) must be identical to the user’s operating system account name on the database server. If OPS$ is not included, the user ID you enter here must be identical to the user’s operating system account name on the database server.</td>
<td>OPS$bsmith bsmith</td>
</tr>
<tr>
<td>Password Not in the Oracle Accounts window</td>
<td>Defines the login password for the user. Passwords are not case sensitive. See &quot;Changing the Password for a User&quot; on page 1-19.</td>
<td>farrier</td>
</tr>
<tr>
<td>Last Name /Last Name</td>
<td>Specifies the user’s surname (family name). Names are not case sensitive; names are automatically capitalized.</td>
<td>Smith</td>
</tr>
<tr>
<td>First Name First Name</td>
<td>Specifies the user’s given name. Names are not case sensitive; names are automatically capitalized.</td>
<td>William</td>
</tr>
<tr>
<td>PSUB Log Directory /PSUB Directory</td>
<td>Specifies the directory where PSUB places output and log files from PSUB jobs. See &quot;Who Needs PSUB?&quot; on page 1-2. RDC users need a PSUB log directory only if they run Validate Study or Validate Site. You create the root directory during installation.</td>
<td>UNIX: /users/bsmith/log Windows (use UNC): \users\bsmith\log</td>
</tr>
<tr>
<td>Report Server Log Directory /Report Server Directory</td>
<td>Specifies the designated location for saving output from the Report Server. Required for generating and viewing Patient Data Reports and most Oracle Clinical reports. You create the directory as part of setting up the Report Server during Oracle Clinical installation.</td>
<td>Must be specified in UNC format. String length cannot exceed 35 characters. \oc_srvr\users\bsmith</td>
</tr>
</tbody>
</table>
Running the Add User Script

Optional Parameters

The remaining parameters are optional. In addition, you can use the Maintain Oracle Accounts form to manage these optional parameters; see “Maintaining Oracle User and Group User Accounts” on page 1-9.

Table 1–2  Add User Script Optional Parameters

<table>
<thead>
<tr>
<th>Prompt/Equivalent in Oracle Accounts Window</th>
<th>Description</th>
<th>Rules and Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Doc Dir /Custom Help Directory</td>
<td>Specifies the location of your site-specific context-sensitive HTML help files.</td>
<td>RXC_PRINTER (This is the default value for the database shipped in the OCL_JOB_PREF local reference codelist.)</td>
</tr>
<tr>
<td>Printer for PSUB /Default PSUB Printer</td>
<td>The printer to which PSUB print jobs are routed, in upper case. This response must match the Short Value of an active entry in the PRINT QUEUE NAME local reference codelist. Required for RDC users only if they run Validate Study or Validate Site.</td>
<td>RXC_BATCH_QUEUE (This is the default value for the database shipped in the OCL_JOB_PREF local reference codelist.)</td>
</tr>
<tr>
<td>Queue /Default PSUB Queue</td>
<td>The designation for the queue in which PSUB batch jobs are executed. This response must match the Short Value of an active entry in the BATCH QUEUE NAME local reference codelist. Must be entered in upper case. Required for RDC users only if they run Validate Study or Validate Site.</td>
<td>RXC_BATCH_QUEUE (This is the default value for the database shipped in the OCL_JOB_PREF local reference codelist.)</td>
</tr>
</tbody>
</table>
Setting Up User Accounts

Maintaining Oracle User and Group User Accounts

After you create a user with the Add User script, the user account appears in the Oracle Accounts window. Use this window to:

- Create group user accounts
- View and modify settings for individual user accounts

Do not use this window to create new individual user accounts because you cannot assign database roles to users here. Use the Add User script.

You can access other windows from Oracle Accounts to do the following tasks:

- **Adding a User to a Group User Account** on page 1-11; use this window to assign a user to a group user account. You can then grant data access to all the users assigned to the group user account at the same time.
- **Granting Data Access to a Study**; use this window to grant access to one study at a time to a single user or group user account
- **Granting Data Access to Programs and Projects** on page 1-12; use this window to grant access to all the studies in a program or project at the same time to a single user or group user account

To create a group user account:

1. From the **Data** menu, select **Insert Record**.
2. Enter values in the fields.

---

### Table 1–2 (Cont.) Add User Script Optional Parameters

<table>
<thead>
<tr>
<th>Prompt/Equivalent in Oracle Accounts Window</th>
<th>Description</th>
<th>Rules and Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printer for the Report Server /Default RS Printer</td>
<td>The printer to which print output from a report server job is routed. This response must match the Short Value of an active entry in the PRINT QUEUE NAME local reference codelist. Must be entered in upper case.</td>
<td>RXC_PRINTER (This is the default value for the database shipped in the OCL_JOB_PREF local reference codelist.)</td>
</tr>
<tr>
<td>Report Server /Default Report Server</td>
<td>Specify a value if you want to override the system default for a specific user. When the user prepares to submit a report request, this response specifies which report server is offered as the default. This response must match the Short Value of an active entry in the REPORT_SERVER local reference codelist.</td>
<td>REPORT_SERVER (This is the default value for the database shipped in the OCL_JOB_PREF local reference codelist.)</td>
</tr>
<tr>
<td>Job Set Report Server /Default Job Set Server</td>
<td>When the user prepares to submit a job set request, this response specifies which report server is offered as the default. This response must match the Short Value of an active entry in the REPORT_SERVER local reference codelist. Required for RDC users only if they run Validate Study or Validate Site.</td>
<td>REPORT_SERVER (This is the default value for the database shipped in the OCL_JOB_PREF local reference codelist.)</td>
</tr>
<tr>
<td>PSUB Scheduler Report Server /Default PSUB Scheduler Report Server</td>
<td>When the user prepares to submit a scheduled PSUB job request, this response specifies which report server is offered as the default. This response must match the Short Value of an active entry in the REPORT_SERVER local reference codelist. Required for RDC users only if they run Validate Study or Validate Site.</td>
<td>REPORT_SERVER (This is the default value for the database shipped in the OCL_JOB_PREF local reference codelist.)</td>
</tr>
</tbody>
</table>
The window contains the following fields:

**Account Type**  If set to Oracle, the record is an individual user account. To create a group user account, set to Group.

**Account Name**  For an individual user account, the user ID; for a group user account, its name.

**Last Name**  Family name or surname of the user with the individual user account; not required for group user accounts.

**First Name**  Given name of the user with the individual user account; not required for group accounts.

**Super User?**  If checked, the user or group user account has superuser status and can see data in all studies on the database. If not checked, the user or group user account can see data only for studies to which they are explicitly allowed access; see "Granting Data Access to User and Group User Accounts" on page 1-11.

**PSUB Directory**  Location for log and output files from Parameterized Submission jobs; see "Creating a PSUB Log Directory" on page 1-3.


---

**Note:** The location designation for any report server log directory, whether directory path or UNC, cannot exceed 35 characters

---

**Custom Help Directory**  Location of company-specific context-sensitive html files, if any.

**Default PSUB Printer**  The printer to which PSUB print jobs are routed, in upper case. This response must match the Short Value of an active entry in the PRINT QUEUE NAME local reference codelist.

**Default RS Printer**  The printer to which print output from a report server job is routed. This response must match the Short Value of an active entry in the PRINT QUEUE NAME local reference codelist. Must be entered in upper case.

**Default PSUB Queue**  The designation for the queue in which PSUB batch jobs are executed. This response must match the Short Value of an active entry in the BATCH_ QUEUE_NAME local reference codelist. Must be entered in upper case.

**Default Report Server**  When the user prepares to submit a report request, this response specifies which report server is offered as the default. This response must match the Short Value of an active entry in the REPORT_SERVER local reference codelist.

**Default Job Set Report Server**  When the user prepares to submit a job set request, this response specifies which report server is offered as the default. This response must match the Short Value of an active entry in the REPORT_SERVER local reference codelist. See "Using Job Sets to Control Execution Order" on page 8-11 for information on job sets.
Default PSUB Scheduler Report Server  When the user prepares to submit a scheduled PSUB job request, this response specifies which report server is offered as the default. This response must match the Short Value of an active entry in the REPORT_SERVER local reference codelist.

**Note:** The location designation for any report server log directory, whether directory path or UNC, cannot exceed 35 characters.

### Adding a User to a Group User Account

You can add all the users who should have access to the same set of studies, programs, or projects to the same group user account and assign data access to all the users at the same time through the group user account. An individual user can belong to multiple group user accounts.

To add a user to a user group:

1. Navigate to Admin, Users, and then Oracle Accounts. The system opens the Oracle Accounts window.
2. Query for the individual user you want to add to a user group.
3. Click Group Membership. The system displays the Group Membership window.
4. From the Group Membership field’s list of values, select the group to which you want to assign the user. If you want to assign the user to multiple user groups, use multiple rows.
5. Save your work.

### Granting Data Access to User and Group User Accounts

You must explicitly give users access to study data, either by granting them Superuser status, which allows access to all studies, or explicitly granting access to specific studies or groups of studies (programs and projects). For RDC users, you can use either the Oracle Accounts window or the Study and Site Security windows; see "Granting Data Access to RDC Users" on page 1-13.

To allow an Oracle Clinical or RDC user—even a user who will work exclusively in RDC—to view study data, you must grant study or superuser access in Oracle Clinical. You can grant data access to a user or group user account in several ways:

- Grant access to data in all studies by granting the user or group user account superuser status, either in the Oracle Accounts window or, for individual users, when you run the Add User script.
- If the user or group user accounts does not have superuser status, you can grant access to data in one or more studies, either to one study at a time or to all studies in a program or project.

You can grant data access at several levels:

- Granting Data Access to Programs and Projects on page 1-12
- Granting Data Access to a Study on page 1-12

See also "Superuser and Study Access Interaction" on page 1-13 and "Revoking User Access" on page 1-13.
Granting Data Access to Programs and Projects

Assigning program or project access to a user or group user account enables either the individual user or the set of users in the group to have access to all studies associated with the program or project.

1. Navigate to **Admin, Users**, and then **Oracle Accounts**. The system opens the Oracle Accounts window.

2. Query for the account with which you want to work.

   **Note:** You cannot assign project or program access to a user or user group whose **Super User?** flag is checked. That account already has access to all programs and projects.

3. Click **Programs/Projects**. The Programs window displays.

4. In the **Program** field, from the list of values select the name of the program to which you want the user or user group to have access.

5. In the **Project** field, from the list of values select the program to which the user will have access. If the user should have access to all projects within a program, enter a percent sign (%) in the project field.

   If you want to assign multiple programs to the account, or multiple (but not all) projects within a program, use multiple rows. There is no limit to the number of programs to which the user can have access.

6. Save your work.

Granting Data Access to a Study

To assign study data access to a user or user group:

1. Navigate to **Admin, Users**, and then **Oracle Accounts**. The system opens the Maintain Oracle Accounts window.

2. Query for the account with which you want to work.

   **Note:** You cannot assign study access to a user or user group whose **Super User?** flag is checked. That account already has access to all studies.

3. Click **Studies**. The Studies window displays.

4. In the **Study** field, from the list of values select a study to which you want the user or user group to have access. If you want to assign multiple studies to the account, use multiple rows. There is no limit to the number of studies you can specify.

5. Save your work.

   **Note:** If a user creates a new study and does not have access to it through the project and/or program it belongs to, the system automatically gives the user access to the study he or she has just created.
Superuser and Study Access Interaction

You cannot assign project or program access to a user or user group whose Super User? flag is checked. That account already has access to all programs and projects.

However, if you assign access to programs, projects, or studies to a user whose Super User? flag is not checked, and then check that user’s Super User? flag, the superuser status overrides the existing specific privileges, but the existing privileges are still displayed in the Projects, Programs, and Studies windows.

If an RDC user has superuser status in the Oracle Accounts window and has access to only a subset of RDC studies in the Study Security window, the superuser status overrides the study-specific privileges and the user has access to all studies. However, you can use the Study Security window to limit the type of access to a particular study. See the Oracle Clinical Remote Data Capture Onsite Administrator’s Guide for information.

Revoking User Access

---

**Note:** If a user has both superuser status and access to specific studies defined, you must revoke the superuser status before you can revoke his/her access to a specific study.

When you revoke access to a program, you revoke access to all projects within that program.

---

To revoke a user’s study, user group, project, or program access:

1. Navigate to Admin, Users, and then Oracle Accounts. The system displays the Oracle Accounts window.
2. Query for the account with which you want to work.
3. Click either Studies, Programs/Projects, or Group Membership.
4. Select the study, program/project combination, or user group from which you want to remove the user.
5. From the Data menu, select Delete Record.
6. Save your work.

Granting Data Access to RDC Users

This section contains the following topics:

- Granting Automatic Access in RDC to Studies Granted in Oracle Clinical on page 1-13
- Configuring Study and Site Security Privileges on page 1-14
- Changing the Default Access to DCIs on page 1-14

Granting Automatic Access in RDC to Studies Granted in Oracle Clinical

You can use a reference codelist setting to determine whether users who have access to data for a particular study in Oracle Clinical automatically have access to the study in RDC.

In the OCL_STATE local reference codelist, set the DMGR RDC ACCESS short value as follows:
When set to **YES**, a user with no study privileges defined for RDC but with study access defined in Oracle Clinical is automatically given RDC Onsite access to the study as well, in both Test and Production modes. The user has all RDC privileges except APPROVE and VERIFY. UPD_LOCK_OC, an Oracle Clinical-specific privilege, is also excluded. You can restrict such a user's access to RDC Onsite by limiting privileges at the study or site level; see the *Oracle Clinical Remote Data Capture Onsite Administrator's Guide* for further information.

When set to **NO**, a user granted access to a study in Oracle Clinical does not automatically have access to that study in RDC Onsite. You can use the Study Security form to assign specific privileges to the user; see the *Oracle Clinical Remote Data Capture Onsite Administrator's Guide* for further information.

Users with the **Super User?** flag selected in the Oracle Accounts form in Oracle Clinical have access to all studies in both Oracle Clinical and RDC, and in both Test and Production modes.

### Configuring Study and Site Security Privileges

You can give RDC users specific privileges for particular studies and sites in the Study and Site Security windows, which are included in both Oracle Clinical and in the RDC Onsite Administrator's Tool; see "Configuring Study and Site Security for Discrepancy Management" on page 3-26.

### Changing the Default Access to DCIs

RDC Onsite includes a predefined set of user roles. By default, these roles have UNRESTRICTED access to all DCIs. You can change the default access for any role to RESTRICTED.

Users assigned to a role with UNRESTRICTED access to DCIs can access any DCI in RDC Onsite unless access has been denied to a particular DCI in a particular study through the DCI Access window in the Oracle Clinical Definition menu; see *Oracle Clinical Creating a Study* for study-level information.

Users assigned to a role with RESTRICTED access to DCIs cannot access any DCIs in RDC Onsite unless access has been granted to a particular DCI in a particular study.

You may want to create custom database roles specifically for the purpose of restricting access to certain DCIs to smaller groups of people (see "Creating Custom Database Roles" on page 2-6 for more information and an example). If you do, you must add the new role in the Maintain DCI Access by Role window and specify a default access of RESTRICTED or UNRESTRICTED for it.

---

**Caution:** If you create a new user role but do not specify a default value for DCI access, users assigned to that role cannot log in to RDC Onsite. You must define the default access to DCIs for every user role you plan to assign.

If you assign a user to more than one role, and those roles have conflicting DCI access, the user cannot log it to RDC Onsite.

---

Before you can change the default DCI access for a user, the user role must exist (must be valid). You cannot change the default DCI access if the user role does not exist.
To define the DCI access for a user role:

1. Open Oracle Clinical.
2. Navigate to Admin and then select Users and Roles.

   Alternatively, you can select one of the following menu options depending upon your administrator privileges and current task:
   - Select Test Default DCI Access if you want to try out DCI access before implementing the feature in a live study.
   - Select Query Default DCI Access by Role if you only want to view the current settings but make no changes.
4. Enter a valid user role in the User Role field. You can:
   - Type the name of a valid user role into the field.
   - Click the List of Values button, and then select a user role from the list. The list includes all the user roles currently defined in the USER GROUP ROLES installation reference codelist.
5. Enter the default DCI access for the selected user role. Valid entries are:
   - UNRESTRICTED — Allows study/site access to all DCIs unless otherwise restricted in the DCI Access form for the study.
   - RESTRICTED — Does not allow access to any DCIs unless you specify exceptions in the DCI Access form for the study.

   You can type a valid entry directly into the field. Alternatively, you can click the List of Values button, and then select from the list.
6. Continue to enter each user role and the type of DCI access allowed.
7. Save your changes.

For each record in the Maintain Default DCI Access by Role form, Oracle Clinical creates and maintain an audit trail.

Upon initial entry to the form, Oracle Clinical populates the form with all the user roles defined in the USER GROUP ROLES reference codelist. For each user role, the Default DCI Access field is set to UNRESTRICTED. You must add any new user roles that you create.

---

**Granting Additional Database Roles to User Accounts**

This section contains the following topics:

- Additional Database Roles for RDC Users on page 1-16
- Additional Database Roles for RDC Users on page 1-16

The Add User script grants the minimum database roles required for a user to access Oracle Clinical and, if you specify that RDC access is required, RDC.

- Oracle Clinical default database roles: CONNECT, RESOURCE, RXCLIN_READ, RXC_ANY, AND OCL_ACCESS
- RDC default database roles: RXC_RDC and RDC_ACCESS

Users require additional database roles to do meaningful work in either Oracle Clinical or RDC.
To grant one or more database roles to a user:

1. Log in to SQL*Plus as SYSTEM.
2. Grant a role to a user:

   grant database_role to user_name

   For example: grant rxc_site to BSMITH

### Additional Database Roles for RDC Users

You must explicitly grant every RDC Onsite user at least one database role. You can use the predefined database roles listed in Table 1–3, selecting the role that matches the user's job function, or define additional database roles if you need to further fine-tune security privileges; see "Creating and Modifying Database Roles" on page 2-3.

These database roles are mapped to user roles in the installation reference codelist USER_GROUP_ROLES. Those user roles are used to define security privileges and to customize various aspects of the user interface. See Chapter 3, “Configuring Discrepancy Management” for further information.

<table>
<thead>
<tr>
<th>Database Role</th>
<th>Typical User Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXC_DMGR</td>
<td>Data manager</td>
</tr>
<tr>
<td>RXC_SUPER</td>
<td>Data manager</td>
</tr>
<tr>
<td>RXC_CRA</td>
<td>Clinical Research Associate (CRA)</td>
</tr>
<tr>
<td>RXC_SITE</td>
<td>Site user, study coordinator, or other person at the remote site responsible for entering patient data</td>
</tr>
<tr>
<td>RXC_INV</td>
<td>Investigator at the remote site who can approve CRFs</td>
</tr>
</tbody>
</table>

### Additional Database Roles for Oracle Clinical Users

The database roles assigned to a user determine which tasks a user can perform by controlling which menu paths the user can see in the user interface—for example, Study Design, Data Entry, or Administration—and within those areas, finer distinctions such as whether the user can make changes or only view existing data. Oracle Clinical includes many predefined database roles for this purpose. You can also define your own database roles; see Chapter 2, "Oracle Clinical Menu-Based Security" for more information.

The roles in Table 1–3 apply to the Oracle Clinical discrepancy management system as well as to RDC. Give one of the above roles to each user who has one of the corresponding job functions and who will be working with discrepancies in Oracle Clinical’s Maintain Discrepancy Database window.

### Setting Up Data Extract Users

Oracle Clinical users who need to generate Data Extract views and write macros require additional setup:

- Creating an Operating System Account on the SAS Server on page 1-17
- Adding User to the OCLSASCR User Group (UNIX only) on page 1-17
- Creating a Queries Table for a Data Extract User on page 1-17
Creating an Operating System Account on the SAS Server

User who need to generate SAS Data Extract view need an operating system account on the SAS server. If they do not already have one, create one; see instructions in "Creating an Operating System Account" on page 1-3.

Adding User to the OCLSASCR User Group (UNIX only)

Add the user to the OCLSASCR user group to give the user access to the RXC_USER directories that hold the SAS Data Extract Views. The OCLSASCR user group is created as part of the Oracle Clinical installation and has all the privileges required to use SAS. Refer to the Oracle Clinical Installation Guide for instructions on creating the OCLSASCR group.

You can add the user to the OCLSASCR user group by:

- using the `usermod` command, or
- editing the `/etc/group` and `/etc/logingroup` files, if these files are not linked; if these files are linked, it is only necessary to modify the `/etc/group` file.

Creating a Queries Table for a Data Extract User

Oracle provides a script, `crusrq.sql`, that creates a table called USER_QUERIES in an individual user's schema. The table provides a location to save the SQL code the user creates using data extract functions. Run the script for each user who will write data extract functions.

To create this table:

1. Change directories to the RXC_INSTALL directory.
2. Run this command:
   ```
   sqlplus @crusrq
   ```

   Alternatively, you can modify the Add User script to create the USER_QUERIES table automatically by adding these lines:

   ```
   connect &&ops_id/&&pwd
   @rxc_install:crusrq.sql
   ```

Setting Up Power Users

Most end users do not require additions to a login script for Oracle Clinical purposes. Only power users who want to run opa_setup from the command line (in order to condition their environment to point to a particular database, or a code environment, or both), must add commands to a login script on that machine, in order to configure their environments.

A power user might use opa_setup to set environment variables for such tasks as:

- running client applications other than Oracle Clinical (such as SQL*Plus, SAS, or a reporting program) against an Oracle Clinical database
- installing patches to Oracle Clinical server code or databases
- running Oracle Clinical administrative SQL scripts

For users who will perform these tasks, commands must be added to the login scripts that:

- define RXC_LOG
Setting Up Passwords

- add SAS
- add source (on C shell).

UNIX

**Bourne/Korn shell**  Edit the user’s .profile file. In the following example, SAS_home is the directory where the SAS executable is located.

```bash
# Include OPA directories
PATH=$PATH:/opa_home/bin:/sas_home; export PATH
# Define RXC_LOG for command line utilities
RXC_LOG=$HOME/log; export RXC_LOG
```

**C shell**  Edit the user’s .cshrc file. For example:

```bash
# Include OPA directories
set path=($path /opa_home/bin /sas_home )
# Define RXC_LOG for command line utilities
setenv RXC_LOG $HOME/log
# Create alias for opa_setup
source /opa_home/bin/copa_setup_alias
```

Windows

No changes to a login script are required. In order to run opa_setup, ensure that opapps_home/bin is in the PATH environment variable.

Setting Up Passwords

Changing passwords depends on whether they are for a schema, a role, or for a user.

The section is comprised of the following topics:

- Changing the Password for a Schema or Role on page 1-18
- Changing the Password for a User on page 1-19
- Enforcing Password Security on page 1-20
- Auditing Passwords on page 1-20
- Operating System Passwords on page 1-20

---

**Note:** Passwords cannot contain these characters: { | } @ ;

Changing the Password for a Schema or Role

Certain Oracle Clinical database objects are protected by password. These objects are listed and described in Table 1–4, "Password-protected database objects".

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXC_MAA</td>
<td>Schema</td>
<td>Required for Data Extract.</td>
</tr>
<tr>
<td>RXC_PD</td>
<td>Schema</td>
<td>Required for Procedure generation.</td>
</tr>
<tr>
<td>RXC_REP</td>
<td>Schema</td>
<td>Required for replication.</td>
</tr>
<tr>
<td>RXC_DISC_REP</td>
<td>Schema</td>
<td>Required for disconnected replication.</td>
</tr>
</tbody>
</table>
Setting Up Passwords

1-19

Encrypted versions of the passwords associated with these objects are stored in two locations:

1. an Oracle dictionary table, and
2. an Oracle Clinical passwords table.

If you want to change any of these passwords, you must change the encrypted value in both locations. You can change the passwords in both places at once with the `SET_PWD` utility, following these steps:

1. Log on to the database server using an account that is set up to run Oracle Clinical back end jobs.
2. Set environment variables for the database and code environment. For example, for database `test` and code environment `46`:
   a. UNIX: `opa_setup test 46`
   b. Windows:
      
      ```
      set p1=test
      set p2=46
      opa_setup
      ```
3. Enter a command in the following format:

   ```
   set_pwd rxc/rxc_password db_object_name db_object_password
   ```

   where `rxc_password` is the password for the rxc account, `db_object_name` is the name of the database object from Table 1–4, "Password-protected database objects", and `db_object_password` is the new password for that object. For example:

   ```
   set_pwd rxc/notrxc rxclin_mod r2d2c3po
   ```

Changing the Password for a User

Users can change their own database passwords either in SQL*Plus or with the Oracle Clinical menu by choosing Admin, then Users, and then Database Password.

In the Oracle Database Password window:

1. Type your password in Enter Password text box.
2. Type it again in the Confirm Password text box.
3. Click OK.

Administrators can change the password for an Oracle Clinical user in SQL*Plus as usual:

1. In SQL*Plus, connect to the database as the SYSTEM user.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXCLIN_MOD</td>
<td>Role</td>
<td>Enables Oracle Clinical users to write to the database. This role is set for users when they log in through the application.</td>
</tr>
<tr>
<td>RXC_SERVLETSP</td>
<td>Schema</td>
<td>Used for accessing study data from Data Entry window in Production mode.</td>
</tr>
<tr>
<td>RXC_SERVLETST</td>
<td>Schema</td>
<td>Used for accessing study data from Data Entry window in Test mode.</td>
</tr>
</tbody>
</table>

Table 1–4 (Cont.) Password-protected database objects
2. Reset the password for the user:

   \texttt{alter user user identified by user\_password;}

**Enforcing Password Security**

Oracle enables a database administrator to enforce various rules about passwords at the database level, including setting a password lifetime, after which users must set a new password; disallowing reuse of previous passwords; locking an account after a user attempts to logon a specified number of times; and creating complexity rules for passwords through a PL/SQL function.

All rules that you set at the database level apply when a user connects through Oracle Clinical. If the password has expired, a dialog box that prompts the user for a new password automatically appears. For further details, refer to your Oracle Administrator’s Guide.

**Auditing Passwords**

Oracle recommends that you turn on the auditing system for roles. For information on the AUDIT command, refer to the Oracle SQL Reference manual.

To track failed attempts to create, alter, drop, or set a role, issue the following statement:

   \texttt{audit role by access whenever not successful}

**Operating System Passwords**

A system-level account must exist on the database server computer in order for Oracle Clinical users to run PSUB jobs. On UNIX, you can choose whether to allow users to store their UNIX account passwords in the Oracle Clinical database in an encrypted format. On Windows, users must store their passwords to be able to run any PSUB job.

You use the Operating System Password window to change whether your system password is stored in Oracle Clinical. This window is accessible either:

- by selecting Admin, then Users, and OS Password, or
- by selecting the Server OS Password button in the Submission Details window.

The Operating System Password window consists of an Enter password text box and a check box entitled, \texttt{Save encrypted password in the database}?

The behavior of the check box depends on your operating system and is described in the following subsections:

- "UNIX Passwords" on page 1-21
- "Windows Passwords" on page 1-21

---

**Note:** If the PSUB server belongs to a Windows domain, users’ local accounts on that server must match the passwords of their domain accounts. In this way, the correct authentication is passed to the server and access is allowed to the domain resources. Otherwise, users’ PSUB jobs do not print.
UNIX Passwords

On UNIX servers, you control the Save encrypted password in the database? check box with the Short Value of the USR_SAVE_OSPASS entry in the OCL_STATE local reference codelist. The default value of USR_SAVE_OS_PASS is "N" (No). In this case, the check box the system displays in the Operating System Password window is clear and not updateable. If you change the value of USR_SAVE_OS_PASS to "Y" (Yes), the check box is still clear, but it is updateable, which allows the user to select it.

Windows Passwords

On Windows servers, a stored password is required for any PSUB server-side job. When end users see the Server OS Password dialog box, the Save encrypted password in the database? check box is selected and cannot be updated. The first time a user runs a PSUB job, the system prompts for a password if it has not already been stored in the database.
Oracle Clinical Menu-Based Security

This section includes the following topics:

- Creating and Modifying Database Roles on page 2-3
- Adding Menu Items to Oracle Clinical on page 2-8

You control which users have access to which menu items in the Oracle Clinical Navigator (see Figure 2–1) by assigning database roles to users.

Oracle Clinical includes a set of predefined database roles that allow access to a predefined set of Oracle Clinical menu items, including second- and third-level menu items (see Figure 2–2). You can enforce security by assigning users only the database roles they need to do their work, preventing them from seeing other parts of the system and taking actions they are not authorized to take.

If necessary, you can modify the menu items associated with the predefined database roles or create entirely new database roles.

Figure 2–1 Oracle Clinical Navigator with Top-Level Menu Items Displayed

Although each company distributes Oracle Clinical tasks differently among its personnel, following is a guideline for which users need which menu items:

- Administrators need some or all of the Admin menu.
- Study designers need some of the Admin menu and the Plan, Design, and Definition menus. They may also need Glib (Global Library), or you may have people who use only the Global Library.
- Data Managers need the Conduct, Data Entry, and Labs menus. They may also need some or all of the Definition menu.
- Data entry operators need the Data Entry menu.
- Programmers who write validation and derivation Procedures and data extract macros need parts of the Definition menu.

For information about the tasks in each menu, see the following Oracle Clinical user documentation:

- The *Oracle Clinical Administrator’s Guide* has information on the Admin menu.
- *Oracle Clinical Creating a Study* has information on the Plan, Design, and Definition menus.
- *Oracle Clinical Conducting a Study* has information on the Conduct, Data Entry, and Labs menus.

In many cases, there are two menu items for the same form associated with different database roles. In this way you control user’s privileges through menu access. Menu items based on the same form may differ as follows:

- **Query and Read-Write Versions**: One menu item allows read-only privileges and the other allows write privileges as well. If a user has access only to the Query version of the form, he or she cannot view the data there but cannot make any changes.
- **Provisional and Active Definitions**: Definitional objects can have a status of Provisional or Active. Some menu items allow the user read and write access only to Provisional objects, while a different version of the same form allows access to Active objects, which may be currently in production use, as well.
- **Test and Production Patient Data**: One menu items allows read and write access to test data and the other allows read and write access to production data.

*Figure 2–2  Oracle Clinical Navigator with the Definition DCIs Menu Displayed*
Predefined Database Roles

To see a complete list of the predefined database roles and the menu items to which they allow access, run the Menu Roles report in the Developer's Toolkit.

To run the Menu Roles report, navigate to DTK, then Menu Roles. To see the Developer's Toolkit (DTK menu item) you must have the DTK_ADMIN database role. See "Granting Additional Database Roles to User Accounts" on page 1-15.

If you create custom roles for your Oracle Clinical database and set up menu security for these roles, you can run the Menu Roles report to confirm that you have set up these roles correctly. The Menu Roles report describes, for both default and custom roles, the menu items to which each role gives a user access. This report applies to the current database only.

Note: If the Menu Roles report does not show a custom role you have defined, you may not have defined a record for that role in the OPA_MENU_ROLES codelist. See "Adding a Custom Role to OPA_MENU_ROLES" on page 2-8.

Creating and Modifying Database Roles

To modify menu security, you must access the Developer's Toolkit (DTK) menu in the Oracle Pharmaceutical Applications Navigator window. Entries on the DTK menu are accessible only to those database accounts granted the DTK_ADMIN role. The DBA should grant this role to those accounts with the responsibility for maintaining Oracle Clinical roles. This section assumes that your account has the DTK_ADMIN role.

This section is comprised of the following topics:

- Viewing Menu-Role Associations on page 2-3
- Modifying Menu-Role Associations on page 2-6
- Creating Custom Database Roles on page 2-6
- Associating Roles with Menus on page 2-7
- Granting a Custom Role Access to a Custom Module on page 2-8
- Adding a Custom Role to OPA_MENU_ROLES on page 2-8

Viewing Menu-Role Associations

This section contains the following topics:

- Organization of the Menu Module Tree on page 2-3
- Navigating the Menu Modules on page 2-5

To view the activities covered by a particular database role, from the Navigator, expand Developer's Toolkit and select Maintain Menu Modules. In the form, press the Query by Role button for a list of values. Choosing a role causes a display of all activities associated with that role. A complete list of database roles and their relation to menu items can be generated by running the Menu Roles report from the Developer's Toolkit.

Organization of the Menu Module Tree

This section describes the internal structure of the Navigator's menus, and the roles and role associations provided by Oracle Clinical.
Internal Menu Module Structure  All activities accessible through the Navigator are organized in a tree, with the root "OPA". Descending from OPA, a node exists for each installed application. For your installation, there will, at a minimum, be nodes for OCL (for Oracle Clinical activities), OPA (for menus and activities generic to all products of Oracle Health Sciences, formerly known as Oracle Pharmaceutical Applications), and DTK (for the Developer’s Toolkit). In turn, each of node is the parent of other menu nodes, and ultimately of leaf nodes, which correspond to executable modules.

Figure 2–3  Maintain Menu Modules Window

Many executable modules can perform more than one task, so to completely define an activity, there is also a task name and a query-only flag. For instance, the same form module, RXCRCAI, performs both query and maintenance of local, installation, and system reference codelists. Consequently, there are six leaf nodes for this module — one for each combination.

The concatenation of nodes, starting at OPA, ending at the leaf node, and including the task and the query mode, is the internal analog of the Navigator menu path to the activity. For instance, the menu path **OC**, then **Data Entry**, and **Initial Log-In**, plus **Entry** corresponds to the series of nodes OPA:OCL:OCL_DATA_ENTRY:RXCDEMLI, plus the task name INITIAL LOG-IN AND FIRST-PASS ENTRY, and a clear ("no") query-only flag.

Role Association Structure  The access an application user has to each node in the menu-module tree is determined by the database role. Each node of the menu tree has associated with it one or more database roles that are allowed access to that node. A user that is not associated with the appropriate role cannot view its corresponding menu or module. The following examples illustrate how the role associated with a user account affects the access the user is given to different menus:

- To view the OCL application menu, a user's Oracle account must be granted the OCL_ACCESS role. This is typically an automatic grant when an Oracle Clinical account is created, along with CONNECT, RESOURCE, RXCLIN_READ, RXCLIN_MOD, and RXC_ANY.
- When Oracle-defined menu-role associations have not been modified, to see the Data Entry menu option of OCL your account must have one of these roles: RXC_DE; RXC_DE2; RXC_DMGR; RXC_SUPER; or RXC_SUPER_NOGL.
- The Initial Log-In and Entry activity requires the same roles, according to the module-role association created in the database by Oracle. So, in order to run
Initial Log-In and Data Entry, your account needs at least two roles: OCL_ACCESS, and one of: RXC_DE, RXC_DE2, RXC_DMGR, RXC_SUPER, RXC_SUPER_NOGL.

Figure 2–4  Menu Entries for Module Window

Navigating the Menu Modules

To view or modify the roles permitted access to the Oracle Clinical menus and activities, navigate to DTK, then Maintain Menu Modules. A Maintain Menu Modules window opens, as shown in Figure 2–4, "Menu Entries for Module Window", with one entry per top-level menu node in the OPA Navigator menu. The record with a blue mark to its left has focus. Change focus by clicking once anywhere on the record of the node you want to examine.

To drill down into the menu nodes from the currently selected node, click Menu Entries, or double-click anywhere in the node's record. Doing this from the Maintain Menu Modules window brings up a new window, as in Figure 2–4, with a title bar naming the parent node, and with records describing the child nodes of that parent. You can continue to drill down within this window until you reach a leaf. If the record that has focus is a module, you have reached a leaf of the tree and the Menu entries button is disabled, as in Figure 2–5.
Modifying Menu-Role Associations

At any node of the menu-module tree, you can see or modify the database roles associated with the node by pressing the Roles button. This button brings up a Security for task dialog box where the roles enabling access to this node are listed and can be modified. Figure 2–5, "Security for Task Dialog Box" illustrates this process for Initial Log-In and Entry.

You can also query the nodes accessible via a role through the Query by Role button, available in the Maintain Menu Modules and Menu Entries for module windows. If you click on this button, you are prompted for a role (a list of values is available). When you enter a role, all menu-module tree nodes accessible via that role are displayed. The Query Top Menus button returns you to a list of the application menu nodes (Figure 2–5).

Creating Custom Database Roles

This section describes how to create a new database role. This may be required if the database roles that are supplied as part of installation do not fit or cannot be modified to fit your business model.

After you create a new database role, grant it access to menu items (see "Modifying Menu-Role Associations" on page 2-6) and add it to a reference codelist (see "Adding a Custom Role to OPA_MENU_ROLES" on page 2-8).

Menu and module access role names must start with the three-letter designator of the application to which they will apply, as shown in Table 2–1, "Prefixes for Role Name, by Application".
Examples of valid role names are: OCL_CRA, RXCBROWSER, DTK_HELP. The Oracle Clinical Remote Data Capture module has no special prefix; its role names are preceded by RXC.

To create a new database role, you must create the role in the database and explicitly grant all the database privileges required for users with the role to do the tasks you intend, including privileges on the related Oracle Clinical tables.

Log in to SQL*Plus as SYSTEM and enter the following:

```sql
create role role_name;
grant privilege on table to role_name;
```

For information on Oracle Clinical tables, see the Oracle Clinical Technical Reference Manual.

### Creating Custom Roles for Restricting DCI Access

You may want to create additional database roles to use in restricting access to DCIs. There is only one predefined role for investigators: RXC_INV. If you want to hide one investigator’s observations from another’s you need more than one investigator role, for example Neurologist (RXC_NEUR, for example) and Oncologist (RXC_ONC, for example). You can create these two roles, create CRFs that are specific to each of those types of observations, and allow one investigator role access to the DCI corresponding to one CRF and the other investigator role access to the other.

Note the following additional tasks required:

- Define default DCI Access for the role; see "Changing the Default Access to DCIs" on page 1-14. You must do this before any users assigned the role try to log in to RDC.
- Assign the roles to users; see "Granting Additional Database Roles to User Accounts" on page 1-15. Be careful not to assign roles with conflicting DCI access to the same user.

### Associating Roles with Menus

Once a new database role has been created and is accessible, select the Maintain Menu Modules option of the DTK menu to identify those menus and activities to which the role gives a user access.

Navigate to each node in the menu-module tree (see "Modifying Menu-Role Associations" on page 2-6) to which this role should give access, then click the Roles button. This brings up a dialog box where the roles that enable access to the node are listed. Add the new role to the list.
Adding a Custom Role to OPA_MENU_ROLES

Custom roles do not appear in the Menu Roles report until you add them to the OPA_ MENU_ROLES installation reference codelist. To add a custom role to this codelist:

1. Choose DTK, then Maintain all Codelists.
2. Query for the OPA_MENU_ROLES codelist.
3. Insert a new record, and define the short value and long value of the codelist. The long value must match the full name of the new database role exactly, and the short name must be three characters or fewer, and unique in that database. The system uses the short name of the role when it generates the Menu Roles report.

Granting a Custom Role Access to a Custom Module

Use these instructions if you are assigning a custom role to a custom module; see "Adding Menu Items to Oracle Clinical" on page 2-8. This procedure allows you to grant the role access to the module as well as to the individual menu items.

1. Open the appropriate menu module file in Oracle Developer 10g Forms Builder.
2. Connect to the database as RXC.
3. In the Object Navigator, highlight the RXCUSER module (not the menu).
4. In the Menu Security property, add the new role. Use the same name as in the database.
5. Assign your new role to the appropriate menu items as described elsewhere in this section.
6. Save, compile, and distribute the resulting .mmx file.

---

Note: If you want to assign a new role to a standard Oracle Clinical module, refer to the "Modifying Menu-Role Associations" section on page 2-6.

---

Adding Menu Items to Oracle Clinical

You can add your own menu items (Developer modules) to the Users and Roles menu, thus extending the functionality of Oracle Clinical. Since you control the Users and Roles menu, you can preserve changes to it across releases of Oracle Clinical. See the Installation Guide for instructions.

Replace the files rxcuser.mmb and rxcuser.fmb with your own menu and form, which will be what is brought up by choosing Admin, and then Users and Roles.
Both Oracle Clinical and the Remote Data Capture (RDC) option use the discrepancy system, which is described in *Oracle Clinical Conducting a Study*. This section describes the tasks for configuring the discrepancy management system:

- **Mapping Database Roles to User Roles** on page 3-1 (Oracle Clinical and RDC)
- **Specifying User Roles for the Oracle Clinical Discrepancy Database** on page 3-2 (Oracle Clinical only)
- **Customizing Layout Definitions** on page 3-4 (Oracle Clinical only)
- **Customizing Profiles** on page 3-5 (Oracle Clinical only)
- **Customizing Flexfields** on page 3-9 (Oracle Clinical only)
- **Defining Reason Codes for Discrepancies** on page 3-11
- **Defining the Possible Review Statuses for Discrepancies** on page 3-14
- **Defining Resolution Reasons for Discrepancies** on page 3-16
- **Setting Values in the OCL_STATE Local Reference Codelist** on page 3-17
- **Configuring Role-Specific Discrepancy Management for RDC** on page 3-17 (RDC only)
- **Configuring Study and Site Security for Discrepancy Management** on page 3-26 (Oracle Clinical and RDC)
- **Setting Up Data Clarification Forms (DFCs)** on page 3-28 (Oracle Clinical only)
- **Creating Reusable Standard Text for Discrepancies and DCFs** on page 3-34 (Oracle Clinical and RDC)

See the *Oracle Clinical Remote Data Capture Onsite Administrator’s Guide* for information on setting up Study/Site discrepancy management privileges.

### Mapping Database Roles to User Roles

This section applies to all RDC users and to Oracle Clinical users who need to work with discrepancies in Oracle Clinical’s Maintain Discrepancy Database window.

User Roles are important because:

- RDC uses them to define access privileges; see the *Oracle Clinical Remote Data Capture Onsite Administrator’s Guide* for more information.
- You can customize RDC discrepancy management, news, and activities for different user roles; see the *Oracle Clinical Remote Data Capture Onsite Administrator’s Guide* for more information.
You can customize the layout of the Maintain Discrepancy Database window for different roles; see “Customizing Layout Definitions” on page 3-4.

You can customize profiles for different roles; see “Customizing Profiles” on page 3-5.

For these customizations and access privileges to be available to a user, the user must have a database role that is mapped to the relevant user role in the User Group Roles installation reference codelist.

There are five default user roles. You can create additional roles if necessary, and map them in the User Group Roles reference codelist to make them available for customizing features in RDC and Oracle Clinical discrepancy management; see "Creating Custom Database Roles" on page 2-6.

Table 3–1 shows the default mapping of database roles to user group roles. The Long Value is not used.

### Table 3–1  Default Values for the USER GROUP ROLES Codelist

<table>
<thead>
<tr>
<th>Database Role (Short Value)</th>
<th>User Group Name (Long Value)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXC_DMGR</td>
<td>DM</td>
<td>Data management role</td>
</tr>
<tr>
<td>RXC_SUPER</td>
<td>DM</td>
<td>Data management role</td>
</tr>
<tr>
<td>RXC_CRA</td>
<td>CRA</td>
<td>CRA role</td>
</tr>
<tr>
<td>RXC_SITE</td>
<td>SITE</td>
<td>Site user</td>
</tr>
<tr>
<td>RXC_BIOS</td>
<td>BIOSTAT</td>
<td>Biostatistics role</td>
</tr>
<tr>
<td>RXC_QC</td>
<td>QUALITY CONTROL</td>
<td>Quality control role</td>
</tr>
<tr>
<td>RXC_INV</td>
<td>INV</td>
<td>Investigator</td>
</tr>
</tbody>
</table>

Note: The USER GROUPS reference codelist is used in Oracle Clinical discrepancy management only. It contains a subset of the User Group Roles in the USER GROUP ROLES reference codelist and determines which of them are available for use in Oracle Clinical discrepancy management; see Chapter 3, “Configuring Discrepancy Management”.

---

### Specifying User Roles for the Oracle Clinical Discrepancy Database

In the USER GROUPS installation reference codelist, you specify which of the roles mapped to database roles in the USER GROUP ROLES reference codelist will be available for use in customizing aspects of the discrepancy management system in Oracle Clinical, including profiles and layouts for use in the Maintain Discrepancy Database window.

The position of a database role in the codelist is important if users have more than one database role. In the case of a user with more than one role, the system uses the database role closest to the top (seq=1) in the codelist as the default; for example, as the default profile when the user opens the Maintain Discrepancy Database window. The long value is not used.
Assigning Function Privileges and Layouts

You control user group discrepancy function privileges and layout definitions in the User Group Admin window. From the Admin menu, select Discrepancy Mgmt, then select User Group Administration.

The window displays a list of existing user group names and their mapped database roles. Select a user group name to view its defined function privileges and layouts.

Assigning and Removing Function Privileges

In the Function Privileges display, you can add or remove privileges for the selected user group. If a group lacks a particular privilege, the system prevents members of that group from performing the function. (The system reference codelist DISCREPANCY FUNCTIONS controls the list of values; you cannot modify it.)

To add a privilege:

1. Click in an empty row. An ellipsis (...) appears.
2. Click the ellipsis. The list of values appears.
3. Select a privilege and click OK. The system adds the privilege.

To remove a privilege, select it and then select Delete Record from the Data menu.

The following table describes the function privileges.

<table>
<thead>
<tr>
<th>Name</th>
<th>Privilege Granted and Method of Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPD DATA ENTRY</td>
<td>Enables the Update Patient Data Special menu item in the Discrepancy Database window, and allows updates while in the Data Entry subsystem.</td>
</tr>
<tr>
<td>BRWS DATA ENTRY</td>
<td>Enables the Browse Patient Data Special menu item in the Discrepancy Database window for read-only access to patient data.</td>
</tr>
<tr>
<td>MANUAL</td>
<td>Enables the Add Manual button in the Maintain Discrepancy Database window, which allows users to create manual discrepancies and manual header discrepancies.</td>
</tr>
<tr>
<td>DCF PRINT DRAFT</td>
<td>Displays the Draft option in the DCF Print Options window; enables the user to print a draft version of a Data Clarification Form (DCF) report.</td>
</tr>
<tr>
<td>DCF PRINT COPY</td>
<td>Enables the Copy option in the DCF Print Options window; enables the user to print a copy of a DCF.</td>
</tr>
</tbody>
</table>
Assigning Custom Layouts

If you define one or more custom layouts for the Discrepancy Database window (see "Customizing Layout Definitions" on page 3-4, assign it to a user group to allow the user group to use it. Users can select the layout they want to use from the User Group Layouts item in the Special menu of the Discrepancy Database window.

To assign a layout to a user group:

1. Click in an empty row. An ellipsis (...) appears.
2. Click the ellipsis. The list of values appears.
3. Select a layout and click OK. The system adds the layout.

To remove a layout, select it and then select Delete Record from the Data menu.

Customizing Layout Definitions

This section applies to Oracle Clinical only; not RDC.

Oracle Clinical provides the Layout Definitions utility for creating different representations of the Maintain Discrepancy Database window for different user groups. You can then assign layouts to different user groups. A user group can have more than one layout, and a layout can be used by multiple user groups.

A layout definition specifies which discrepancy details appear in the Master, or upper section, of the Maintain Discrepancy Database window in multi-record view, and which appear in the Detail, or lower section. The layout also determines the display location of each discrepancy detail, relative to the others in the same section of the window.

The Master section of the Maintain Discrepancy Database window in multi-record view is displayed above all the discrepancies. The system populates the fields in the upper section with values from the discrepancy that is highlighted in the lower section. All the fields in the Master section are visible without scrolling.

Of the discrepancy details that remain in the lower section of the window, put those that are most useful at the top of the list, so that they are displayed farthest left in the window. They, too, will be visible without scrolling.

To open the Layout Definitions window, from the Admin menu, select Discrepancy Mgmt Admin, then select Layout Definitions. You can also open this form directly from the User Group Admin form by clicking the Layout Definitions button. Users can create their own temporary layout directly in the multi-record view of the Maintain Discrepancy Database window; see "Changing Discrepancy Display Layout and Filtering the Data Displayed" in Oracle Clinical Conducting a Study.

To view the definition of an existing layout, select its name in the top section, labeled "Layout."
To create a new layout:

1. Enter a name for the new Layout in the top section, labeled "Layout." If no line is available, select Data, then select Insert Record.

2. One by one, select the fields you want to move into the other section and move them, using the sideways arrows:
   - Use the Left arrow to move the selected field into the Master, or upper, section of the window in multi-record view.
     The system allows only eight fields in the Master section, specifically: Patient, Visit, Subevent Number, DCM Name, Review Status, DCF ID, CRF Page, and Investigator. To maximize the number of fields visible without scrolling, put all eight fields into the Master section.
   - Use the Right arrow to move the selected field from the Master section into the Detail, or lower, section of the window.

3. Use the Up and Down arrows to adjust the display order of the items. The topmost item in either the Master or Details list appears farthest left in the appropriate section of the Maintain Discrepancy Database window. The second item from the top appears second to the left, and so on.
   In the Details section, the display order determines which fields are visible without scrolling. The number of fields that are visible without scrolling depends on the size of the fields you select.

4. Click OK to save your changes or click Back to return without saving your changes.

Users have access to the same utility for customizing the layout of the Maintain Discrepancy Database form. For a more comprehensive description of this utility, see "Using the Maintain Discrepancy Database Window" in Oracle Clinical Conducting a Study.

Customizing Profiles

This section applies to Oracle Clinical only; not RDC.

To open the Maintain Discrepancy Database form, a user must have a user role with a defined profile. Profiles control access to discrepancy records and fields, review status codes, data entry, layout definitions, and printing and tracking Data Clarification Forms (DCFs).

You can create master profiles for different user roles; individual users can modify their profile when they open the Maintain Discrepancy Database form.

See "Mapping Database Roles to User Roles" on page 3-1 for information about user roles.

To open the Profile Administration window, from the Admin menu, select Discrepancy Mgmt Admin, then select Profile Administration. The Profile Administration window opens in multi-mode view. You can toggle between modes by clicking the View Mode button in the upper right corner of the form.

Filtering Profile Views

The Profile Administration form includes a Profile Criteria section. You can specify the available criteria to suit different audiences. In multi-view mode the criteria are
columns of fields. In single-view mode, these fields are in the middle panel, labeled Profile Criteria.

Specifying Default Profile Criteria
You can refine a profile group’s view by specifying default profile criteria values in the Profile Administration form. Many of the fields have lists of values. If you do not specify a particular criterion, the system allows a user with that profile to query all possible values. If you do not lock a profile criterion, (see the following section), users can override the Master profile’s default settings.

Locking Profile Criteria
Profile users can customize their profiles, and thereby modify their views of the discrepancy database. You control the extent to which they can modify a profile by locking or unlocking each criterion. The locks are boxes that, in single-view mode reside between each criterion name and its field, and in multi-mode view reside to the right of each criterion’s field. (See Figure 3–1 and Figure 3–2 for examples of each view mode, with diagonal arrows indicating the box positions.) The lock prevents users from modifying the default profile settings. If you do not lock a profile criterion, a profile user can override the profile's default setting.

In multi-view mode, for criteria like Accessible Data Only? where toggle boxes control viewing, the Lock box is to the right of the criteria names. (See Figure 3–1 on page 3–6.)

In single-view mode, for criteria like Accessible Data Only? where toggle boxes control viewing, the Lock box is to the left of each criterion’s toggle box. (See Figure 3–2.)

Example 3–1 Profile Administration Form in Multi-View Mode
Figure 3–1 on page 3-6 is a screen capture of a region of the Profile Administration form in multi-view mode. (Many of the intervening criteria fields have been scrolled out of view in order to include the Accessible Data Only? column in the illustration.) The diagonal arrows point to the Lock box columns for the Creation Ts To column’s Lock box, near the top-center of the capture, and the Accessible Data Only? Lock box on the right. The table following Figure 3–1 on page 3-6 describes how the settings in Figure 3–1 control profile criteria viewing for the four listed profiles.

Figure 3–1 Sample Multi-View Region of the Profile Administration Form

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Creation Ts To</th>
<th>Accessible Data Only?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOSTAT Master Profile</td>
<td>12-NOV-2003</td>
<td>■</td>
</tr>
<tr>
<td>CRA Master Profile</td>
<td>12-NOV-2003</td>
<td>■</td>
</tr>
<tr>
<td>DM Master Profile</td>
<td></td>
<td>■</td>
</tr>
<tr>
<td>QUALITY CONTROL Master</td>
<td></td>
<td>■</td>
</tr>
</tbody>
</table>
Table 3-4  Legend for Figure 3-1

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Creation Ts To Access Criterion Access</th>
<th>Accessible Data Only? Criterion Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOSTAT Master Profile</td>
<td>Only records dated before 12-NOV-2003 appear by default, but users can change this date because it is not locked.</td>
<td>Because the left box is not selected, both data types appear by default. But users can limit their view to accessible data because the criterion is not locked (the Lock box to its right is not selected).</td>
</tr>
<tr>
<td>CRA Master Profile</td>
<td>Only records dated before 12-NOV-2003 appear, and users cannot change this date because it is locked.</td>
<td>Because the left box is not selected, only accessible data appears by default. But users can include inaccessible data in their view because the criterion is not locked.</td>
</tr>
<tr>
<td>DM Master Profile</td>
<td>All data appears, regardless of creation date, because the creation date is not specified. But because the criterion is not locked, users can filter by creation date if they choose.</td>
<td>Both data types appear by default, and because this setting is locked, users cannot change it.</td>
</tr>
<tr>
<td>QUALITY CONTROL Master Profile</td>
<td>All data appears, regardless of creation date, because the creation date is not specified. Because the criterion is locked, users cannot filter their view by creation date.</td>
<td>Only accessible data appears by default, and because this setting is locked, users cannot change it.</td>
</tr>
</tbody>
</table>

Example 3-2  Profile Administration Form in Multi-View Mode

Figure 3-2 is depicts a portion of the single-view Profile Administration window. The arrows point to the Lock boxes for, from left to right, the (Creation Ts To) criterion, the Last Modified By criterion, and the Accessible Data Only? criterion.

Figure 3-2  Sample Single-View Region of the Profile Administration Window

In single-view mode, for criteria like Accessible Data Only? where toggle boxes control viewing, the Lock box is to the left of the toggle box.

Adding SQL Statements

In addition to defining access to discrepancy details in the Profile Criteria fields, you can further limit data access by entering SQL statements in the SQL Text section. To prevent users from bypassing the SQL statements, check the locking box for each entry. (In single-view mode the locking box is to the right of the SQL Text label. In multi-view mode, the locking box is to the left of the SQL Text column’s fields.) Your SQL statements have a size limit of 2000 characters.

Filtering Profile Views by Review Status

You can limit a profile’s view of discrepancy records according to the records’ current review status classifications. Select a profile in the Profile Administration form, then click the Review Status button to open the Discrepancy Review Status Codes dialog box. (To define the codes you see in this box, see “Defining the Possible Review Statuses for Discrepancies” on page 3-14.) The dialog box contains rows with three
columns: the **Selected?** box column, the **Status Name** column and the **Locked?** box column.

Each **Selected?** box controls a profile's default access to records that have that review status classification code. Select a code's **Selected?** box to include discrepancy records; deselect it to prevent viewing records with its code.

Users can override each of these settings in their personal profiles unless you lock them. If you lock a review status code, profile users cannot override the profile's default setting.

*Table 3–5 on page 3-8 describes the outcome for the four possible combinations of **Selected?** and **Locked?** boxes:*

<table>
<thead>
<tr>
<th>Selected?</th>
<th>Locked?</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unchecked</td>
<td>Unchecked</td>
<td>Not viewed by default, but profile user can reveal it.</td>
</tr>
<tr>
<td>Checked</td>
<td>Unchecked</td>
<td>Viewed by default, but the profile user can hide it.</td>
</tr>
<tr>
<td>Unchecked</td>
<td>Checked</td>
<td>Not viewed by default, and profile user cannot reveal it.</td>
</tr>
<tr>
<td>Checked</td>
<td>Checked</td>
<td>Viewed by default, and profile user cannot hide it.</td>
</tr>
</tbody>
</table>

**Updating Status Codes**

If the values in the DISCREPANCY REV STATUS CODE reference codelist change, you can update all profiles with the new values by clicking the **Add Review Status** button. (To edit the codelist, see "Defining the Possible Review Statuses for Discrepancies" on page 3-14.)

**Filtering Profile Views by Discrepancy Field**

You can control a profile's access to discrepancy review status types and individual discrepancy fields by setting its privileges. By default, a new profile has no update privileges; you must add review status codes, and then discrepancy field values to each master profile. To open the Privileges for DM Master Profile window, from the Admin menu, select Discrepancy Mgmt Admin, and then select Profile Administration. Select a master profile, then click the Privileges button.

**Adding Update Privileges by Review Status Code** In the Update Discrepancy records... Review Status column, add the discrepancy record review status types that you want to be accessible by users of the currently selected profile. You can select from an list of values. (To define the review status codes that appear in this dialog box, see "Defining the Possible Review Statuses for Discrepancies" on page 3-14.)

**Note:** Adding an update code does not make its discrepancy fields accessible. You must also specify each field. See the following section.

**Setting Update Privileges by Discrepancy Field** In the Update Discrepancy Fields panel there are two columns: Field Name and Privilege. Add the field names for fields that users of this profile can update. You can select them from an list of values. In each corresponding Privilege field, add the type of privilege. If you do not add a field name, users of the profile cannot update the data in the Maintain Discrepancy Database form. You can select from lists of values for both fields. *Table 3–6 shows the...*
corresponding field names for each field status value. (The Update Discrepancy Fields section contains the list of fields the profile has access to change if the current discrepancy record’s review status is on the Update Discrepancy Records list. For Oracle Clinical Release 4.6, the only allowable value for the Privilege field is Update.)

**Table 3–6  Field Status Values**

<table>
<thead>
<tr>
<th>Field Status Value</th>
<th>Field in the Maintain Discrepancy Database Form or Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REVIEW STATUS</td>
<td>Review Status</td>
</tr>
<tr>
<td>RESOLUTION STAT</td>
<td>Resolution Status</td>
</tr>
<tr>
<td>COMMENT TEXT</td>
<td>Comment Text</td>
</tr>
<tr>
<td>INT COMMENTS</td>
<td>Internal Comments</td>
</tr>
<tr>
<td>RESOLUTION TEXT</td>
<td>Resolution Text</td>
</tr>
<tr>
<td>ASSOCIATED ID</td>
<td>Associate a discrepancy for passive review field</td>
</tr>
<tr>
<td>CRF PAGE NUMBER</td>
<td>CRF Page Number</td>
</tr>
<tr>
<td>FLEXFIELD1</td>
<td>User-definable field (See “Customizing Flexfields” on page 3-9.)</td>
</tr>
<tr>
<td>FLEXFIELD2</td>
<td>User-definable field (See “Customizing Flexfields” on page 3-9.)</td>
</tr>
</tbody>
</table>

**Setting Review Status Updating Privileges**

The Can Change Review Status To section contains the allowable values for the review status if it can be updated. The Own Manual Only flag identifies allowable values for the Review Status field for manual discrepancy records owned by the same group as the current user. For example, you could specify that users of the CRA profile could only close discrepancies that they manually created.

---

**Note:** If users can choose an IRRESOLVABLE status, they must also have access to the Resolution Status; whenever the status is Irresolvable, the Resolution Status is required. Otherwise users cannot enter a value for lack of privileges.

---

**Customizing Flexfields**

The Maintain Discrepancy Database window includes two editable fields that have the default labels Flexfield1 and Flexfield2. You can use them to store information you need. You can change their labels, make the fields mandatory if you want, and create a dynamic or static list of values for each field:

- **Static List of Values.** By default, the fields get their list of values from a database view that references the installation reference codelists DISC_FLEX1_VALUES and DISC_FLEX2_VALUES. You can populate these reference codelists to create lists of values for the two fields.

- **Dynamic List of Values.** Alternatively, you can reprogram the view to reference another Oracle Clinical table or even an Oracle table outside Oracle Clinical, such as an adverse event code maintained in a different Oracle system. In each field you can display any column value or concatenation of column values.
The views, DISCREP_FLEX1 and DISCREP_FLEX2, respectively, are created during installation as follows:

```sql
create or replace view discrep_flex1 as
select ref_codelist_value_short_val VALUE,
       long_value DESCRIPTION
from reference_codelist_values
where ref_codelist_name = 'DISC_FLEX1_VALUES'
and active_flag='Y';
```

**DISC_FLEX1 and DISC_FLEX2**

Use these local reference codelists to customize the label for Flexfield1 or Flexfield2, to enable the field, make the field mandatory or not, and to specify whether or not there is a list of values for the field.

**DISC_FLEX1_VALUES and DISC_FLEX2_VALUES**

If you set LOV_VALIDATE to Y in DISC_FLEX1, you can create a static list of values by entering each allowed value in a row in this codelist. When a user displays the list of values in the Discrepancy Database window, the system displays the short value and description for each row you enter here. (Alternatively, create a dynamic list of values; see "Customizing Flexfields" on page 3-9.)

The long value is used as the description in the Flexfield1 (or 2) field's list of values. The short value is stored in the FLEX_FIELD1 (or 2) column in the DISCREPANCY_ENTRIES table in the database. The default value has no effect.

**Table 3–7 DISC_FLEX1 and DISC_FLEX2 Settings**

<table>
<thead>
<tr>
<th>Short Value</th>
<th>Long Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Y or N</td>
<td>Set the long value to Y to allow users to enter values in the field. Set to N to prevent users from entering values in the field.</td>
</tr>
<tr>
<td>required</td>
<td>Y or N</td>
<td>Set the long value to Y to require users to enter text in this field. Set to N to make the field optional.</td>
</tr>
<tr>
<td>prompt</td>
<td>Y or N</td>
<td>Enter freeform text as the long value. This text becomes the flexfield's label in the Discrepancy Database window.</td>
</tr>
<tr>
<td>lov_validate</td>
<td>Y or N</td>
<td>Set the long value to Y to require the system checks the entry against an list of values. An invalid entry triggers the system to display the associated list of values.</td>
</tr>
</tbody>
</table>

**Table 3–8 DISC_FLEX1_VALUES and DISC_FLEX2_VALUES Codelists**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq</td>
<td>Determines the order of the values in the flexfields’ lists of values.</td>
</tr>
<tr>
<td>Short Value</td>
<td>Is the stored value when the user selects the row from the list of values.</td>
</tr>
<tr>
<td>Long Value</td>
<td>Can duplicate the short value; not displayed in the list of values.</td>
</tr>
<tr>
<td>Active box</td>
<td>Check an entry’s Active box to have the entry appear in the list of values.</td>
</tr>
<tr>
<td>Default</td>
<td>The default value.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the value; displayed in the list of values.</td>
</tr>
</tbody>
</table>
Example

To customize Flexfield1 to ask the creator of a manual discrepancy record if the record is finished, and test the response entry against a list of three values, you would set the flexfield1 codelist values to the values in the following two tables, with the three values defined in the second. (Values that differ from the default values are in italics):

### Table 3–9 DISC_FLEX1 Settings to Make a Finished? Prompt

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Active</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENABLED</td>
<td>Y</td>
<td>3</td>
<td></td>
<td>Flexfield1 Displayed</td>
</tr>
<tr>
<td>2</td>
<td>REQUIRED</td>
<td>N</td>
<td>3</td>
<td></td>
<td>Flexfield1 Required</td>
</tr>
<tr>
<td>3</td>
<td>PROMPT</td>
<td>Finished?</td>
<td>3</td>
<td></td>
<td>Flexfield1 Prompt</td>
</tr>
<tr>
<td>4</td>
<td>LOV_VALIDATE</td>
<td>Y</td>
<td>3</td>
<td></td>
<td>Flexfield1 list of values Validate</td>
</tr>
</tbody>
</table>

### Table 3–10 DISC_FLEX1_VALUES Settings for a Finished? List of Values

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Active</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Y</td>
<td>Yes</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>No</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>U</td>
<td>Undetermined</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

### Defining Reason Codes for Discrepancies

This section contains the following topics:

- "Reason Codes and Descriptions for Manual Discrepancies"
- "Reason Codes and Descriptions for Univariate Discrepancies"
- "Reason Codes and Descriptions for Multivariate Discrepancies"

You define reason codes to separate discrepancies into categories. Reasons provide an explanation of why the discrepancy exists. They are called Reasons in RDC and Category in Oracle Clinical.

### Reason Codes and Descriptions for Manual Discrepancies

When a user creates an Operator Comment (a manual field or section discrepancy in RDC Onsite), the system prompts the user to select a reason code from a list of reasons that is populated by the MANUAL SOURCE TYPE CODE reference codelist in Oracle Clinical. The user can also enter an additional explanation for the discrepancy.

You can add and remove values in the reference codelist (see Figure 3–3):

- For each reason you add, enter a value in the Short Value field and the Description field.
- Set one reason to Default. The first time the user creates a manual discrepancy during a login session, the system inserts the default reason. The user can select a different reason. For subsequent manual discrepancies, RDC Onsite displays the last entered reason. The user can always select a different reason.

Oracle Clinical stores the short value in the database.
Defining Reason Codes for Discrepancies

The system creates a univariate discrepancy when the data entered does not match the requirements defined in the underlying question, such as data type or length. You cannot change these underlying causes, but you can change the reason text or define more than one reason associated with a single underlying cause, from which the user can choose.

The system automatically populates the reason code and description when it raises the discrepancy.

To define reason codes and descriptions for univariate discrepancies:

1. Open Oracle Clinical.

### Table 3–11  Values for MANUAL SOURCE TYPE CODE Reference Codelist

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>STUDY ASSUMP</td>
<td></td>
<td>Study assumption</td>
</tr>
<tr>
<td>2</td>
<td>CRA COMMENT</td>
<td></td>
<td>CRA Correction, Investigator consulted</td>
</tr>
<tr>
<td>3</td>
<td>CDA COMMENT</td>
<td></td>
<td>CRA Correction, Source Data consulted</td>
</tr>
<tr>
<td>4</td>
<td>DE COMMENT</td>
<td></td>
<td>Data Entry Comment</td>
</tr>
<tr>
<td>5</td>
<td>SOURCE DATA REV</td>
<td></td>
<td>Source Data Review</td>
</tr>
</tbody>
</table>

### Figure 3–3  Defining Reasons for a Manual Discrepancy
2. Select Admin, Discrepancy Mgmt Admin, and then select Standard Text Maintenance. See Figure 3–4.

You use the fields in the Standard Text Maintenance form to define descriptions for each type of univariate discrepancy as follows:

- **Text Type** — Select COMMENT to specify descriptions for univariate discrepancies.

- **Sub Type** — Select from the list of valid univariate discrepancy types. RDC Onsite uses the text in the Sub Type field to populate the Reason field when a discrepancy of that type is raised.

- **Standard Text Name** — Ordinarily, you specify a name to match the Sub Type. However, if you want to specify more than one description for a discrepancy type, use this field to specify a unique identifier.

- **Default** — If you choose to create a selection of multiple descriptions for a certain discrepancy type, select which description is the default value. RDC Onsite uses the default value the first time a discrepancy of that type is raised. The user can always select one of the alternative descriptions provided.

- **Standard Text** — Specify the description that you want RDC Onsite to display for the discrepancy type raised. Note that you can use variables to include the data value of the discrepant response as well as the definitional components of the question; for example:

  Value of \VALUE_TEXT\ for \SAS_LABEL\ is not a valid \DATA_TYPE\.

where VALUE_TEXT is the data response entered, SAS_LABEL is the SAS label defined for the Question, and DATA_TYPE is the data type defined for the Question.
Defining the Possible Review Statuses for Discrepancies

When a user takes action on a discrepancy, the discrepancy goes to a new review status. For system-generated discrepancies (univariate and multivariate), the system assigns the default review status. For manual discrepancies, the user selects the review status.

Both the Oracle Clinical and RDC discrepancy management systems use the DISCREPANCY REV STATUS CODE installation reference codelist to define all the discrepancy statuses possible to use in your discrepancy management workflow.

RDC Onsite uses the text string in the Description field to display the status of a discrepancy in any discrepancy management-related window, form, page, or report. Oracle Clinical uses the text string in the Short Value field to display the status of a discrepancy.

Reason Codes and Descriptions for Multivariate Discrepancies

Oracle Clinical raises multivariate discrepancies when user-defined validation procedures detect invalid or inconsistent data. You specify the reason for multivariate discrepancies in the procedure definition; see Oracle Clinical Creating a Study for details. Oracle Clinical displays the reason; RDC Onsite does not.

Figure 3–4 Defining Descriptions for Univariate Discrepancies

<table>
<thead>
<tr>
<th>Text Type</th>
<th>Sub Type</th>
<th>Standard Text Name</th>
<th>Default</th>
<th>Standard Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMENT</td>
<td>DATA TYPE</td>
<td>DATA TYPE</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>DVG</td>
<td>DVG</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>DVG SUBSET</td>
<td>DVG SUBSET</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>INDICATOR NO</td>
<td>INDICATOR NO</td>
<td></td>
<td>Data was entered but \SAE_LAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>INDICATOR YES</td>
<td>INDICATOR YES</td>
<td></td>
<td>No data was entered but \SAE_</td>
</tr>
<tr>
<td>COMMENT</td>
<td>LENGTH</td>
<td>LENGTH</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>LOWERBOUND</td>
<td>LOWERBOUND</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>MANDATORY</td>
<td>MANDATORY</td>
<td></td>
<td>Value for \SAE_LABEL\ has not</td>
</tr>
<tr>
<td>COMMENT</td>
<td>MISSING_PT</td>
<td>MISSING_PT</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>MISSING_SCT</td>
<td>MISSING_SCT</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>PARTIAL_DATE</td>
<td>PARTIAL_DATE</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>PRECISION</td>
<td>PRECISION</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
<tr>
<td>COMMENT</td>
<td>THESAURUS</td>
<td>THESAURUS</td>
<td></td>
<td>Value of \VALUE_TEXT\ for \SAE\</td>
</tr>
</tbody>
</table>
You can edit the review status codes available and sequence the order in which they appear in lists of values. If you add a status, you must do the following as well:

- **In Oracle Clinical**, update all profiles with the new status by clicking the **Add Review Status** button; see “Customizing Profiles” on page 3-5.
- **In RDC**, add the new status' short value to each DISCREPANCY STATUS ROLE codelist. In addition, if you want users of a particular role to be able to route discrepancies to the status, add the status to the relevant DISCREPANCY ACTIONS ROLE codelist. See "Configuring Discrepancy Display by User Role" on page 3-18 and "Configuring the Actions Allowed on Discrepancies" on page 3-23.

### Default Entries for the DISCREPANCY REV STATUS CODE Codelist

Table 3–12 lists the entries in the DISCREPANCY REV STATUS CODE codelist following the initial installation of Oracle Clinical.

<table>
<thead>
<tr>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRA REVIEW</td>
<td>Null</td>
<td>Under CRA Review</td>
</tr>
<tr>
<td>INV REVIEW</td>
<td>Null</td>
<td>Under Investigator Review</td>
</tr>
<tr>
<td>RESOLVED</td>
<td>IRRESOLVABLE</td>
<td>Closed - resolved</td>
</tr>
<tr>
<td>IRRESOLVABLE</td>
<td>IRRESOLVABLE</td>
<td>Irresolvable (That is, closed. Cannot be resolved.)</td>
</tr>
<tr>
<td>TMS EVALUATION</td>
<td>TMS EVALUATION</td>
<td>TMS Evaluation</td>
</tr>
<tr>
<td>UNREVIEWED</td>
<td>Null</td>
<td>Not yet reviewed</td>
</tr>
<tr>
<td>TMS IN PROGRESS</td>
<td>TMS IN PROGRESS</td>
<td>TMS in Progress - Set/Reset by system</td>
</tr>
<tr>
<td>DM REVIEW</td>
<td>Null</td>
<td>Under DM Review</td>
</tr>
<tr>
<td>INT DM REV</td>
<td>Null</td>
<td>Internal - Under DM Review</td>
</tr>
<tr>
<td>INT CRA REV</td>
<td>Null</td>
<td>Internal - Under CRA Review</td>
</tr>
</tbody>
</table>

### Rules for the DISCREPANCY REV STATUS CODE Codelist

When entering and modifying values in the DISCREPANCY REV STATUS CODE reference codelist, you must follow these rules:

- The codelist must contain at least one short value entry with a long value of IRRESOLVABLE, which indicates that a user must specify a resolution reason when setting a discrepancy to this status. By default, the codelist contains the short values RESOLVED and IRRESOLVABLE, which are each assigned the long value IRRESOLVABLE.
- The codelist must contain a short value entry of UNREVIEWED. The Active check box corresponding to the value must always be selected. The UNREVIEWED value is the system-coded default status of any newly created discrepancy, except multivariate discrepancies, whose initial review status is defined in the Details block of the Oracle Clinical validation procedure that generates the discrepancy.
- The codelist must **NOT** contain a short value entry of CLOSED. The system reserves the CLOSED value as the status for any system-resolved discrepancy, that is, a data discrepancy that was resolved as the result of an update to a non-discrepant value.
The Description field must be entered for each short value. In RDC, the system uses this value to display the status of discrepancies in any discrepancy management-related windows, forms, tasks tabs, or reports. In Oracle Clinical, the system uses the short value to describe the status of a discrepancy.

The Active checkbox must be selected for each review status that you want to allow for a certain user role. In other words, if you set an entry in a DISCREPANCY STATUS role codelist to active, then you must also set the corresponding entry in the DISCREPANCY REV STATUS CODE codelist to active. Conversely, if you set an entry in the DISCREPANCY REV STATUS CODE codelist to inactive, you must also set the corresponding entry in each DISCREPANCY STATUS role codelist to inactive. If not, users receive an error message that problems exist with the discrepancy management system.

**Defining Resolution Reasons for Discrepancies**

Users can manually resolve discrepancies. When doing so, the user must also provide an explanation — a reason — for resolving the discrepancy. The user must select a resolution reason from the list that you define in the DISCREPANCY RESOLU TYPE CODE installation reference codelist.

Enter values in the Description field and the Short Value field. The system displays the text string that you specify in the Description field to display the list of resolution reasons to the user. The corresponding short value is stored in the database.

---

**Note:** Oracle reserves the CND BLK DELETED value as the resolution for a manual discrepancy associated with a conditional block that was deleted. The value is hard-coded. Do not add this value to the DISCREPANCY RESOLU TYPE CODE codelist.

---

Table 3–13 lists the default entries in the DISCREPANCY RESOLU TYPE CODE codelist.

<table>
<thead>
<tr>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRA VERIFY</td>
<td>CONFIRMED</td>
<td>CRA Correction</td>
</tr>
<tr>
<td>CRA VERIFY-INV</td>
<td>CONFIRMED</td>
<td>CRA Correction, Investigator consulted</td>
</tr>
<tr>
<td>CRA VERIFY-SRC</td>
<td>CONFIRMED</td>
<td>CRA Correction, Source Data consulted</td>
</tr>
<tr>
<td>INV VERIFY</td>
<td>CONFIRMED</td>
<td>Investigator Correction</td>
</tr>
<tr>
<td>STUDY ASSUMP</td>
<td>CONFIRMED</td>
<td>Study Assumption</td>
</tr>
<tr>
<td>NO ACTION REQD</td>
<td>CONFIRMED</td>
<td>No Action Required</td>
</tr>
<tr>
<td>ELIMINATED</td>
<td>IRRESOLVABLE</td>
<td>Data value changed. Disc no longer applicable.</td>
</tr>
<tr>
<td>OVERRULED</td>
<td>NON DISCREPANT</td>
<td>Disc not considered a validation error.</td>
</tr>
<tr>
<td>DATA MODIFIED</td>
<td>IRRESOLVABLE</td>
<td>Data value changed. Disc no longer applicable.</td>
</tr>
<tr>
<td>INV-NO INFO</td>
<td>IRRESOLVEABLE</td>
<td>Investigator queried. No further information available.</td>
</tr>
</tbody>
</table>
If you add a value, select a long value from the following:

- NULL: No value entered.
- CONFIRMED
- IRRESOLVABLE: Used for values which correspond to manually closed discrepancies. This makes the RESOLUTIONTYPE CODE and COMMENT available when user selects an IRRESOLVABLE status.
- NOT DISCREPANT. Used for manual discrepancies only; indicates that the discrepancy was raised for a comment, not because there was a problem with the data.

Setting Values in the OCL_STATE Local Reference Codelist

The local reference codelist OCL_STATE includes several parameters that control discrepancy database and DCF functions, as well as many other parameters. The following table describes the discrepancy management parameters.

**Table 3–14 Codelist OCL_STATE Settings for Discrepancy Database Functions**

<table>
<thead>
<tr>
<th>Short Value (Parameter Name)</th>
<th>Long Value (Settings)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISC_DCM_PROMPT</td>
<td>default_prompt or question_name or sas_label</td>
<td>The single-record mode of the Maintain Discrepancy Database form’s Characteristics panel has a button that toggles the three long value settings. Choose the value to display by default.</td>
</tr>
<tr>
<td>DCF_TEXT_SYNC</td>
<td>y or n</td>
<td>If set to y, changes to comment and resolution text entries in DCFs automatically propagate to the discrepancy database</td>
</tr>
<tr>
<td>DISC_AUTO_HDFT</td>
<td>y or n</td>
<td>If set to y, the system automatically populates header and footer text fields with the default values set in the Standard Text Admin form.</td>
</tr>
<tr>
<td>DISC_AUTOR_CRFPG</td>
<td>y or n</td>
<td>If set to y, the system automatically populates the CRF Page Number field of the Maintain Discrepancy Database form.</td>
</tr>
</tbody>
</table>

Configuring Role-Specific Discrepancy Management for RDC

This section includes the following topics:

- Defining Reason Codes for Discrepancies
- Configuring Discrepancy Display by User Role
- Configuring the Actions Allowed on Discrepancies
- Preventing Update to OTHER Discrepancies

You use the reference codelists in Oracle Clinical to define your discrepancy management system for both Oracle Clinical and RDC Onsite.

To configure most settings for your discrepancy management system, you use the installation reference codelists in Oracle Clinical.

To access the installation reference codelists:

1. Open Oracle Clinical.
2. Select Admin, Reference Codelists, and then select Installation Codelists.

See Chapter 7, "Reference Codelists" for general information on setting reference codelist values.
Configuring Discrepancy Display by User Role

This section contains the following topics, which apply only to RDC:

- How RDC Indicates Discrepancies in the User Interface on page 3-19
- Rules for the DISCREPANCY STATUS role Codelists on page 3-20
- Comparison of the Default Values for the DISCREPANCY STATUS role Codelists on page 3-21
- DISCREPANCY STATUS CRA on page 3-21
- DISCREPANCY STATUS DM on page 3-22
- DISCREPANCY STATUS INV on page 3-22
- DISCREPANCY STATUS SITE on page 3-22

RDC uses the DISCREPANCY STATUS role installation reference codelists to determine how discrepancies with a particular status are displayed for users with a particular role. There is a different reference codelist for each default user role: CRA, DM, INV, and SITE.

Use these codelists to ensure that discrepancies are displayed appropriately for users of different roles. For example, a discrepancy with a review status of Under CRA Review should appear as ACTIVE to a CRA, but as OTHER to an investigator.

---

**Note:** These reference codelists do not determine what actions a user can perform on discrepancies or their underlying patient data. The DISCREPANCY ACTIONS role codelists determine what routing and resolution actions each user role can take on a discrepancy.

---

**Note:** If you create additional roles for use in discrepancy management you must create a new reference codelist called DISCREPANCY STATUS role for each of them and set it up the same way that these reference codelists are set up.

---

Every status defined in the DISCREPANCY REV STATUS CODE codelist must be included in the DISCREPANCY STATUS role codelist for each role, with a long value that determines how the discrepancy is presented (or not) to the user. The possible long values are:

- **ACTIVE:** (For open discrepancies) The current user can take action against this discrepancy.
- **OTHER:** (For open discrepancies) The discrepancy is assigned to a user with a different role. For RDC Onsite, you can prevent users from taking action on OTHER discrepancies with the reference codelist DISCREPANCY NO OTHER UPDATE; see "Preventing Update to OTHER Discrepancies" on page 3-25.
- **CLOSED:** (For closed discrepancies) System-closed discrepancies cannot be re-opened by users with any role. If the discrepancy was manually closed by a user, any user with UPDATE or UPD_DISCREP privilege can re-open the discrepancy.
- **HIDDEN:** (For open discrepancies) The current user cannot view or take action against this discrepancy. This functionality is intended only for section-type discrepancies. If a user selects a univariate or multivariate discrepancy, the Action
drop-down list excludes any action that would route the discrepancy to a status that is HIDDEN for any role.

There is another technique for hiding any type of discrepancy (section, univariate, multivariate) at a particular status (for example Internal DM Review) from a particular user role (for example SITE). In this example, simply uncheck the 'Active' checkbox for the status Internal DM Review in the codelist DISCREPANCY STATUS SITE.

**Note:** If you change the long value to HIDDEN for a status that has already been applied to univariate or other types of discrepancies, these existing discrepancies are hidden to users with the relevant role, even though only section discrepancies are intended to allow hiding.

**Note:** Long values must be in uppercase.

### How RDC Indicates Discrepancies in the User Interface

RDC uses the settings of these codelists to indicate to the current user whether a CRF, patient, or individual response is associated with a discrepancy and if so, whether it is a discrepancy that requires action by the current user. The table below describes the colors that RDC uses for this purpose. RDC uses these colors to highlight the patient icon, the CRF icon, and the individual fields in a CRF that have one or more discrepancies.

Note that:

- HIDDEN discrepancies are not highlighted in any color because they are not visible to particular user roles.
- RDC uses green to highlight the fields in a CRF that have a discrepancy that was manually closed by the user. Fields with a system-closed discrepancy are not highlighted.

**Table 3–15  Colors Used to Indicate Discrepancy Access Status**

<table>
<thead>
<tr>
<th>Color</th>
<th>Access Status</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>ACTIVE</td>
<td>The CRF contains at least one open discrepancy that requires attention by the user role to which the current user is assigned.</td>
</tr>
<tr>
<td>Yellow</td>
<td>OTHER</td>
<td>The CRF contains only open discrepancies that require the attention of a user role different from the one to which the current user is assigned.</td>
</tr>
<tr>
<td>White</td>
<td>CLOSED</td>
<td>The CRF contains no visible open discrepancies. Three conditions may be true for a white CRF or patient icon:</td>
</tr>
</tbody>
</table>

- The CRF may contain discrepancies that are hidden from the current user’s user role.
- The CRF may have contained discrepancies at one time, but all discrepancies are closed or obsolete.
- The CRF never had any discrepancies.
Rules for the DISCREPANCY STATUS role Codelists

When entering and modifying values in a DISCREPANCY STATUS role codelist, you must follow these rules:

- Each DISCREPANCY STATUS role codelist must include all short values that appear in the DISCREPANCY REV STATUS CODE codelist, and must not contain any values not in that codelist. When you add a short value to one codelist, you must add the same short value to the other codelist. (An exception is the CLOSED status, described below).
- Each codelist must contain the short value CLOSED, which has a corresponding long value CLOSED. This is the status that is used for any system-resolved discrepancy, that is, a data discrepancy that was resolved as the result of an update to a non-discrepant value.
- The RESOLVED and IRRESOLVABLE short values must have a long value of CLOSED for all roles.
- If a review status is CLOSED for one user role it must be either CLOSED or HIDDEN for all other user roles. Note also that for any review status that appears as CLOSED in a DISCREPANCY STATUS role codelist, the DISCREPANCY REV STATUS CODE codelist must represent the review status with a long value of IRRESOLVABLE.
- Do not change any long value with a default value of CLOSED.

---

**Note:** Both RDC Classic and RDC Onsite use these reference codelist values. RDC Classic does not support the CLOSED display status for any review status except RESOLVED and IRRESOLVABLE.

---

- The setting of the Active check box determines whether discrepancies of that status are visible to users with the relevant role. If the Active check box is not selected, users with the role cannot see discrepancies of that status. If the Active check box is selected, users with the role can see discrepancies of that status. The way discrepant values are displayed depends on the long value.

---

**Note:** Either disabling the Active check box or setting the long value to HIDDEN has the effect of hiding discrepancies of the relevant status from users with the relevant role. However, use of the text string ‘HIDDEN’ only works for hiding section discrepancies. The technique of unchecking the ‘Active’ checkbox can be used to hide any type of discrepancy: section, manual field, univariate, or multivariate.

---

- If an entry in a DISCREPANCY STATUS role codelist is active, then the corresponding entry in the DISCREPANCY REV STATUS CODE codelist must also be active. If not, the discrepancy configuration is invalid. RDC Onsite will display an error message to alert users to the problem.
- The Description field is optional for all entries.
- The setting of the Default check box has no effect. The default status of a new discrepancy is always UNREVIEWED.
Comparison of the Default Values for the DISCREPANCY STATUS role Codelists

Table 3–16 provides a comparison of how each default discrepancy status is displayed by default for each user role.

### Table 3–16  User Roles and the Default RDC Onsite Discrepancy Access Statuses

<table>
<thead>
<tr>
<th>Short Value – Oracle Clinical Discrepancy Review Status</th>
<th>Long Value – RDC Onsite Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNREVIEWED</td>
<td>CRA: ACTIVE DM: ACTIVE INV: ACTIVE SITE: ACTIVE</td>
</tr>
<tr>
<td>CRA REVIEW</td>
<td>CRA: ACTIVE DM: OTHER INV: OTHER SITE: OTHER</td>
</tr>
<tr>
<td>INV REVIEW</td>
<td>CRA: OTHER DM: OTHER INV: ACTIVE SITE: OTHER</td>
</tr>
<tr>
<td>DM REVIEW</td>
<td>CRA: OTHER DM: ACTIVE INV: OTHER SITE: OTHER</td>
</tr>
<tr>
<td>TMS EVALUATION</td>
<td>CRA: OTHER DM: OTHER INV: OTHER SITE: OTHER</td>
</tr>
<tr>
<td>TMS IN PROGRESS</td>
<td>CRA: OTHER DM: OTHER INV: OTHER SITE: OTHER</td>
</tr>
<tr>
<td>RESOLVED</td>
<td>CRA: CLOSED DM: CLOSED INV: CLOSED SITE: CLOSED</td>
</tr>
<tr>
<td>IRRESOLVABLE</td>
<td>CRA: CLOSED DM: CLOSED INV: CLOSED SITE: CLOSED</td>
</tr>
<tr>
<td>CLOSED</td>
<td>CRA: CLOSED DM: CLOSED INV: CLOSED SITE: CLOSED</td>
</tr>
<tr>
<td>INT CRA REV</td>
<td>CRA: ACTIVE DM: OTHER INV: HIDDEN SITE: HIDDEN</td>
</tr>
<tr>
<td>INT DM REV</td>
<td>CRA: OTHER DM: ACTIVE INV: HIDDEN SITE: HIDDEN</td>
</tr>
<tr>
<td>INT RESOLVED</td>
<td>CRA: CLOSED DM: N/A INV: CLOSED SITE: CLOSED</td>
</tr>
</tbody>
</table>

**DISCREPANCY STATUS CRA**

This codelist contains discrepancy status groupings for the CRA role.

### Table 3–17  Values for the DISCREPANCY STATUS CRA Reference Codelist

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short value</th>
<th>Long value</th>
<th>Active Check box</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UNREVIEWED</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>CRA REVIEW</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>INV REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>DM REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>TMS EVALUATION</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>TMS IN PROGRESS</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>RESOLVED</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>IRRESOLVABLE</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>CLOSED</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>10</td>
<td>INT CRA REV</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
<tr>
<td>11</td>
<td>INT DM REV</td>
<td>OTHER</td>
<td>Y</td>
</tr>
</tbody>
</table>
DISCREPANCY STATUS DM
This codelist contains discrepancy status groupings for the Data Management role.

Table 3–18  Values for the DISCREPANCY STATUS DM Reference Codelist

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short value</th>
<th>Long value</th>
<th>Active Check box</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UNREVIEWED</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>CRA REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>INV REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>DM REVIEW</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>TMS EVALUATION</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>TMS IN PROGRESS</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>RESOLVED</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>IRRESOLVABLE</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>CLOSED</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>10</td>
<td>INT CRA REV</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>11</td>
<td>INT DM REV</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
</tbody>
</table>

DISCREPANCY STATUS INV
This codelist contains discrepancy status groupings for the INVESTIGATOR role.

Table 3–19  Values for the DISCREPANCY STATUS INV Reference Codelist

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short value</th>
<th>Long value</th>
<th>Active Check box</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UNREVIEWED</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>CRA REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>INV REVIEW</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>DM REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>TMS EVALUATION</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>TMS IN PROGRESS</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>RESOLVED</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>IRRESOLVABLE</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>CLOSED</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>10</td>
<td>INT CRA REV</td>
<td>HIDDEN</td>
<td>Y</td>
</tr>
<tr>
<td>11</td>
<td>INT DM REV</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>11</td>
<td>INT RESOLVED</td>
<td>CLOSED</td>
<td>N</td>
</tr>
</tbody>
</table>

DISCREPANCY STATUS SITE
This codelist contains discrepancy status groupings for the SITE role.

Table 3–20  Values for the DISCREPANCY STATUS SITE Reference Codelist

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short value</th>
<th>Long value</th>
<th>Active Check box</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UNREVIEWED</td>
<td>ACTIVE</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>CRA REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
</tbody>
</table>
Configuring Role-Specific Discrepancy Management for RDC

Configuring the Actions Allowed on Discrepancies

In RDC Onsite, a user changes the review status of a discrepancy by selecting an option from the list in the Action field. You use the DISCREPANCY ACTIONS role codelists to define the set of routing and resolution actions that each user role can take against discrepancies with a particular status.

Each DISCREPANCY ACTIONS role codelists specifies allowed actions for one of the default user roles:

- DISCREPANCY ACTIONS CRA
- DISCREPANCY ACTIONS DM
- DISCREPANCY ACTIONS INV
- DISCREPANCY ACTIONS SITE

Only RDC uses the set of DISCREPANCY ACTIONS role codelists.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short value</th>
<th>Long value</th>
<th>Active Check box</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>INV REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>DM REVIEW</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>TMS EVALUATION</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>TMS IN PROGRESS</td>
<td>OTHER</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>RESOLVED</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>IRRESOLVABLE</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>CLOSED</td>
<td>CLOSED</td>
<td>Y</td>
</tr>
<tr>
<td>10</td>
<td>INT CRA REV</td>
<td>HIDDEN</td>
<td>Y</td>
</tr>
<tr>
<td>11</td>
<td>INT DM REV</td>
<td>HIDDEN</td>
<td>Y</td>
</tr>
<tr>
<td>12</td>
<td>INT RESOLVED</td>
<td>CLOSED</td>
<td>N</td>
</tr>
</tbody>
</table>

Note: RDC disallows routing of all but section discrepancies to a HIDDEN status as follows: at run time, if the user selects a univariate or multivariate discrepancy, the Action drop-down list excludes any action that would route the discrepancy to a status that is HIDDEN (that is, has a long value of HIDDEN in the DISCREPANCY STATUS role codelist) for any role.

However, the same restriction does not apply if you use the alternative method for hiding discrepancies from one or more user roles. That is, simply uncheck the ‘active’ checkbox in the DISCREPANCY STATUS role codelist for the ‘blinded’ user role.
To enable users of the relevant role to route discrepancies to a particular status:

1. In the Short Value field, enter the name of the discrepancy status—as it appears in the DISCREPANCY REV STATUS CODE codelist—to which you want users of the role specified in the reference codelist name to be able to route discrepancies.

   **Note:** The codelist must contain one and only one row with ‘CLOSED’ as a short value.

2. In the Long Value field, enter the Actions drop-down item text that should appear for users with the role.

3. Be sure the Active check box is checked.

   **Note:** To remove the item from the Actions drop-down list, uncheck the Active check box.

The Description field is optional.

4. Save your work.

**Rules for the DISCREPANCY ACTIONS role Codelists**

When entering and modifying values in a DISCREPANCY ACTIONS role reference codelist, you must follow these rules:

- Each DISCREPANCY ACTIONS role codelist must contain a subset of the short values (the statuses) defined in the DISCREPANCY REV STATUS CODE codelist. The long value specifies an action that the user can take against a discrepancy. The corresponding short value, which must match a short value in the DISCREPANCY REV STATUS CODE codelist, identifies the status RDC Onsite assigns to the discrepancy when the user selects the action.

- CLOSED should **not** appear as a short value in any DISCREPANCY ACTIONS role codelist.

- You must specify text in the Long Value field. RDC Onsite displays this text in the Action drop-down list.

**DISCREPANCY ACTIONS CRA**

This codelist contains discrepancy actions for the CRA role. The initial short and long values are:

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short value</th>
<th>Long value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DM REVIEW</td>
<td>Null</td>
<td>Send to Data Mgt</td>
</tr>
<tr>
<td>2</td>
<td>RESOLVED</td>
<td>IRRESOLVABLE</td>
<td>Closed - Resolved</td>
</tr>
<tr>
<td>3</td>
<td>IRRESOLVABLE</td>
<td>IRRESOLVABLE</td>
<td>Irresolvable</td>
</tr>
<tr>
<td>4</td>
<td>INT DM REV</td>
<td>Null</td>
<td>Internal Data Mgt review</td>
</tr>
</tbody>
</table>
DISCREPANCY ACTIONS DM
This codelist contains discrepancy actions for the DATA MANAGER role. The initial short and long values are:

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short value</th>
<th>Long value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INT REVIEW</td>
<td>Null</td>
<td>Send to site</td>
</tr>
<tr>
<td>2</td>
<td>TMS REVIEW</td>
<td>Null</td>
<td>Send for classification</td>
</tr>
<tr>
<td>3</td>
<td>RESOLVED</td>
<td>IRRESOLVABLE</td>
<td>Closed - Resolved</td>
</tr>
<tr>
<td>4</td>
<td>IRRESOLVABLE</td>
<td>IRRESOLVABLE</td>
<td>Irresolvable</td>
</tr>
<tr>
<td>5</td>
<td>INT CRA REV</td>
<td>Null</td>
<td>Internal CRA review</td>
</tr>
</tbody>
</table>

DISCREPANCY ACTIONS INV
This codelist contains discrepancy actions for the INVESTIGATOR role. The initial value is:
- DM REVIEW — Send to Data Mgt

DISCREPANCY ACTIONS SITE
This codelist contains discrepancy actions for the SITE role. The initial value is:
- DM REVIEW — Send to Data Mgt

Preventing Update to OTHER Discrepancies
Only RDC Onsite uses the DISCREPANCY NO OTHER UPDATE installation codelist. You can use the DISCREPANCY NO OTHER UPDATE codelist to specify which user roles do not have access to and cannot update discrepancies that appear to them with a status of OTHER; see "Configuring Discrepancy Display by User Role" on page 3-18.

By default, the DISCREPANCY NO OTHER UPDATE codelist has no values. All users can update discrepancies with a status of OTHER. To prevent users from updating OTHER discrepancies, you add one or more user roles to the codelist.

To prevent users with a particular role from updating OTHER discrepancies:
1. Open the DISCREPANCY NO OTHER UPDATE codelist.
2. Enter the role name in the Short Value field. The value you enter must be exactly the same as one of the long values in the USER GROUP ROLES reference codelist. For example, CRA, INV, or SITE.
3. Select the Active check box. An active entry indicates the user role cannot update OTHER discrepancies.
4. Save your work.

The Seq, Long Value, Default, and Description fields are not used by RDC Onsite.

Caution: The system does not check the validity of your entries. You must be careful to specify only valid user roles. If the values do not match exactly, users with the role will still be able to update OTHER discrepancies.
You can grant this privilege to any number of roles.

To allow the update of OTHER discrepancies for a role that you added to the DISCREPANCY NO OTHER UPDATE codelist, you can either:

- Delete the record by using the command on the Data menu.
- Make the value inactive by clearing its Active check box.

**Configuring Study and Site Security for Discrepancy Management**

Once you open either the Maintain Access to Studies form or the Maintain Access to Sites within a Study form, you can use the standard menu commands, toolbar icons, or shortcut keys to:

- Query for one or more records. You can use the % sign as a wildcard search character.
- Add a new record or update existing records.
- Delete one or more records.
- Switch to a different study or site.

For the Study field, Site field, and User field, you can type directly into the field. You can also open a list of valid values and select from the list.

To add or modify the privileges for a user:

1. Open either RDC Administration or Oracle Clinical.
   - In RDC Administration, navigate to Maintain.
   - In Oracle Clinical, navigate to Admin, Users and Roles.
2. Open the correct form:
   - To grant privileges to a user for a particular study, select **Study Security**.
   - To grant privileges to a user for a particular site, select **Site Security**.
3. Query for a particular record or query all records and navigate to the record you want to update. Alternatively, press F6 to insert a blank row and add a new record.
4. Click the **Privilege** column for the user whose privileges you want to update. The dialog box for configuring privileges opens. See Figure 3–5.
5. Select the privileges to assign to the user:
   - To select one privilege, click that privilege.
   - To select several privileges, **Ctrl-click** each privilege. Ctrl-click also toggles the selection on and off.
   - To select a range, **Shift-click** the first and last privilege in the range.
6. Click **OK** to save the privileges for the selected user. Add or modify privileges for other users, as appropriate. Save your changes when finished.

**Note:** A user granted access to a study can see the study in RDC only if the study has at least one site with an investigator assigned and has at least one patient enrolled. Similarly, a user can see a site only if the site has an investigator and patient enrolled.
### Table 3–23 Function Privileges for Discrepancy Management

<table>
<thead>
<tr>
<th>Name</th>
<th>Privilege Granted and Method of Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPD DATA ENTRY</td>
<td>Enables the Update Patient Data menu item under the Special menu, and allows updates while in the Data Entry subsystem.</td>
</tr>
<tr>
<td>BRWS DATA ENTRY</td>
<td>Enables the Browse Patient Data menu item under the Special menu for read-only access to patient data.</td>
</tr>
<tr>
<td>MANUAL</td>
<td>Enables the <strong>Add Manual</strong> button in the Maintain Discrepancy Database window, which allows users to create manual discrepancies and manual header discrepancies.</td>
</tr>
<tr>
<td>DCF PRINT DRAFT</td>
<td>Displays the Draft option in the DCF Print Options window; enables the user to print a draft version of a Data Clarification Form (DCF) report.</td>
</tr>
<tr>
<td>DCF PRINT COPY</td>
<td>Enables the Copy option in the DCF Print Options window; enables the user to print a copy of a DCF.</td>
</tr>
<tr>
<td>DCF PRINT FINISH</td>
<td>Enables the Final option in the DCF Print Options window; enables the user to print a final version of a DCF report.</td>
</tr>
<tr>
<td>DCF REPRINT</td>
<td>Enables the Reprint option in the DCF Print Options window; enables the user to reprint a final version of a DCF report.</td>
</tr>
<tr>
<td>CREATE DCF</td>
<td>Enables the Create DCF option in the group selection menu in the Maintain Discrepancy Database form.</td>
</tr>
</tbody>
</table>

---

### Figure 3–5 Assigning Privileges to a User for a Particular Study

![Data Extract Installation Configuration](image)
Updating a Discrepancy

For a user with the requisite function profile privilege(s) assigned, the system permits the user to update the discrepancies under the following circumstances:

- If there are no privileges assigned to any site and to the study, the user has the UPD_DISCREP privilege.
- If any privilege is assigned to the user at the current site, then the UPDATE or UPD_DISCREP must be assigned to the user at the site level in order for the user to update a discrepancy.
- If the user is not assigned any privileges at the current site-level, but is assigned one or more privileges at the study-level or another site within the study, then the user must be assigned the UPDATE or UPD_DISCREP at the study level.

Navigating to Data Entry

For a user with the requisite function profile privilege(s) assigned, the system permits that user to navigate to data entry and either browse data or update data under the following circumstances. If the user is not able to either browse or update data, the system does not permit navigation to data entry.

In contrast to other study- and site-level security settings, only the site-level BROWSE and/or UPDATE privileges, in conjunction with the function profile privileges, affect a user's ability to browse or update data.

- To update data:
  - UPDATE is assigned at the site-level and UPD DATA ENTRY is assigned as a function profile privilege
  - no privileges are assigned at any site in the study and UPD DATA ENTRY is assigned as a function profile privilege
- If update data is not permitted, to browse data:
  - BROWSE is assigned at the site-level and either UPD DATA ENTRY or the BRWS DATA ENTRY is assigned as a function profile privilege
  - UPDATE is assigned at the site-level and BRWS DATA ENTRY is assigned as a function profile privilege
  - no privileges are assigned at any site in the study and BRWS DATA ENTRY is assigned as a function profile privilege

Setting Up Data Clarification Forms (DCFs)

Oracle Clinical includes a utility for printing and tracking Data Clarification Forms (DCFs) as a way of resolving discrepancies in a clinical trial’s response data. Setting up DCFs for your organization requires that you define DCF status codes and lay out the DCF report.

This section includes the following topics:

- Defining DCF Statuses and their Behavior on page 3-29
- Laying Out the DCF on page 3-30
- Creating New Standard Text on page 3-34
Defining DCF Statuses and their Behavior

There are several Installation Reference Codelists that affect DCF Statuses. You can modify them if you want to for your organization’s needs.

**DCF STATUS CODES**  The DCF Status reflects the stage in the review process of the DCF as a whole. Oracle Clinical comes with many review statuses defined in this reference codelist. You can make them inactive if you do not want to use them or create new ones, with the following limitations:

- You cannot make these statuses inactive: Sent, Received, and Closed.
- Do not make status Created inactive unless you also make it optional in the DCF OPTIONAL STATUS CODES reference codelist.
- Do not make statuses Incomplete, Part Received, or Received inactive if you want the system to automatically update the DCF Status when users update the status of individual DCF Pages.
- If you add a status, you must set its DISPLAY_SN, which determines the order in which the statuses can be set. If you adjust the DISPLAY_SN, you may need to adjust the DCF LOCK CONDITIONS and the DCF OPTIONAL STATUS CODES installation reference codelists, both of which refer to DCF Statuses by their number in the DCF STATUS CODES installation reference codelist.

**DCF OPTIONAL STATUS CODES**  This installation reference codelist refers to the DCF Statuses listed in the DCF STATUS CODES reference codelist. The number in the Short Value column refers to the display number of the status in the DCF STATUS CODES codelist. (Note that the description is incorrect for numbers 7 and 8, which should be Incomplete and Part Received.)

All status codes referenced and active in this codelist are optional. All that are inactive or not entered here are mandatory, meaning that a DCF must be assigned to that status before it can be assigned a subsequent status, as defined in the Seq column. As shipped, only CREATED, SENT and CLOSED are mandatory (no rows exist with short values 2, 6, or 12).

To make a status mandatory if it is included in this reference codelist, uncheck its Active box.

**DCF LOCK CONDITIONS**  This installation reference codelist determines what actions can be taken on discrepancies and DCFs where the DCF has a particular status. The Long Value refers to the number of the status in the DCF STATUS CODES codelist. As shipped, the codelist sets the following behavior, in order:

- Discrepancies belonging to a DCF whose status is Final (#4) or higher cannot be deleted.
- Discrepancies belonging to a DCF whose status is Ready (#5) or higher cannot be modified.
- DCFs whose status is Ready (#5) or higher cannot be deleted (unless the status exceeds or equals a status with a DISPLAY_SN of 1000; see last point).
- DCFs whose status is Ready (#5) or higher cannot be modified.
- DCFs whose status is #1000 or higher cannot be closed. We recommend that you do not change DCF_CLOSE.

To stop enforcing any of these rules, set the Long Value to a high value. To change the DCF Status that prevents any of the actions described in the Short Value column,
change the number in the Long Value column to the number in the DCF STATUS CODES codelist of the status you prefer.

Laying Out the DCF

Laying out the DCF’s contents and arranging its fields includes the following tasks:

- Replacing the DCF Place-holder Graphic on page 3-30
- Modifying Codelist DCF REPORT LABELS on page 3-30
- Modifying the DCF Views on page 3-31
- Defining DCF Headers and Footers on page 3-32

See the Default DCF Layout Diagram on page 3-32 for a geographic representation of the DCF’s default values and their placements.

OPA views dcf_rpt_master and dcf_rpt_detail control much of the content and the appearance of DCFs. Local codelist DCF REPORT LABELS controls the report’s label text. DCFs also include several unchangeable parameters, such as the Revision # and other values that print along the bottom of each DCF page.

Replacing the DCF Place-holder Graphic

Oracle Clinical includes a placeholder graphic file located, typically, in your installation’s ..\OPA\release_number\oc directory, and named rxcdcf.bmp. You can replace rxcdcf.bmp with your own graphic file, but you must name it rxcdcf.bmp. We recommend that your graphic does not exceed 200 pixels in height.

Modifying Codelist DCF REPORT LABELS

The labels for the value fields of the DCF views correspond to their position on the report. The following table describes each of the columns on the report. For field labels, it identifies the codelist member in the DCF REPORT LABELS local codelist and their default table and column values.

<table>
<thead>
<tr>
<th>Short Value</th>
<th>Long Value</th>
<th>Default Data Value from DCF_RPT_MASTER and DCF_RPT_DETAIL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1_left</td>
<td>To:</td>
<td>ocl_sites.name</td>
<td>Site name’s short value</td>
</tr>
<tr>
<td>line3_left</td>
<td>Date:</td>
<td>sysdate</td>
<td>Date of this DCF’s creation: mm/dd/yyyy</td>
</tr>
<tr>
<td>line3_right</td>
<td>Reviewer</td>
<td>oracle_accounts.firstname (and) .lastname</td>
<td>Reviewer’s initials</td>
</tr>
<tr>
<td>line2_left</td>
<td>Investigator:</td>
<td>ocl_investigators.first_name (and) .lastname</td>
<td>Investigator’s first and last name.</td>
</tr>
<tr>
<td>line1_right</td>
<td>Patient #:</td>
<td>patient_positions.patient</td>
<td>Patient position</td>
</tr>
<tr>
<td>line2_right</td>
<td>Patient Initials:</td>
<td>patient_positions.reported_first_name (and) .reported_last_name</td>
<td>Patient’s initials</td>
</tr>
<tr>
<td>q_field1</td>
<td>Form Name/ Visit Name:</td>
<td>dcms.description</td>
<td>Mapped to view value qline1_1</td>
</tr>
<tr>
<td>q_field2</td>
<td>Page #:</td>
<td>discrepancy_entries.crf_page_number</td>
<td>CRF page number</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mapped to view value qline1_2</td>
</tr>
</tbody>
</table>
Table 3–24 (Cont.) DCF Labels and Field Mappings

<table>
<thead>
<tr>
<th>Default Field Label from DCF_REPORT_LABELS</th>
<th>Default Data Value from DCF_RPT_MASTER and DCF_RPT_DETAIL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Value</td>
<td>Long Value</td>
<td>Table.Column</td>
</tr>
<tr>
<td>q_field3</td>
<td>Date:</td>
<td>discrepancy_entries.creation_ts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q_field4</td>
<td>Questions/Comments:</td>
<td>dcf_discrepancies.comment_text</td>
</tr>
<tr>
<td>q_field5</td>
<td>Resolution:</td>
<td>dcf_discrepancies.resolution_text</td>
</tr>
<tr>
<td>mstr_sort_order1</td>
<td>line1_right</td>
<td>patient_positions.patient</td>
</tr>
<tr>
<td>dtl_sort_order1</td>
<td>q_line1_1</td>
<td>dcm.description</td>
</tr>
<tr>
<td>rpt_orientation1</td>
<td>Landscape</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Modifying the DCF Views

Table 3–25 and Table 3–26 describe all of the parameters in the two DCF report views. The view scripts for the DCF are in file rxcviews.sql. You can map other discrepancy values to the DCF parameters, or comment out the parameters to remove them from the DCF output. Note that if you remap parameters, you may have to change their labels (See "Modifying Codelist DCF REPORT LABELS" on page 3-30).

Caution: Do not change parameter values that have a _DNC suffix.

Table 3–25 View DCF_RPT_MASTER Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Data Value (Table.Column)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcf_id_dnc</td>
<td>data_clarification_forms.dcf_id</td>
<td>A system-generated number.</td>
</tr>
<tr>
<td>title1</td>
<td>clinical_studies.short_title</td>
<td>Prints the study’s title.</td>
</tr>
<tr>
<td>title2</td>
<td>'Data Clarification Form’</td>
<td>Prints the text string Discrepancy Clarification Form.</td>
</tr>
<tr>
<td>line1_left</td>
<td>ocl_sites.name</td>
<td>Site name's short value</td>
</tr>
<tr>
<td>line3_left</td>
<td>sysdate</td>
<td>Date of this DCF's creation: MM/DD/YYYY</td>
</tr>
<tr>
<td>line3_right</td>
<td>oracle_accounts.firstname and .lastname</td>
<td>DCF creator's account name</td>
</tr>
<tr>
<td>line2_left</td>
<td>ocl_investigators.first_name (and) .lastname</td>
<td>Investigator's first and last name.</td>
</tr>
<tr>
<td>line1_right</td>
<td>patient_positions.patient</td>
<td>Patient position</td>
</tr>
<tr>
<td>line2_right</td>
<td>patient_positions.reported_first_name (and) .reported_last_name</td>
<td>Patient’s initials</td>
</tr>
<tr>
<td>header_text</td>
<td>header_text</td>
<td>Prints the discrepancy's header text. Define header text in the Standard Text Maintenance form.</td>
</tr>
</tbody>
</table>
Table 3–26 describes the values of the DCF_RPT_DETAIL view, and its default data values. In DCF report printouts, the system draws a rectangular border around the output of this view, and may include more than one discrepancy’s details.

### Table 3–26 View DCF_RPT_DETAIL Parameters

<table>
<thead>
<tr>
<th>View Parameter</th>
<th>Default Data Value (Table.Column)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcf_id_dnc</td>
<td>data_clarification_forms.dcf_id</td>
<td></td>
</tr>
<tr>
<td>q_line1_1</td>
<td>dcms.description</td>
<td>Value for q_field1 label</td>
</tr>
<tr>
<td>q_line1_2</td>
<td>discrepancy_entries.crf_page_number</td>
<td>Value for q_field2 label</td>
</tr>
<tr>
<td>q_line1_3</td>
<td>received_dcms.dcm_date</td>
<td>Value for q_field3 label</td>
</tr>
<tr>
<td>q_line1_4</td>
<td>dcf_discrepancies.comment_text</td>
<td>Value for q_field4 label</td>
</tr>
<tr>
<td>q_line1_5</td>
<td>dcf_discrepancies.resolution_text</td>
<td>Value for q_field5 label</td>
</tr>
<tr>
<td>q_line2_1</td>
<td>clinical_planned_events.description</td>
<td></td>
</tr>
<tr>
<td>status_dnc</td>
<td>dcf_discrepancies.status</td>
<td></td>
</tr>
<tr>
<td>disc_type</td>
<td>discrepancy_entries.de_sub_type_code</td>
<td></td>
</tr>
<tr>
<td>disc_rev</td>
<td>discrepancy_entries.discrepancy_rev_status</td>
<td></td>
</tr>
</tbody>
</table>

### Defining DCF Headers and Footers

You can define standard text headers and footers in the Standard Text Maintenance form (see "Creating New Standard Text" on page 3-34). You can define any number of standard headers and footers and designate one header and one footer as the default. When you create a DCF, the system assigns it the default header and footer. You can use the list of values in the Header and Footer fields to choose a different standard header or footer.

You can also modify the standard text of the header or footer by clicking the Text button in the main DCF screen and editing the displayed text.

### Default DCF Layout Diagram

The following diagram shows the relative position of the labels (in bold font), and the default data values of a default DCF printout. The section between the rxcdf.bmp graphic and the ruled rectangle contains most of the data defined in the DCF_RPT_MASTER view. The DCF_RPT_DETAIL view populates the contents in the rectangle. A
DCF can accommodate the descriptions of more than one discrepancy on each page. The legend for the DCF layout diagram precedes the diagram itself.

```
status_dnc
dcf_discrepancies.status

title1 clinical_studies.short_title
title2

To: ocl_sites.name
line1_left

Patient: patient_
line1_right

Investigator: ocl_
line2_left investigators.fi
rst_name (and)
.lastname

Patient Initials: patient_
line2_right
(and) .reported_ last_name

Date: sysdate
line3_left

Reviewer: oracle_
line3_right
(and) .lastname

Form Name/ Visit Name q_field1 Page # q_field2 Questions/Comments q_field4 Resolution q_field5

Date q_field3

DCF_id_dnc q_line1_2 q_line1_4 q_line1_5
data_discrepancy_ dcf_dcf_ clarification_entries.crf_ discrepancies.c
_forms.dcf_id page_number discrepancies.resolution_ comment_text
c_line1_3
dcms.description received dcms.dcm_date

Closed: status_dnc
dcf_ discrepancies.status

Disc ID: discrepancy_ entry_id_dnc
discrepancy_ discrepancies.discrepa
tentries

d_line1_3
dcms.description received dcms.dcm_date

Type: disc_type
discrepancy_
entries.de_sub_
type_code
```
Creating Reusable Standard Text for Discrepancies and DCFs

This section includes the topics:

- **Creating New Standard Text** on page 3-34
- **Inserting Replacement Parameters** on page 3-35
- **Customizing Default Standard Text Entries** on page 3-36

The standard text utility allows you to create uniform, reusable text for discrepancy comments and error messages and for DCF headers and footers. Using standard text saves time and promotes consistency. There are four types of standard text, used as follows by the system:

- **Comment.** Displayed as an error message during Data Entry and Data Entry Update for univariate discrepant responses; also displayed in the Comment field in the Maintain Discrepancy Database window for univariate discrepancies. Oracle Clinical ships with a set of default univariate error messages, one for each type of univariate discrepancy. You can edit the existing default and/or create alternatives to be available for use in Data Entry, Data Entry Update, and discrepancy management; see "Customizing Default Standard Text Entries" on page 3-36.

  Comment-type standard text is also available by pressing F9 in the Comment field of the Maintain Discrepancy Database window. Comment text is displayed in the Create DCF window List of Values to help the user choose which discrepancy to include in a DCF when creating a DCF by discrepancy number.

- **Internal Comment.** In the Maintain Discrepancy Database window, users can enter an original comment in free form text or, by pressing F9 in the Inter Com field, choose from the internal comments you define here.

- **Resolution.** In the Maintain Discrepancy Database window, users can enter an original comment in free form text when they resolve a discrepancy or, by pressing F9 in the Res Com field, choose from the resolution comments you define here.

- **Header and Footer Text on DCFs.** The header and footer text you define appears on each page of a DCF. See "Defining DCF Headers and Footers" on page 3-32.

  For each type of standard text definition, you can include replacement parameters, or variables. When the system displays the standard text, it substitutes the actual value of the parameter; see "Inserting Replacement Parameters" on page 3-35.

Creating New Standard Text

To create new standard text entries:

1. From the Admin menu, select Discrepancy Mgmt, then select Standard Text Maintenance.
The Standard Text Maintenance form opens, with columns for Text Type, Sub Type, Standard Text Name, a Default box, and a line for the standard text string.

2. Choose a text type. The text type identifies where the system uses the definition: In DCF headers or footers or discrepancy comments and Data Entry. The choices are: Comment, Footer, Header, Internal Comment, Resolution. For further information, see the introduction to this section, "Creating Reusable Standard Text for Discrepancies and DCFs" on page 3-34.

3. Choose a subtype only if you are creating an alternative univariate error text (you must have chosen a Text Type of Comment as well). See "Customizing Default Standard Text Entries" on page 3-36. Choose the subtype corresponding to the univariate error type for which you are creating a standard text.

For discrepancy comments, internal comments, and resolution comments, and DCF headers and footers, do not enter a subtype. If you do, the text will not be available for use even though it is successfully saved.

4. Name the entry in the **Standard Text Name** field. Each standard text definition must have a unique name. Users see the name and the text definition in the list of values for Comments, Internal Comments, and Resolution Comments of the Maintain Discrepancy Database.

5. Toggle the **Default** box. For the COMMENT type, the box controls which standard text value the system displays during Data Entry and DE Update. For HEADER and FOOTER types, a check identifies the default when a new DCF is created. Only one entry can be checked as default for each text type/subtype combination.

6. Write the text in the **Standard Text** field.

You can embed variables in your standard text that the system replaces with values from individual records. Press F9 to see the list of values, select and insert a variable. For further information, see "Inserting Replacement Parameters" on page 3-35 and, for examples of how the system replaces variables with actual values, Table 3–29, "Examples of Alternative Standard Text for Univariate Errors" on page 3-37.

**Inserting Replacement Parameters**

Standard text definitions can contain standard replacement parameters, or variables. You can insert one of the replacement parameters shown in Table 3–27 into your standard text. When the system displays the standard text, it displays the actual value of the parameter. The replacement parameters are in UPPERCASE and are delimited by backslashes (\); for example, \SAS_LABEL\.

To insert a parameter in a standard text string, put your cursor in the location in the text where you want to insert the variable, press F9 or click the ellipsis (...) to display the list of values, and select it from the list of values. Table 3–27, "Replacement Parameters" describes each available variable and its source table or view and column.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>assoc_id</td>
<td>Associated Id for the discrepancy</td>
<td>discrepancy_management.associated_id</td>
</tr>
<tr>
<td>crf_page_no</td>
<td>CRF Page number for the discrepancy</td>
<td>discrepancy_management.crf_page_number</td>
</tr>
<tr>
<td>data_type</td>
<td>Data Type of the question</td>
<td>dcm_questions.question_data_type_code</td>
</tr>
<tr>
<td>date_time_format</td>
<td>Precision for Date and Time</td>
<td>dcm_questions.date_time_type_code</td>
</tr>
<tr>
<td>dcm_name</td>
<td>DCM Name for the discrepancy</td>
<td>discrepancy_management.name</td>
</tr>
</tbody>
</table>
Customizing Default Standard Text Entries

You can modify the default standard text in the Standard Text Maintenance form. From the Admin menu, select Discrepancy Mgmt Admin, then select Standard Text Maintenance. The window is populated with the default values for each type of univariate error message shipped with Oracle Clinical. Table 3–28 lists the default standard text definitions for Oracle Clinical Release 4.6.

Table 3–28 Default Univariate Discrepancy Messages

<table>
<thead>
<tr>
<th>Discrepancy Type</th>
<th>Default Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>data type</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> is not a valid <code>\DATA_TYPE\</code></td>
</tr>
<tr>
<td>dvg</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> not found in <code>\DISCRETE_VALUES\</code></td>
</tr>
<tr>
<td>dvg subset</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> not found in <code>\DISCRETE_VALUES\</code></td>
</tr>
<tr>
<td>length</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> exceeds expected length of <code>\LENGTH\</code></td>
</tr>
<tr>
<td>lowerbound</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> below the minimum value of <code>\LOWER_BOUND\</code></td>
</tr>
<tr>
<td>mandatory</td>
<td>Value for <code>\SAS_LABEL\</code> has not been supplied</td>
</tr>
<tr>
<td>missing_pt</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> is awaiting classification</td>
</tr>
<tr>
<td>missing_sct</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> is awaiting classification</td>
</tr>
<tr>
<td>partial date</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> is an incomplete date or time</td>
</tr>
<tr>
<td>precision</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> exceeds <code>\DECIMAL_PLACES\</code> decimal places</td>
</tr>
<tr>
<td>thesaurus</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> is not in the lookup thesaurus</td>
</tr>
<tr>
<td>upperbound</td>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS_LABEL\</code> above the maximum value of <code>\UPPER_BOUND\</code></td>
</tr>
</tbody>
</table>
Example 3–3  Standard Text

The following table illustrates how you can use standard text definitions with different replacement parameters to create appropriate alternative error messages for DVG univariate discrepancies.

The system generates a different error message from same standard text definition by populating the replacement parameters with values from the question definition and entered data. The first column contains standard text definitions using replacement parameters. The second column shows the error message that would appear during data entry if the operator entered "X" as a response to the question PATIENT_SEX, which has a DVG containing the values M(ale) and F(emale). The third column shows the error message that would result from the same standard text if an operator entered "X" for the question SMOKING, which has a DVG containing the values Y(es) and N(o).

Table 3–29  Examples of Alternative Standard Text for Univariate Errors

<table>
<thead>
<tr>
<th>Standard Text</th>
<th>Sex Example</th>
<th>Smoking Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value <code>\VALUE_TEXT\</code> for question <code>\SAS.LABEL\</code> is not in expected list <code>\DISCRETE_VALUES\</code>.</td>
<td>Value 'X' for question SEX is not in expected list 'M,F.'</td>
<td>Value 'X' for question <code>\SMOKING\</code> is not in expected list 'Y,N.'</td>
</tr>
<tr>
<td>Value of <code>\VALUE_TEXT\</code> does not exist in Discrete Value Group <code>\DVG_NAME\</code>.</td>
<td>Value of X does not exist in Discrete Value Group PATIENT_SEX.</td>
<td>Value of X does not exist in Discrete Value Group Yes/No.</td>
</tr>
<tr>
<td>Value of <code>\VALUE_TEXT\</code> for <code>\SAS.LABEL\</code> is invalid.</td>
<td>Value of X for SEX is invalid.</td>
<td>Value of X for SMOKING is invalid.</td>
</tr>
</tbody>
</table>
Configuring the Mass Changes Utility

Setting up the Mass Change Utility (MCU), which is described in Oracle Clinical Conducting a Study, requires performing the following general steps, each described in the following sections:

- Creating and Assigning Mass Changes Roles on page 4-1
- Customizing Mass Changes Local Codelists on page 4-1
- Customizing the Field Display on the Candidate Data Set Form on page 4-2

Creating and Assigning Mass Changes Roles

Oracle Clinical supplies two default database roles for the Mass Changes Utility: RXC_MC and RXC_MC_TEST. RXC_MC has access to all production mass change options and RXC_MC_TEST has access to the test mass change options only. Your organization can also choose to create your own roles.

Assign these roles to the users that work with the Mass Changes Utility.

Customizing Mass Changes Local Codelists

The Mass Changes Utility uses the local reference codelists described in the following table. You can modify them to suit your organization’s needs. The affected tablename.columnname column indicates the database columns that are affected by or use each reference codelist. From the Admin menu, select Reference Codelists, then select Local Codelists and query the codelists named in the following table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Affected Tablename.Columnname</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC CDS SORT ORDER</td>
<td>Default sort order for viewing and applying the candidate data set (CDS); the columns you can order.</td>
<td>mass_changes.order_by_cols</td>
</tr>
<tr>
<td>MC COLUMNS²</td>
<td>The list of columns that can be used in LHS or RHS of criteria, such as RDCLSITE</td>
<td>mass_change_criteria.col mass_change_criteria.value</td>
</tr>
<tr>
<td>DISC COLS2</td>
<td>Columns available for the LHS and RHS of the CDS Criteria form, regardless of the discrepancy type specified, such as DE.CREATION_TS</td>
<td>mass_change_criteria.col mass_change_criteria.value</td>
</tr>
<tr>
<td>UNI DISC COLS2</td>
<td>Columns available for the LHS and RHS of the CDS Criteria form, where the discrepancy type specified is UNIVARIATE, such as DE.DISCREPANCY_TYPE_CODE</td>
<td>mass_change_criteria.col mass_change_criteria.value</td>
</tr>
<tr>
<td>IND DISC COLS2</td>
<td>Columns available for the LHS and RHS of the CDS Criteria form, where the discrepancy type specified is INDICATOR</td>
<td>mass_change_criteria.col mass_change_criteria.value</td>
</tr>
</tbody>
</table>
Customizing the Field Display on the Candidate Data Set Form

These local codelists control the display of fields on the Candidate Data Set form for the four Mass Changes Utility change types:

- FLD RXCMCMCD RDCI KEY
- FLD RXCMCMCD RDCM KEY
- FLD RXCMCMCD RDCI DELETE
- FLD RXCMCMCD RESPONSE

From the Admin menu, select Reference Codelists, then select Local Codelists, and query FLD% in the Name field. The following table describes how the codelist values control all four codelists:

<table>
<thead>
<tr>
<th>Table 4–1 (Cont.) Mass Changes Utility Reference Codelists</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>MULTI DISC COLS2</td>
</tr>
<tr>
<td>MANHD DISC COLS2</td>
</tr>
<tr>
<td>FLD RXCMCMCD RDCI KEY</td>
</tr>
<tr>
<td>FLD RXCMCMCD RDCM KEY</td>
</tr>
<tr>
<td>FLD RXCMCMCD RDCI DELETE</td>
</tr>
<tr>
<td>FLD RXCMCMCD RESPONSE</td>
</tr>
<tr>
<td>OCL MC PREFS</td>
</tr>
</tbody>
</table>

1 Only the display sequence and whether the field should be sorted in ascending or descending order should be changed. New fields should not be added and existing ones should not be removed.

2 The values in this reference codelist determine the contents of the list of values for mass change criteria. Additional values may be added to the reference codelist but they must be from one of the existing tables already defined for values, and they must use the alias provided for the table. No validation occurs against the values in this codelist. If a value is not in the list of values it can still be added as a criterion for specifying the candidate data set.

3 OCL MC PREFS currently has one value, MAX_CDS_RECORDS, which controls the default maximum number of records to include in a Candidate Data Set.

Customizing the Field Display on the Candidate Data Set Form

These local codelists control the display of fields on the Candidate Data Set form for the four Mass Changes Utility change types:

- FLD RXCMCMCD RDCI KEY
- FLD RXCMCMCD RDCM KEY
- FLD RXCMCMCD RDCI DELETE
- FLD RXCMCMCD RESPONSE

From the Admin menu, select Reference Codelists, then select Local Codelists, and query FLD% in the Name field. The following table describes how the codelist values control all four codelists:

<table>
<thead>
<tr>
<th>Table 4–2 CDS Display Codelist Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Codelist Field Name</strong></td>
</tr>
<tr>
<td>Seq</td>
</tr>
<tr>
<td>Short Value</td>
</tr>
<tr>
<td>Long Value</td>
</tr>
</tbody>
</table>
Oracle Clinical has default values for these codelists that reflect the most likely scenario for displaying the Candidate Data Set. You can modify the default values. The following table lists all possible column values. You can add, remove, modify, or rearrange the display fields. Footnotes and a legend with descriptions of all of the table’s symbols follows the table.

### Table 4–3 Candidate Data Set Form Codelist Values

<table>
<thead>
<tr>
<th>Column Value</th>
<th>Short Name</th>
<th>Display Settings by Mass Changes Utility Change Type</th>
<th>Recommended</th>
<th>Set by Default</th>
<th>Can be Set</th>
<th>List of Values</th>
<th>Updateable</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdr_status_code</td>
<td>cdr_status_code</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>validate_comment</td>
<td>val_comment</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>change_reason_code²</td>
<td>c_reason_code</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>audit_comment</td>
<td>audit_comment</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>how_updated</td>
<td>how_updated</td>
<td>I, M, R</td>
<td>I, M, R</td>
<td>I, M, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>received_dci_id</td>
<td>received_dci_id</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>document_number</td>
<td>doc_number</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>document_number_new</td>
<td>doc_number_new</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>investigator</td>
<td>investig</td>
<td>I</td>
<td>I</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>investigator_new</td>
<td>investig_new</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>site</td>
<td>site</td>
<td>I</td>
<td>I</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>site_new</td>
<td>site_new</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>patient</td>
<td>patient</td>
<td>I</td>
<td>I</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>patient_new</td>
<td>patient_new</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>dci_short_name</td>
<td>dci_sh_name</td>
<td>I</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 4–3 (Cont.) Candidate Data Set Form Codelist Values

<table>
<thead>
<tr>
<th>Column Value</th>
<th>Short Name</th>
<th>Display Settings by Mass Changes Utility Change Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Recommended</td>
</tr>
<tr>
<td>dci_short_name_new</td>
<td>dci_sh_name_new</td>
<td>I</td>
</tr>
<tr>
<td>clin_plan_eve_name</td>
<td>cpe_name</td>
<td>I</td>
</tr>
<tr>
<td>clin_plan_eve_name_new</td>
<td>cpe_name_new</td>
<td>I</td>
</tr>
<tr>
<td>visit_number</td>
<td>vis_number</td>
<td>I</td>
</tr>
<tr>
<td>subevent_number</td>
<td>sub_number</td>
<td>I</td>
</tr>
<tr>
<td>subevent_number_new</td>
<td>sub_number_new</td>
<td>I</td>
</tr>
<tr>
<td>dci_date</td>
<td>dci_dt</td>
<td>I</td>
</tr>
<tr>
<td>dci_date_new</td>
<td>dci_dt_new</td>
<td>I</td>
</tr>
<tr>
<td>dci_time</td>
<td>dci_tm</td>
<td>I</td>
</tr>
<tr>
<td>dci_time_new</td>
<td>dci_tm_new</td>
<td>I</td>
</tr>
<tr>
<td>comment_text</td>
<td>cm_txt</td>
<td>I, M</td>
</tr>
<tr>
<td>comment_text_new</td>
<td>cm_txt_new</td>
<td>I, M</td>
</tr>
<tr>
<td>blank_flag</td>
<td>blank_flag</td>
<td>I, M</td>
</tr>
<tr>
<td>blank_flag_new</td>
<td>blank_flag_new</td>
<td>I, M</td>
</tr>
<tr>
<td>received_dcm_id</td>
<td>received_dcm_id</td>
<td>M, R</td>
</tr>
<tr>
<td>rdc_i Clin Plan_eve_name</td>
<td>rdc_i cpe_name</td>
<td>M, R</td>
</tr>
<tr>
<td>rdc_i Visit_number</td>
<td>rdc_i vis_number</td>
<td>M, R</td>
</tr>
<tr>
<td>rdc_i Subevent_number</td>
<td>rdc_i sub_number</td>
<td>M, R</td>
</tr>
<tr>
<td>rdc_i Comment_text</td>
<td>rdc_i cm_txt</td>
<td>M, R</td>
</tr>
<tr>
<td>rdc_i Blank Flag</td>
<td>rdc_i blank_flag</td>
<td>M, R</td>
</tr>
<tr>
<td>DCM Name</td>
<td>dcm_name</td>
<td>M</td>
</tr>
<tr>
<td>DCM Subset Name</td>
<td>dcm_subset_name</td>
<td>M</td>
</tr>
<tr>
<td>DCM Layout Sn</td>
<td>dcm_layout_sn</td>
<td>M</td>
</tr>
<tr>
<td>DCM Date</td>
<td>dcm_dt</td>
<td>M</td>
</tr>
<tr>
<td>DCM Date New</td>
<td>dcm_dt_new</td>
<td>M</td>
</tr>
<tr>
<td>DCM Time</td>
<td>dcm_tm</td>
<td>M</td>
</tr>
<tr>
<td>DCM Time New</td>
<td>dcm_tm_new</td>
<td>M</td>
</tr>
<tr>
<td>Qualifying Value</td>
<td>qual_value</td>
<td>M</td>
</tr>
<tr>
<td>Qualifying Value New</td>
<td>qual_value_new</td>
<td>M</td>
</tr>
<tr>
<td>Data Comment Text</td>
<td>dta_cm_txt</td>
<td>M</td>
</tr>
<tr>
<td>Data Comment Text New</td>
<td>dta_cm_txt_new</td>
<td>M</td>
</tr>
<tr>
<td>Repeat Sn</td>
<td>repeat_sn</td>
<td>R</td>
</tr>
<tr>
<td>Response Id1</td>
<td>response_id1</td>
<td>R</td>
</tr>
<tr>
<td>Validation Status1</td>
<td>valid_status1</td>
<td>R</td>
</tr>
</tbody>
</table>
### Table 4–3 (Cont.) Candidate Data Set Form Codelist Values

<table>
<thead>
<tr>
<th>Column Value</th>
<th>Short Name</th>
<th>Display Settings by Mass Changes Utility Change Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>exception_value_text1</td>
<td>e_val_txt1</td>
<td>Recom-mended</td>
</tr>
<tr>
<td>value_text1</td>
<td>val_txt1</td>
<td>R</td>
</tr>
<tr>
<td>full_value_text1</td>
<td>f_val_txt1</td>
<td>R</td>
</tr>
<tr>
<td>full_value_text1_new²</td>
<td>f_val_txt1_new</td>
<td>R</td>
</tr>
<tr>
<td>data_comment_text1</td>
<td>dta_cm_txt1</td>
<td>R</td>
</tr>
<tr>
<td>data_comment_text1_new</td>
<td>dta_cm_txt1_new</td>
<td>R</td>
</tr>
<tr>
<td>response_id2</td>
<td>response_id2</td>
<td>R</td>
</tr>
<tr>
<td>validation_status2</td>
<td>valid_status2</td>
<td>R</td>
</tr>
<tr>
<td>exception_value_text2</td>
<td>e_val_txt2</td>
<td>R</td>
</tr>
<tr>
<td>value_text2</td>
<td>val_txt2</td>
<td>R</td>
</tr>
<tr>
<td>full_value_text2</td>
<td>f_val_txt2</td>
<td>R‡</td>
</tr>
<tr>
<td>full_value_text2_new²</td>
<td>f_val_txt2_new</td>
<td>R‡</td>
</tr>
<tr>
<td>data_comment_text2</td>
<td>dta_cm_txt2</td>
<td>R‡</td>
</tr>
<tr>
<td>data_comment_text2_new</td>
<td>dta_cm_txt2_new</td>
<td>R‡</td>
</tr>
<tr>
<td>response_id3</td>
<td>response_id3</td>
<td>R</td>
</tr>
<tr>
<td>validation_status3</td>
<td>valid_status3</td>
<td>R</td>
</tr>
<tr>
<td>exception_value_text3</td>
<td>e_val_txt3</td>
<td>R</td>
</tr>
<tr>
<td>value_text3</td>
<td>val_txt3</td>
<td>R</td>
</tr>
<tr>
<td>full_value_text3</td>
<td>f_val_txt3</td>
<td>R‡</td>
</tr>
<tr>
<td>full_value_text3_new²</td>
<td>f_val_txt3_new</td>
<td>R‡</td>
</tr>
<tr>
<td>data_comment_text3</td>
<td>dta_cm_txt3</td>
<td>R‡</td>
</tr>
<tr>
<td>data_comment_text3_new</td>
<td>dta_cm_txt3_new</td>
<td>R‡</td>
</tr>
<tr>
<td>response_id4</td>
<td>response_id4</td>
<td>R</td>
</tr>
<tr>
<td>validation_status4</td>
<td>valid_status4</td>
<td>R</td>
</tr>
<tr>
<td>exception_value_text4</td>
<td>e_val_txt4</td>
<td>R</td>
</tr>
<tr>
<td>value_text4</td>
<td>val_txt4</td>
<td>R</td>
</tr>
<tr>
<td>full_value_text4</td>
<td>f_val_txt4</td>
<td>R‡</td>
</tr>
<tr>
<td>full_value_text4_new²</td>
<td>f_val_txt4_new</td>
<td>R‡</td>
</tr>
<tr>
<td>data_comment_text4</td>
<td>dta_cm_txt4</td>
<td>R‡</td>
</tr>
<tr>
<td>data_comment_text4_new</td>
<td>dta_cm_txt4_new</td>
<td>R‡</td>
</tr>
</tbody>
</table>

¹ Validation issue: Value must be valid in the reference code list.
² Validation issue: Must pass field validation in data entry.
### Table 4–4  Legend for Table 4–3

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A blank cell indicates that the value does not qualify for any of the change types.</td>
</tr>
<tr>
<td>L</td>
<td>RDCI KEY change type</td>
</tr>
<tr>
<td>D</td>
<td>RDCI DELETE change type</td>
</tr>
<tr>
<td>M</td>
<td>RDCM KEY change type</td>
</tr>
<tr>
<td>R</td>
<td>RESPONSE change type</td>
</tr>
<tr>
<td>All</td>
<td>Applies to all change type codes: RDCI KEY, RDCI DELETE, RDCM KEY, RESPONSE</td>
</tr>
<tr>
<td>-</td>
<td>The system does not display this field. Instead a display field is used as with the Data Entry Log-in form. This field displays the data retrieved into the base table field in the appropriate format and translates the information entered into the new fields to the format needed by the base table fields. This field contains an list of values, or is update-able, pertains to the display field, with the update of the base table fields by the system. It is necessary for the reference codelist to have the display fields rather than the base table field. For these fields, the labels, Column value refers to the base table field and short name refers to the display field.</td>
</tr>
<tr>
<td>--</td>
<td>The clin_plan_eve_name_new field list of values contains the visit number as well, for reference purposes only.</td>
</tr>
<tr>
<td>†</td>
<td>Responses and associated old and new values are required and present by default, if they exist.</td>
</tr>
<tr>
<td>‡</td>
<td>An list of values is present for exception_value_text if an alpha data code discrete value group exists for the DCM question, and values from this group are part of the list of values.</td>
</tr>
<tr>
<td>˚</td>
<td>An list of values is present for value_text if a discrete value group exists for the DCM question, and values from this group will appear in the list of values</td>
</tr>
<tr>
<td>ˌ</td>
<td>There is an list of values for the full_value_text if any discrete value group exists for the DCM question, and values from these groups appear in the list of values.</td>
</tr>
</tbody>
</table>
This section includes the following topics:

- Customizing Data Entry Behavior on page 5-1
- Customizing the Oracle Clinical Log-in Window Layout on page 5-7
- Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report on page 5-10
- Customizing Flex Fields for DCI Forms on page 5-20
- Customizing CRF Column Headers in the RDC Classic Surround on page 5-22
- Customizing Online Help on page 5-24

**Customizing Data Entry Behavior**

A number of features, behaviors, and even the appearance of the Log-In and Data Entry screens can be configured depending on configuration settings, user preferences, and special layout editing tools. This section provides an overview of these features; more detail on each feature is available in later sections.

The features discussed in this section are:

- Define Data Entry Configuration Settings on page 5-1
- Configuring Additional Data Entry User Preferences on page 5-5
- Configuring Privileged Update on page 5-7

**Define Data Entry Configuration Settings**

Data entry configuration settings, such as whether univariate validation failures alert the First-Pass and Second-Pass data entry operators, can be set at the local database level, the study level, or the user level.

Data entry configuration settings can be set at the:

- **Database Level** through the Maintain Installation Configuration window; see Configure Database-Level Data Entry Settings on page 5-2
- **Study Level** through the Maintain Study Configuration window (via Maintain Clinical Study States window); see Configure Study-Level Data Entry Settings on page 5-4
User Level through the Maintain Oracle Accounts window; Configure User-Level Data Entry Settings on page 5-4

As the configuration level becomes more specific, from database to study to user, its settings usually take precedence over the more general level. At the local database level, each setting is either enabled or disabled. At the study and user levels, each setting is either enabled, disabled, or not set. If the value of a setting is "Not Set" it serves as a "pass-through" to the next higher level value for that setting. Initially, all values at the study- and user-level are set to "Not Set" so that the database level, which is set up during installation, is in effect until you modify the settings for a given study or user.

Study-level configuration settings affect all users who have access to that study, except when user-level configuration or Study/Site security settings are set up for a user. If a setting at this level has a value of Not Set, all users who access the study use the database-level setting, unless the value for the setting at the user level is Enabled or Disabled for a specific user.

Note that certain privileges that are assigned via the Study and/or Site Security windows take precedence over user-level data entry configuration settings.

The settings available at each level are identical; the only difference is the availability of the "Not Set" value at the study- and user-levels. Refer to Table 5–1, "Local Database-Level Data Entry Configuration Settings" for a complete list of settings.

**Configure Database-Level Data Entry Settings**

Local database configuration settings are maintained in the Maintain Installation Configuration window. You can customize the configuration settings for the local database by changing the default shipped values for the configuration settings.

At the local level, configuration settings are either enabled, disabled, or have a numeric value, as applicable. Table 5–1 describes each setting.

Use this procedure to change the local settings:

1. Navigate to Admin, then DE Admin, and DE Config Settings. The Maintain Installation Configuration window opens.

2. Navigate to the configuration setting that you want to modify, and change its value. The default page height and width settings allow numeric values within upper and lower bounds. The other settings can be set to either enabled or disabled.

3. Click Save to commit changes.
<table>
<thead>
<tr>
<th>Configuration parameter</th>
<th>Description</th>
<th>Values</th>
<th>Default setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second-Pass Comparison Failure Alert</td>
<td>(Oracle Clinical only) Controls whether the data entry operator is notified when a first-pass/second-pass comparison error occurs. When this setting is not enabled, the operator is in &quot;silent&quot; mode during second-pass data entry, and any first/second pass comparison failures that occur must be resolved during comparison reconciliation.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Manual Discrepancy in Browse</td>
<td>(Oracle Clinical only) Whether the data entry operator can create or modify a manual discrepancy (operator comment) in browse mode (only applicable if the user has access to Browse mode).</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Resolve Discrepancies in Data Entry</td>
<td>(Oracle Clinical and RDC) Whether the data entry operator has authority to resolve discrepancies during data entry, that is, the permission to set a discrepancy to closed status. For RDC Classic, set this parameter to Enabled.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Privileged Update</td>
<td>(Oracle Clinical and RDC) Whether the data entry operator can perform the following while in update mode: ■ update data for locked RDCMs and RDCIs, ■ override protected repeating defaults, and ■ exceed the Maximum # of Repeats to a Repeating Question Group, even when Enforce Repeats is set (Oracle Clinical and RDC Classic only).</td>
<td>Enabled/Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>List of Values for Thesaurus Questions</td>
<td>(Oracle Clinical only) Whether a list of values is available for thesaurus questions based on external dictionaries.</td>
<td>Enabled/Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Univariate Failure Alert</td>
<td>(Oracle Clinical and RDC Classic) Whether the data entry operator is notified when a univariate validation error occurs. When this setting is disabled, the operator is in &quot;silent&quot; discrepancy mode. Univariate discrepancies are still created in &quot;silent&quot; mode, but the operator is not notified of their creation. RDC Onsite does not use this setting. Instead, it uses a configuration setting in the RDC Administration tool, which can be over-ridden by the end user, if Preferences are made available.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Initiate DE session using DCI Book</td>
<td>(Oracle Clinical only) Whether DCI book sequencing is the default sequencing mode during log-in and data entry.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Unenrolled patient alert</td>
<td>(Oracle Clinical only) Whether the data entry operator is notified when a received DCI is logged in for a patient not enrolled in the study.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Prevent Second-pass Entry by First-pass operator</td>
<td>(Oracle Clinical only) Whether Oracle Clinical prevents the data entry operator who did first-pass entry on a given RDCI from performing second-pass entry on the same RDCI.</td>
<td>Enabled/Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Browse accessible data only</td>
<td>(Oracle Clinical only) Determines if data entry operators can browse data.</td>
<td>Enabled/Disabled</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
Customizing Data Entry Behavior

Configure Study-Level Data Entry Settings

If you require that a particular study have different data entry configuration settings from those set at the local database level, you can change the settings at the study level by modifying the Clinical Study State record for that study. Study-level configuration settings override local database-level configuration settings for that study.

The study-level configuration settings are identical to the local database-level data entry configuration settings, but at the study level, an additional value, "Not Set", is available for all configuration settings. If a setting is not set, it is not defined and the next higher configuration setting takes effect. Because study-level settings take precedence over database-level, by default, all study-level configuration settings are set to "Not Set", which results in the local-level settings taking effect. Refer to Table 5–1, "Local Database-Level Data Entry Configuration Settings" for a listing of the settings.

Values for each non-numeric setting are available from the list of values and can be either "Enabled", "Disabled", or "Not Set". The default page height and width settings allow numeric values within upper and lower bounds.

To change a study-level data entry configuration setting:

1. Navigate to Conduct, then Security, and then Clinical Study States. The Maintain Study States window opens.

2. Query for the study you want to update. Choose DE Configs in the Special menu. The Maintain Study Configuration window opens.

3. For each configuration setting that you want to modify, change its value as necessary.

4. Save the changes and click Back to return to the Maintain Clinical Study States window, or click Back without saving to abandon your changes.

Changes at this level affect all users working in the study, unless user-level data entry configuration settings are defined for a user. Refer to "Configure User-Level Data Entry Settings" for information on modifying these settings.

Configure User-Level Data Entry Settings

User-level settings affect all studies to which the user has access.

If you require that a user have different data entry configuration settings from those set at the study or the database level, you can change specific settings at the user level that supersede those higher level settings.

### Table 5–1 (Cont.) Local Database-Level Data Entry Configuration Settings

<table>
<thead>
<tr>
<th>Configuration parameter</th>
<th>Description</th>
<th>Values</th>
<th>Default setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCI and DCM Date Required</td>
<td>(Oracle Clinical and RDC) The data entry must enter the DCI and DCM Visit date.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Default height for Data Entry page in DCM</td>
<td>(Oracle Clinical only) Default value for the Data Entry Page Height for DCMs. This setting determines only the value that is supplied as the default, which can be overridden during DCM definition.</td>
<td>Number between 10 and 60</td>
<td>22</td>
</tr>
<tr>
<td>Default width for Data Entry page in DCM</td>
<td>(Oracle Clinical only) The default value for the Data Entry Page Width for DCMs. This setting determines only the value that is supplied as the default, which can be overridden during DCM Definition.</td>
<td>Number between 10 and 240</td>
<td>80</td>
</tr>
</tbody>
</table>
The user-level configuration settings are identical to the study- and database-level data entry configuration settings. As with the study level settings, an additional value of Not Set is available for all nonnumeric settings. If a setting is Not Set, the next higher configuration setting takes effect. Because user-level settings take precedence over database-level, by default they are set to Not Set, which results in the local-level settings taking effect. Table 5–1, "Local Database-Level Data Entry Configuration Settings" has a complete list of settings.

---

**Note:** The DCI and DCM Date Required setting is inactive at the user-level.

---

Values for each nonnumeric setting are available from the list of values and can be either Enabled, Disabled, or Not Set. The default page height and width settings allow numeric values within upper and lower bounds.

To change user-level data entry configuration settings, follow this procedure:

1. Navigate to Admin, then Users, then Oracle Accounts. The Maintain Oracle Accounts multi-view window displays.
2. Query the user record you want to modify. Choose Special, DE Configs. The Maintain User Configuration window displays.
   
   To define a user-level configuration setting, change the value of the configuration setting in this window to any value other than not set. You can change the value of a configuration setting without changing all of them.

3. Change the any setting, as needed.
4. Save the changes and then click Back to return to the Maintain Oracle Accounts form, or click Back without saving to abandon your changes.

**Configuring Additional Data Entry User Preferences**

Oracle Clinical supplies a set of default values for user preferences, which are displayed in Table 5–2. These preferences remain in effect for all data entry operators unless operators override the defaults and save their own values (see Chapters 3 and 4 in Oracle Clinical Conducting a Study).

To change the default values for user preferences:

1. Navigate to Admin, then DE Admin, and then DE User Prefs. The Maintain Installation Preferences window opens.
2. Navigate to the user preference that you want to modify, and change its value. Values for each user preference are represented either as checkboxes or in Lists of Values.
3. Click Save as Default to save the changes, then click Exit to close the window.
<table>
<thead>
<tr>
<th>User preference</th>
<th>Description</th>
<th>Values</th>
<th>Default setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Skip</td>
<td>On fields that have been defined as Auto Skip fields in DCM definition, determines whether the cursor automatically skips to the next field when the current field is filled to its predefined length.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Auto Fill</td>
<td>Applicable only for fields that have lists of values (DVG- or thesaurus-based). Determines whether entering a few unique characters of a valid value will result in the system's filling in the rest of the value upon navigation out of the field.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Univariate Beep</td>
<td>Determines whether an audible signal will accompany the Univariate validation failure pop-up window. Not applicable if the Univariate validation failure alert configuration setting is disabled.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Comparison Beep</td>
<td>Determines whether an audible signal accompanies the First-Pass/Second-Pass Comparison Failure pop-up window. Not applicable if the second-pass comparison failure alert configuration setting is disabled.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>End of Form Beep</td>
<td>Determines whether you receive an audible beep when you reach the end of a form.</td>
<td>Enabled/Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Data Entry Input Format</td>
<td>For all log-in and data entry tasks, for DCI and DCM dates and data entry fields of type date, determines how date formats are applied to interpreting data that is input to the field. Must be the same as data entry display format (see below), unless display format is standard. Standard format (see below) is always accepted.</td>
<td>US (mm-dd-yyyy) European (dd-mm-yyyy) Swedish (yyyy-mm-dd)</td>
<td>US</td>
</tr>
<tr>
<td>Data Entry Display Format</td>
<td>For DCI and DCM Dates, the Display Format configuration determines the date format used for displaying dates. For data entry fields, the format is controlled by the date order of the DCM, which is either DYNAMIC or is one of the four date formats (US, EUROPEAN, SWEDISH, or STANDARD). If the DCM's date order is DYNAMIC, data entry fields that are dates are displayed using the same data entry configuration Display Format setting used for DCI and DCM dates. Otherwise, data entry fields are displayed using the format indicated by the Date Order for the DCM. Additionally, whether the day and month portion of a data entry field date is displayed is controlled by the Date Time Format of the DCM question on which it is based. There are four possible Date Time Formats for a date field: DMY (day, month, and year are displayed), MY (day is not displayed), and Y (only year is displayed).</td>
<td>US (MM-DD-YYYY) European (DD-MM-YYYY) Swedish (YYYY-MM-DD) Standard (DD-MON-YYYY)</td>
<td>US</td>
</tr>
<tr>
<td>RDCI Sort Order</td>
<td>For query sequencing, determines the ordering for the RDCIs retrieved by a query for data entry processing.</td>
<td>Document Number - Patient - Visit - DCI Name - DCI Date Entry Order</td>
<td>Entry Order</td>
</tr>
</tbody>
</table>
Configuring Privileged Update

This section includes the following topics:

- **Set Privileged Update Using Configuration Settings** on page 5-7—applies to Oracle Clinical and RDC
- **Set Privileged Update Using Study/Site Security** on page 5-7—applies to Oracle Clinical only

Users with privileged update can perform the same tasks on a locked document that they can on an unlocked document. In all data entry modes: log in, first pass, second pass, update, reconciliation, and key changes.

Within RDC, enabling privileged update means that the user can take any action consistent with the study- or site-security privileges assigned to the user’s name. For RDC purposes, privileged update can only be assigned through the data entry configuration settings.

In Oracle Clinical you can assign privileged update through data entry configuration settings and through study/site security.

- If Privileged Update is not enabled for a user through the data entry configuration settings, you can use the UPD_LOCK_OC privilege to grant Privileged Update access to a user. The advantage of this approach is that you can grant the privilege for a specific site within a study, while the Data Entry Configuration settings allow specification only at the database, study, or user level.
- If Privileged Update is enabled through the data entry configuration settings, the UPD_LOCK_OC privilege assigned through study/site security is not applicable.

### Set Privileged Update Using Configuration Settings

By default, privileged update is not enabled for a user. In most circumstances, you set the Privileged Update data entry configuration setting at the user level. To do this, refer to "Configure User-Level Data Entry Settings" on page 5-4.

- To set privileged update at the study level, refer to "Configure Study-Level Data Entry Settings".
- To set privileged update at the local database level, refer to "Configure Database-Level Data Entry Settings".

These settings apply to Oracle Clinical and RDC users.

### Set Privileged Update Using Study/Site Security

For Oracle Clinical users only, when Privileged Update is not enabled for a user in the data entry configuration settings, you can grant Privilege Update access by assigning the user the UPD_LOCK_OC privilege for a particular study or site within a study.

Navigate to **Admin, Users and Roles, Study Security** or **Site Security**.

---

**Customizing the Oracle Clinical Log-in Window Layout**

Oracle Clinical data entry operators use the log-in and data entry windows to enter and display the patient data defined by Study DCMs and DCIs (see Oracle Clinical Creating a Study). (Once entered, patient data, or documents, are called received DCIs (RDCIs) and received DCMs (RDCMs).) The appearance of these windows can be modified by the log-in layout editor tool to resemble the received DCI and DCM header information that appears on the CRF. You can change the header information...
field prompts, change the field sequences, or even hide fields after supplying them with default values.

This section includes the following topics:

- Using the Log-in Layout Editor
- Modify the Received DCI window
- Modifying the Received DCM Window
- Modifying the Smart Received DCM Window

Using the Log-in Layout Editor

The header information in CRFs is captured in the received DCI and received DCM windows. The Log-in Layout Editor allows you to match online screens with header information on the paper CRF that you may use for source data entry.

The log-in layout editor is a different tool from the DCM layout editor. You use the log-in layout editor to match the online Log-In windows with common CRF header information. You can change the header information field prompts, change the sequence of the fields, or hide fields that have default values.

"Customizing the Oracle Clinical Log-in Window Layout" on page 5-7 describes the log-in layout editor. Use the DCM layout editor to modify the appearance of the Data Entry windows (see "Laying out the data entry screen” in the Oracle Clinical Creating a Study manual).

The log-in layout editor can modify three windows:

- the Received DCI (RDCI) window,
- the Received DCM (RDCM) window, and
- the Smart Received DCM (Smart RDCM) window.

During log-in and data entry, the Received DCI window is the first screen that the system presents to a data entry operator. In all modes, under normal navigation, the window displayed immediately after the Received DCI window is the Smart RDCM window. This window displays underneath the Received DCI window and shows only the received DCM information required for context. To view all received DCM information, the user must navigate to the Received DCM window by invoking the function [RDCM].

To access the Log-in Layout Editor, navigate to Admin, the DE Admin, and Log-In Layout Editor. The RDCI window displays. Note the drop-down list field in the bottom-left corner of your screen. Use this window selection field to navigate between the windows. When you click on the arrow to the right of the drop-down list field, the names of all the windows are displayed. To change to another window, select it from the drop-down list. If you have changes pending when you attempt to change to another window, you are prompted to save your changes, or you can discard them.

Modify the Received DCI window

To modify the Received DCI window, select RDCI window from the drop-down list. The Received DCI window displays, showing all the received DCIs fields and their default prompts.
When you click on a field or on its prompt, the following information about the field or prompt is displayed to the right of the window selection window:

- the prompt or field name (depending on whether you clicked on the field or on its prompt),
- the X and Y coordinates for the field or prompt, and
- the length of the prompt or field.

These fields are display-only and cannot be updated.

You can change the length of a prompt or of a field by clicking on the prompt or on the field and then clicking on the Increase Width or Decrease Width buttons, which increase or decrease the length of the prompt or the field.

You can change the screen position of the field and its prompt by clicking on the prompt and dragging it to a new location. The field follows the prompt. If you click on the field itself and drag it, it moves separately from the prompt.

You can modify the prompt text by clicking on the prompt and then editing the text.

In the RDCI window, you are limited to six lines of vertical screen space, and 80 characters of horizontal space. In the data entry form, the window will only display the height of the screen occupied by fields or their prompts. For example, if you configure the RDCI window in the log-in layout editor such that you have fields and prompts on only the first four lines, only those four lines are displayed at data entry time, leaving more room for the data entry fields.

You can hide a field by dragging it to the Items Not Displayed section of the form. This field is not visible to the data entry operator during normal data entry functioning.

You can make a field non-updateable by the data entry operator by double-clicking on the field. When a field has been made non-updatable, it is displayed in red in the RDCI window. You can make it updateable by double-clicking on the field again.

Save pending changes by selecting Save. If you want to discard changes that you have made but have not yet saved, you can select Revert, which rolls back your changes to your last save. To close the form, select Exit.

To edit the layout of another window, click the arrow to the right of the Window Selection drop-down list field, and then select the window whose layout you next want to edit. If you have pending changes when you try to change windows, you are prompted to save your changes.

**Modifying the Received DCM Window**

The RDCM window is displayed to the data entry operator when the [RDCM] function is invoked.

To modify the RDCM window, select RDCM Window from the window selection drop-down list. You can modify the information that is displayed for the received DCM in the RDCM window in the same way that you modified the information in the RDCI window. All navigation and other behavior is identical. The only difference is that for RDCMs, you have twelve lines to work with vertically, rather than the six lines allotted to RDCIs. The horizontal restriction remains 80 characters for rdcms, the same as for RDCIs.

**Modifying the Smart Received DCM Window**

In all log-in and data entry modes, the Received DCI window is the first window the data entry operator sees. This window may be used to capture RDCI information in
Log-In modes, or it may be used only to display RDCI information, as in the data entry modes.

Additional RDCM-level information may need to be captured, or displayed for context. For this purpose during log-in and data entry, instead of the entire RDCM window, the Smart Received DCM window is displayed underneath the Received DCI window. The Smart RDCM window contains only those RDCM fields that may require user input, or that provide minimal context for the user. In addition, page fields are displayed to indicate which RDCM of the parent RDCI is currently being displayed. Full RDCM information is available to the data entry operator by invoking the [RDCM] function.

To modify the Smart RDCM window, choose it in Window Selection drop-down list field. The Smart RDCM window is displayed.

The following fields are displayed in the Smart RDCM window:

- Qualifying Value
- Clinical Planned Event Name
- Subevent Number
- Visit Number
- DCM Date
- DCM Time
- Lab Name

Because of the unique character of the Smart RDCM window, there are limitations on the changes that you can make to the fields in this window, as follows:

- You cannot change any characteristics of the Qualifying Value field.
- You cannot change the position of any of the fields.
- You cannot choose not to display one of the fields.

All six fields must be displayed, and their location cannot be changed. For Smart RDCM window fields other than the Qualifying Value, you can change the field prompt and length.

After you save your changes, the next time that a data entry operator performs a log-in function in that study, the changes that you made will be visible.

Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report

This section consists of the following topics:

- Setting and Enforcing Values on page 5-12
- Settings on page 5-12
  - DCI Form Definition
  - DCI Form Runtime
  - General
  - Graphic Layout Editor/Updater
  - Graphic Layout Generator - General
You set the default values for DCI Form and Graphic Layout settings for all studies in the current database in the DCI Form Local Database Settings window. These settings affect the appearance of data entry windows in RDC Onsite and the PDF output of the Patient Data Report and Blank Casebook Report in Oracle Clinical and RDC.

For each setting, you can choose to enforce the value across all studies or allow modification on the study level in the DCI Form Local Study Settings window (under Design).

By default the roles that have access to this form are RXC_USER, RXC_SUPER_NOGL, RXC_ADMIN. There is also a read-only Query version of this form (QRY Global Settings, in the same path). To query the form you must have either the RXC_SUPER or RXC_ANY role.

Access the window by navigating to Admin, then DCI Form Local Database Settings.

Settings are logically grouped, and when you open the window only the groups are displayed. To see individual settings, click the + node.

Figure 5–1  DCI Form Local Database Settings Window
Setting and Enforcing Values

For each individual setting you can choose to:

- Change the default value
- Select the **Enforce Local DB Setting** check box

If you select **Enforce Local DB Setting** here, study designers cannot change the value at the study level in the DCI Form Local Study Settings window. If you do not select **Enforce Local DB Setting** here, the value is modifiable at the study level.

For further information about the DCI Form Local Study Settings window, refer to the Oracle Clinical Creating a Study manual.

Settings

This section describes the following groups of settings:

- DCI Form Definition
- DCI Form Runtime
- General
- Graphic Layout Editor/Updater
- Graphic Layout Generator - General
- Graphic Layout Generation - DCMS
- DCI Form Generation Defaults
- Default Settings for Showing DCM Header Fields
- Default DCM Header Field Prompts
- Version Migration
- Patient Data Report
- Validation

**DCI Form Definition**

The settings for this category are:

- **DCI Form Definition Enabled** If set to Y, DCI Form definition—the use of graphic layouts—is enabled by default for all studies in the database. The study-level setting is not in the DCI Form Local Study Settings window but in **Clinical Study States** (under **Conduct**, then **Security**). The Easy Study Design feature does not include an explicit setting for enabling DCI Form definition; if study designers want to change the default value, they can use the Clinical Study States form.

- **GLIB DCI Forms Definition Enabled** If set to Y, DCI Forms can be defined in the Global Library (under **Glib**, then **DCMs DCIs Procedures**, then **DCMs** or **DCIs**). There is no corresponding study-level setting.

**DCI Form Runtime**

The settings for this category are:

- **Label for Customizable Patient Identifier** If you are using a customizable patient identifier and you would like to display a label other than Reference (the default) in RDC Onsite, enter the label text you prefer. The label appears next to the field in the Search screen of the Home and Casebooks pages. You can use this setting only
Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report

if Use customizable patient identifier? is set to Y. See the RDC Administrator’s Guide for further information.

- **DCI Form Entry Enabled** If set to Y, data entry in RDC Onsite is enabled. The study-level setting is not in the DCI Form Local Study Settings window but in the Clinical Study States window (under Conduct, then Security). The Easy Study Design feature does not include an explicit setting for enabling DCI Form definition; if study designers want to change the default value, they can use the Clinical Study States window.

- **DCI Form Field Length Restriction** If set to Y, users cannot enter more characters in a field than specified by the Length attribute on the DCM Question definition for that field. If set to N and a user enters more characters than specified, the system creates a discrepancy. Overflow data that cannot be displayed on the output prints into an overflow section.

- **Display Label for DCM Question** Select the source for the label of each field in the CRF; either the SAS label, the question name, or the default prompt of the corresponding question definition. The label is then used as a reference in the Discrepancies, Investigator Comments, and Audit History Navigators in the RDC Onsite data entry window.

- **Display Visit Owning Interval on MPC Page?** If set to Y, the Casebooks page in RDC Onsite displays the interval—phase, period, or subperiod—to which the displayed visit belongs.

- **Enable Entry of Investigator Comments** If set to Y, Investigator comments are allowed in RDC Onsite.

- **Page Labeling Compatible with Page Tracking?** If set to Y, the page label in a physical page uses the same syntax as the page identification in the page tracking system. This setting applies only to studies that use Page Tracking; refer to Oracle Clinical Creating a Study for more information.

- **Represent Disabled Blocks as** This setting applies to studies using conditional in-form branching. Select Greyed if you want conditional fields that are not expected for a patient to be displayed but grayed out. Select Hidden if you do not want such fields to be displayed at all. In this case, the next expected fields, if any, are displayed in the same area, so that there is no empty space in the middle of the page. The empty space appears at the end of the page.

- **Suppress Change Reason for new Responses** If set to Y, the data entry user is not prompted for a Change Reason the first time a response is entered even if the CRF has been previously saved.

- **Suppress Change Reason Prompt for New Investigator Comment** If set to Y, the Investigator is not prompted for a Change Reason the first time he or she enters a comment on a particular response.

- **Suppress Warning for Non-migrated CRFs** If set to Y, the data entry user does not receive a warning when working on a CRF that was entered via another user interface—Oracle Clinical, RDC Classic, or Batch Data Load—and the CRF has not been migrated to a DCI Form version. Also refer to the setting Allow Migration of Classic RDCIs? under the Version Migration category.

---

**Note:** You cannot uncheck Enforced for this setting. There is no corresponding setting at the study level, so the value you set here is automatically enforced.
Use customizable patient identifier? If set to Y, the system allows you to customize an additional patient identifier field for Search purposes in RDC Onsite. Values determined by your customization are stored in the Reported Reference column of the Patient Positions column in Oracle Clinical. You can change the label for the field using the Label for Customizable Patient Identifier setting. See the RDC Administrator’s Guide for further information.

General
The settings for this category are:

- **Default Unplanned Use Allowed for DCIs not in Book** If set to Y, DCIs not included in a DCI Book are available for unplanned use; the default setting in the DCI Book Constraints window for the Unplanned Use Allowed if not listed below field is checked.

  This setting has no effect on existing DCI Books or on DCIs already listed in the DCI Book Constraints window.

  Refer to the section on DCI Books in the Oracle Clinical Creating a Study manual for more information.

- **Layout Unit of Measurement** Select the unit of measurement to be used when dimensions related to layouts are displayed. The options are: inches, centimeters, and points.

Graphic Layout Editor/Updater
This category has one setting: Enforce Length as Field Size. If set to Y, the system uses the character length defined for the question to set the minimum size of the field in the layout editor. When set to Y, if you increase the question's length in the Study DCM Questions window, the system sets the Needs Update flag to indicate the field width needs to be increased.

**Note:** This setting has no effect if the DCM Question Attribute for Determining Field Width setting, which a constituent of Graphic Layout Generation - DCMS is set to DISPLAY LENGTH.

Graphic Layout Generator - General
The settings in this category are:

- **Default Checkbox Check Style** Select the default value for the Checkbox Style field in the startup dialog for the DCM Graphic Layout generator, which determines the symbol used in selected checkboxes. The default options are: Check, Circle, Cross, and Square.

- **Default Checkbox Shape** Select the default value for the Checkbox Shape field in the startup dialog for the DCM Graphic layout generator, which determines the checkbox shape. The options are: Circle and Square.

- **Default Checkbox Size** Select the default point size for checkboxes. The options are: 10, 12, 15, 20.

**Note:** You can change the set of options by modifying the DCIF CHECKBOX SIZE installation reference codelist.
Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report

- **Default Landscape Form Layout Template** Select the default layout template that the system uses for horizontal layouts. The list of values is populated by the templates that are available in the database.

- **Default Portrait Form Layout Template** Select the default layout template that the system uses for vertical layouts. The list of values is populated by the templates that are available in the database.

- **Field Font Size** Select the default font point size for fields. The options are: 8, 9, 10, 11, 12, 14.

  **Note:** You can change the set of options by modifying the DCIF FONT TYPESIZE installation reference codelist.

- **Field Font Typeface** Select the default font for fields; must be a monospace font. This list is based on the DCIF Typefaces table, which is not modifiable. In Oracle Clinical Release 4.6, this list includes only Courier.

- **Prompt Font Size** Select the default font point size for prompts. The options are: 8, 9, 10, 11, 12, 14.

  **Note:** You can change the set of options by modifying the DCIF FONT TYPESIZE installation reference codelist.

- **Prompt Font Typeface** Select the default font for prompts. The options are: Arial, Courier New, Symbol, and Times New Roman.

**Graphic Layout Generation - DCMS**

This category has one setting: **DCM Question Attribute for Determining Field Width**. Select the question attribute to use to determine the size of the field:

- If set to **Length**, the display area accommodates the maximum number of characters allowed for the question, without scrolling. If the page is not wide enough to accommodate the field on one line, the layout generator changes it to a multi-line field.

- If set to **Display Length**, the display area may not be large enough to see the full response at one time. If the page is not wide enough to accommodate the field, the layout generator will extend the field to the page margin, but will not change it to a multi-line field.

The user can scroll to view or edit the overflow of text that might occur. A Patient Data Report (PDR) that includes such a field displays the entire value in the Overflow Section of the report.

  **Note:** Refer also to the Enforce Length as Field Size setting.

**DCI Form Generation Defaults**

The settings in this category are:

- **Default Landscape Page Definition** Select a default page size for horizontal pages: either US letter (OCL_USL_L) or A4 (OCL_A4_L).

- **Default Portrait Page Definition** Select a default page size for vertical pages: either US letter (OCL_USL_P) or A4 (OCL_A4_P).
Note: These settings have the same list of values, which is populated by the DCIF PAGE DEFINITION installation reference codelist. Be careful to select a landscape value for the landscape setting and a portrait value for the portrait setting.

**Default Settings for Showing DCM Header Fields**

This category controls the default display of DCM header field definitions.

Note that you use next category, Default DCM Header Field Prompts, to define default text for these fields’ labels.

The settings for this category are:

- **Default for Show Blank Flag?** If set to Y, the DCM header includes a Blank Flag field.

- **Default for Show Comment?** If set to Y, the DCM header includes a Comment field, also known as the DCM Internal Comment. The DCM header data comments are available only in the Oracle Clinical Data Entry subsystem. You cannot modify the field in RDC Data Entry. Set this option to Y only if you want to produce Patient Data Reports that show RDCM header comments entered in through Oracle Clinical Data Entry.

- **Default for Show Data Comment?** If set to Y, the DCM header includes a Data Comment field. The DCM header data, which is the same as internal comments, is only available in the Oracle Clinical Data Entry subsystem. You cannot modify the field in RDC Data Entry. Set this option to Y only if you want to produce Patient Data Reports that show RDCM header data comments entered in through Oracle Clinical Data Entry.

- **Default for Show Lab?** Set to Y to enable displaying and entering the lab for an RDCM by default if the lab has any lab questions. If there are no lab questions for a DCM, Show Lab is set to N regardless of this setting.

- **Default for Show Qualifying Value?** Set to Y to display the qualifying value. If there is a qualifying question for the DCM but there is no default value, Show Qualifying Value has a value of Y even if this value is set to N. If there is no qualifying question for a DCM, Show Qualifying Value has a value of N for the DCI module record regardless the value of this setting.

- **Default Visit Display Code** Select the way you want the DCM header to display visit information. The options are:
  - NAME /SUB# - Visit Name, Subevent Displayed in separate fields
  - NAME+ SUB# - Visit Name, Subevent both displayed in Visit field
  - NAME ONLY - Visit Name

- **Hide Visit by Default?** If set to Y, the DCM header does not include the visit identifier. The system sets Visit Display Code to HIDDEN by default, overriding the previous setting. Exceptions: if there is no defined clinical planned event, or the Use DCI Date setting is not selected, you cannot select value HIDDEN for the Visit Display for a DCM, and Visit Display Code defaults to the value you set for the previous setting.

**Default DCM Header Field Prompts**

You control the default display prompts of DCM header field definitions in this category. You control the display of these definitions in the previous category, see "Default Settings for Showing DCM Header Fields" on page 5-16.
- **(Internal) Comment prompt** Enter prompt text for an internal comment field.
- **Blank Flag prompt** Enter prompt text to identify a header indicator that a DCM is blank.
- **Data Comment prompt** Enter prompt text to identify a DCM header data comment field.
- **Date prompt** Enter prompt text to identify a DCM header Date field.
- **Generate DCM Header Divider?** If set to Y, the system generates a line between the DCM Header and the DCM.
- **Lab prompt** Enter prompt text to identify a DCM header Lab identifier field.
- **Length for (Internal) Comment Prompt** Enter a number to determine the maximum number of characters a comment field can hold.

**Note:** This field is misnamed. It is not the length of the prompt, but the length of the comment field itself.

- **Length for Data Comment** Enter a number to determine the maximum number of characters a data comment field can hold.
- **Length of Visit Name** Enter a number to determine the maximum number of characters a Visit Name field can hold.
- **Subevent Prompt** Enter prompt text to identify a DCM subevent identifier field.
- **Time prompt** Enter prompt text to identify a Time field.
- **Visit Name Prompt** Enter prompt text to identify a Visit Name field.
- **Visit Name+Sub# Prompt** Enter prompt text to identify the Visit Name and subevent identifier field combination.

### Version Migration

If the data definitions that comprise a DCI Form change after a study has gone into production, you need to create a new layout version. These settings control whether and how to allow existing data to be migrated.

- **Allow Migration of Approved Documents?** If set to Y, approved RDCIs (collected patient data) are included whenever patient RDCIs are migrated to new DCI Form versions.
- **Allow Migration of Classic RDCIs?** If set to Y, patient RDCIs entered in Oracle Clinical or RDC Classic data entry are included whenever RDCIs are migrated to new DCI Form version(s).

**Note:** It is not necessary to migrate such RDCIs in order to open them in RDC HTML data entry.

If set to N, such RDCIs are not included in the migration. There are two other settings dictating whether or not users can open non-migrated CRFs, and if they can, whether a warning message will be issued.

- **Allow HTML data entry for non-migrated CRFs** This setting is available in the RDC Administration form under RDC Configuration models. If set to N, non-migrated CRFs cannot be opened in HTML data entry.
Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report

- **Suppress Warning for Non-migrated CRFs** This setting is found under the category DCI Form Runtime. If set to N, whenever a user opens a non-migrated CRF in HTML data entry, a warning message is issued: "Please Note: The data displayed on this form was originally entered using another interface."

- **Allow Migration of Locked Documents?** If set to Y, locked RDCIs (documents) are included whenever patient RDCIs are migrated to new DCI Form versions.

- **Default Reason to Retain Approval Verification** Select the default reason to supply if approvals or verifications are retained during DCI Form version migration. You must create the available values in the APPROVE VERIFY RETAIN CODE installation reference codelist.

- **Default Reason to Reverse Approval/Verification** Select the default reason to supply if approvals or verifications are reversed during DCI Form version migration. You must create the available values in the APPROVE VERIFY REVERSE CODE installation reference codelist.

- **Default Setting for Reverse Approval Status** If set to Y, DCI Form version migration changes approved RDCIs' approval status to Unapproved. If set to N, DCI Form version migration keeps approved RDCIs' approval status as Approved.

- **Default Setting for Reverse Verification Status** If set to Y, DCI Form version migration changes verified RDCIs' verification status to Unverified. If set to N, DCI Form version migration keeps verified RDCIs' approval status as Verified.

- **Last Migrateable Entry Status** Specify the highest status at which CRFs are included in version migration. The possible statuses are, in order from lowest to highest, with the Oracle Clinical term given first and the RDC equivalent following:
  - Received (Blank)
  - (not applicable) (Created)
  - Pass 1 Started (Entry Started)
  - Pass 1 Complete (Entry Complete)
  - Batch Loaded (not applicable)
  - Pass 2 Pending (not applicable)
  - Pass 2 Started (not applicable)
  - Pass 2 Complete (not applicable)

In addition to these statuses, the keyword ALL allows RDCIs at any status to migrate, and the keyword NONE disallows any RDCI from migrating.

- **User Override to Reverse Approvals?** If set to Y, the user running Form Version Migration can specify whether that particular execution of Form Version Migration should reverse the status of all approved RDCIs migrated and can select a different reason for the reversal, if another option is available.

If set to N, the user running the migration cannot change the setting you selected for **Default Setting for Reverse Approval Status** and cannot change the default reason you set in **Default Reason to Retain Approval Verification** or **Default Reason to Reverse Approval/Verification**.

- **User Override to Reverse Verifications?** If set to Y, the user running Form Version Migration can specify whether that particular execution of Form Version
Migration should reverse the status of all verified RDCIs migrated and can select a different reason for the reversal, if another option is available.

If set to N, the user running the migration cannot change the setting you selected for Default Setting for Reverse Approval Status and cannot change the default reason you set in Default Reason to Retain Approval Verification or Default Reason to Reverse Approval/Verification.

**Patient Data Report**

The settings in this category are:

- **Bookmark Ancillary Data Section** If set to Y, the system generates bookmarks for the Ancillary Data sections of the Patient Data Report (PDR).

- **Bookmark Subevents** If Y, the system generates bookmarks for Visit Subevents in the PDR.

- **Bookmark Title for Ancillary Data Section** Specify a title to be used for bookmarks to the ancillary data section for a CRF (if Bookmark Ancillary Data Section is set to Y). The default value is "Ancillary Data Section". In the bookmark, the system appends the word "for" followed by the bookmark label of the CRF to the value specified. Therefore, with the default value the bookmark text is "Ancillary Data Section for CRF bookmark label.

- **Exclude Overflow for Hidden Protected Repeating Defaults** The Patient Data Report includes all default text for repeating questions in the ancillary pages. Set to Y if you do not want to include text for repeating default questions if they are hidden.

  If set to Y and the CRF response field for a protected repeating default is less than one character long, the Overflow section of a Patient Data Report does not list the default values for the field. This setting provides support for a mechanism to hide certain fields in a CRF simply by restricting the field length to less than 1 character.

- **Include Approval Information** If set to Y, approval information for the CRF is included in the ancillary data section. A line appears under the title of the report stating that the document was approved, who it was approved by and the date and time of approval. If the CRF is approved but has no other ancillary data, the ancillary data page is included with just the approval information.

- **Include Audit History for Fields Not Displayed in CRF** This setting has effect only when Audit History is selected when the PDR is submitted. If set to Y, the audit history for CRF fields that are not displayed in the CRF is displayed at the end of the Ancillary Data Section. It is not attached to a superscript but lists all audit information for fields that are not displayed on the form—for example, if the blank flag was changed for a CRF but the blank_flag is not displayed in the form. If set to N, the audit history is not displayed for undisplayed fields.

- **Include TOC in Page Numbering** If set to Y, the cover page and table of contents are counted when determining PDR page numbers. For example, if CONMED is the first domain, and the cover page and table of contents each consisted of one page, CONMED would begin on page 3 if Include TOC in Page Numbering is set to Y and on page 1 if it is set to N.

- **PDR Bookmark Data Domain** Select DCI if you want Patient Data Report bookmarks to be at the DCI level, or DCM if you want bookmarks at the DCM level. See Oracle Clinical Creating a Study for information on DCIs and DCMs. **Enforce Local DB Setting** is checked and cannot be unchecked. The setting cannot be changed at the study level. The default value is DCI.
Validation

This category contains one value: **Execute TMS validation during site/patient validation?**. If set to Y, TMS processing is executed during site and patient validation. If a question is defined as a TMS parent question, the value is sent to TMS immediately and, if the value can be autoclassified in TMS, the derived responses are sent back. However, during patient validation TMS processing is always performed for the study as a whole, including for sites to which the current user may not have access, and the audit trail represents the changes as having been made by the user who invoked patient validation.

To avoid this, turn off TMS validation entirely in the context of patient validation by setting this parameter to N. TMS processing still occurs during batch validation.

Customizing Flex Fields for DCI Forms

The purpose of flex fields is to allow the CRF designer to include fields in the CRF header and/or footer that display data based on functions that you define. The input parameters to the function include data specific to the current document, such as document number, investigator id, etc., which allows you to include information based on these parameters in the document.

You add flex fields to your CRF design using the Form Layout Template (FLT) Layout Editor. The system allows you to define up to ten flex fields, any or all of which can be included in an FLT. Refer to *Oracle Clinical Creating a Study* for information about the procedure to insert a flex field in a FLT.

This section describes the flex fields that are defined by default and outlines the procedure you use to customize and activate additional flex fields.

- How Flex Fields Work on page 5-20
- Flex Field Components on page 5-21

How Flex Fields Work

Each flex field is customized through the use of a function in the package `rdcpbh_client.sql`. When the flex field is properly configured, it can be added to the header and/or footer, using the FLT layout editor.

The function is called by the FLT layout editor.

- RDC Onsite runtime uses Flex Field Name (pKey) and Value (pValue).
- The GLE uses the Flex Field Name (pKey) and Description (pDescription) parameters to populate the

At runtime, any flex fields that are in the DCI header or footer are displayed and are populated with the value returned by a call to the function that is associated with the field. By default, if there is an error in the function, the system returns NULL. No error message is provided. You can modify the program for debugging purposes so that if there is an error in the function, the system returns an error message, which it displays in the relevant flex field.

There is no separate audit history for flexfield values that change due to modifications of the underlying function call or as the result of changes to any data points upon which a function is based.
Flex Field Components

The functions associated with flex fields are declared in rdcps_client.sql and are written in rdcpb_client.sql. These files are copied to the $RXC_INSTALL directory during the server installation. They are executed against the database when you upgrade or install it.

The flex field functions are defined as follows:

- Function name is of the form "FLEX_FIELDn", where n is an integer from 1 to 10, inclusive
- Flex field name is of the form "RDCI_FLEX_FIELDn", where n is an integer from 1 to 10, inclusive, and links the flex field name to the function name.

The parameters that are available in the flex field functions are listed and described in

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pTestProd</td>
<td>Describes the current mode; value is &quot;P&quot; if called from production mode, &quot;T&quot; if called from test mode. Refer to the shipped functions for examples of usage.</td>
</tr>
<tr>
<td>pRdcPopulating</td>
<td>Describes if the function is called during runtime or from the layout editor.</td>
</tr>
<tr>
<td>pUserId</td>
<td>The Oracle account of the person logged in to the system.</td>
</tr>
<tr>
<td>pUserRole</td>
<td>The RDC role of the user who is logged in to the system. If the user has multiple roles, it takes the first one on the list.</td>
</tr>
<tr>
<td>pStudy</td>
<td>The name of the current study.</td>
</tr>
<tr>
<td>pStudyId</td>
<td>The ID associated with the current study.</td>
</tr>
<tr>
<td>pStudyVerId</td>
<td>The version of the live study.</td>
</tr>
<tr>
<td>pDocNum</td>
<td>The document number of the CRF.</td>
</tr>
<tr>
<td>pBook</td>
<td>The name of the book to which the CXRF is assigned.</td>
</tr>
<tr>
<td>pBookId</td>
<td>The Book ID of the book to which the CRF is assigned.</td>
</tr>
<tr>
<td>pDci</td>
<td>The name of the DCI for the CRF.</td>
</tr>
<tr>
<td>pDciId</td>
<td>The DCI ID of the DCI for the CRF.</td>
</tr>
<tr>
<td>pSite</td>
<td>The site of the CRF.</td>
</tr>
<tr>
<td>pSiteId</td>
<td>The Site ID of the site for the CRF.</td>
</tr>
<tr>
<td>pInv</td>
<td>The name of the investigator for the site.</td>
</tr>
<tr>
<td>pInvId</td>
<td>The investigator ID of the investigator for the site.</td>
</tr>
<tr>
<td>pPatient</td>
<td>The patient associated with the CRF.</td>
</tr>
<tr>
<td>pPatientId</td>
<td>The patient ID of the patient associated with the CRF.</td>
</tr>
<tr>
<td>pCpeName</td>
<td>The name associated with the clinical planned event (CPE) for the CRF.</td>
</tr>
<tr>
<td>pCpeId</td>
<td>The CPE ID of the CPE for the CRF.</td>
</tr>
<tr>
<td>pSubNo</td>
<td>The subevent number for the CPE.</td>
</tr>
<tr>
<td>pKey</td>
<td>The name of the flex field, for example, &quot;FLEX_FIELD1&quot;.</td>
</tr>
<tr>
<td>pValue</td>
<td>The value that is displayed in the flex field. This is only applicable when</td>
</tr>
</tbody>
</table>
Customizing CRF Column Headers in the RDC Classic Surround

You can customize the three rows of information in the CRF Column Header in the RDC Spreadsheet using functions that are supplied.

- Default Behavior on page 5-23
- Functions Used to Modify CRF Column Headers on page 5-23

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| pDescription  | The prompt that is used to designate the flex field in the

**pRdcPopulating**

This parameter returns "Y" if it is called from the runtime environment. It returns "N" if it is called from the layout editor. When it is set to "N", most of the other parameters are "NULL". The select statements that you use to return a flex field value should only be executed when the value of this parameter is "Y". SQL statements based on the parameters that process successfully during runtime (pRdcPopulating = 'Y') will generally fail when called from the layout editor (pRdcPopulating = 'N'). Refer to FLEX_FIELD1 and FLEX_FIELD2 in rdcpb_client.sql for examples of this parameter's usage.

**Shipped Functions**

Oracle Clinical ships with two "starter" functions and seven undefined functions:

1. FLEX_FIELD1 – Patient Initials
2. FLEX_FIELD2 – Investigator Name
3. FLEX_FIELD3 through FLEX_FIELD10 – Null

The purpose of the shipped functions is to give you an example of how to write the functions to return values based on input parameters that come from the current document. You can modify the undefined functions or the defined functions to create flex fields functions that match your particular business needs.

The functions are coded so that exceptions in the SQL statements are handled silently. This ensures that errors in flexfields do not prevent RDC data entry. For de-bugging, we recommend that you use opa_trace.tableon, opa_trace.debugon, and opa_trace.debugmsg(msgstring) functions to record debug information in the opa_debug table.

The select statements used for deriving values based on the key parameters (from pStudy to pCpeName) are inside the 'IF' clause, "if pRdcPopulating='Y'". If you put the statements outside of this 'IF' clause, the function will fail when called from the layout editor when these values are null.
Default Behavior

The CRF Column header consists of three lines of information, which, when taken together, uniquely identify a set of CRFs. When a column intersects a Patient row, it results in a CRF cell, which uniquely identifies a CRF for which data has been, or will be, collected for a specific patient.

By default, the information displayed in a CRF Column header is:

- Line 1 – Visit name; if the visit is unplanned (such that the subevent is not zero), it also includes the subevent number, using dot notation
- Line 2 – Page number
- Line 3 – DCI/CRF name

The character width of Line 1 is 11 characters. If the display value is greater than 11 characters, it is truncated. If the display value is 11 characters or less and the font size and width of characters cause the value to extend beyond the physical length of 11 characters, the user must scroll to view the entire value.

Lines 2 and 3 are not truncated. If the value of either exceeds the physical width of the column header, it is wrapped to a new line.

Functions Used to Modify CRF Column Headers

You have the option to modify any or all of the CRF Column Header lines using a separate set of functions.

Functions

You have the option to modify any or all of the CRF Column Header lines using a separate set of functions, which are included in the file rdcpb_client.sql. When you customize all three lines, the values are displayed in full and are not truncated. Wrapping may occur depending on the font, the font size, and the width of the characters returned in the value. If wrapping occurs on either Line 1 or 2, the subsequent lines are displayed lower down in the CRF Column Header area. This may result in making all or part of the Line 3 value unviewable in the RDC Spreadsheet.

Although you have the option of customizing all three lines of the column header, if the combined length of text for the three values exceeds 50 characters, the system
displays a warning message (number 988300) and the combined text is truncated at 50 characters. The RDC user will then have to exit the application.

The three functions that can be customized are:

1. GetReadableColumnHeaderLine1 – customize Line 1 (Visit Name)
2. GetReadableColumnHeaderLine2 – customize Line 2 (Page No)
3. GetReadableColumnHeaderLine3 – customizes Line 3 (DCI/CRF Name)

**Note:** The GetReadableColumnHeaderLine1 function takes precedence over GetReadableVisitName. That is, if both functions are customized, the system displays the value returned by GetReadableColumnHeaderLine1.

All three functions have the same input parameters, which are as follows:

- **vTabType** Spreadsheet view tab type; like VISIT, STUDY or PHASE
- **vStudy** The name of the study
- **vBook** The name of the book
- **vVisit** The name of the visit
- **vSubno** The subevent number for the visit.
- **vPage** The page number
- **vDciSName** The short name of the DCI
- **bPlanned** A boolean operator that indicates whether the CRF is planned or unplanned.
- **vDefault** The default value for the line that would normally be displayed in the RDC Spreadsheet.

### Customizing Online Help

Oracle Clinical provides utilities for implementing a customized version of the system's extended help (Xhelp), which is HTML-based and context-sensitive. By setting up custom help through the system, you can activate a button in Oracle Clinical's field help window that, when clicked by an end user, displays your information. The system determines the user's environment by determining the form, block, and task for the current focus. You can activate or deactivate both Oracle Clinical's help (the More button) and your own (the Custom Help button).

Oracle Clinical's help system has a three-frame HTML interface that has embedded Java script macros and XML metadata. However, the application should work with any file type that your computers recognize. You can write topic files to suit your needs and completely ignore our help system, or you can copy our system into a new directory, rewrite the topic files of your choice, and then configure calls to them.

- Modifying Calls to Online Help Topic Files on page 5-25
- Copying Xhelp Topic Files on page 5-26
Modifying Calls to Online Help Topic Files

You can create customized HTML online help files with a context sensitive call from any window in Oracle Clinical. To see your online help, user click Help and then Custom.

You must create your own HTML files, put them in an accessible location, and insert the URL for the Help for each window in the OCL_DOC_INDEX table in the Developer’s Toolkit.

The Oracle Clinical Help engine determines which topic file to open by comparing the user's current environment to environments listed in table OCL_DOC_INDEX. The environment is the combination of module, task, and block values. You change these calls with the Document Index form. To access this form, you must have access to the Oracle Clinical Developer's Toolkit. Navigate to Admin, then Client Doc Index. The Maintain Client DOC Index window displays. It has these fields:

- **Module Name**: The code module, or form name. You can find this value in any form by navigating to Action, then Environment.
- **Task Name**: A case-sensitive string that often resembles the screen name. Many forms have task names that distinguish browse mode from write mode. You can find this value in any form by navigating to Action, then Environment.
- **Block Name**: A subdivision within the form. Some forms have one block. Others have many. You can find this value in any form by navigating to Action, then Environment.
- **Field Name**: You can make calls that are context sensitive to the field level. If you leave this field blank, the system drops through to the block level. You can find this value by invoking Help and reading the value from the field window.
- **Show Oracle Clinical Help**: This check box controls the More button. If you un-check it, you deactivate Oracle Clinical’s packaged help topic for the current environment.
- **Oracle Clinical Doc Name**: This is Oracle Clinical's topic identifier. Our calls contain several parameters for invoking our help system. The Oracle Clinical Doc Name is the second half of a URL. The first half is the value of a Web Server registry variable. The help engine concatenates the two parts, invokes a new browser instance, and passes the URL to the browser.
- **Show Client Help**: This check box activates the Custom button.
- **Client Doc Name**: This is the value you enter to create a custom call to your help topic. The Client Doc Name is the second half of a URL. The first half is the value of Web Server registry variable OPA_CUSTOM_DOC_DIR.
- **Product**: The value can be AERS, RXA, RXC, or TMS.

---

**Note:** Context-sensitive calls to custom help is available from within Oracle Clinical only. There is no similar mechanism for RDC.

---

To modify a call to the Xhelp topic files:

1. Navigate to Admin, then Client Doc Index to open the Maintain Client DOC Index window.
2. Query the module, task, or block name field for the form with the help you want to modify. (This information is available by opening the form and choosing Environment in the Action menu to display the Environment window.)
3. Set the **Show Oracle Clinical Help** and **Show Client Help** checkboxes as necessary for your system.

4. Enter your path/filename in the **Client Document Name** field, in the same row as the environment it references.

---

**Note:**

- Calls are case sensitive. Windows Explorer may not give an accurate view of the case of a filename, or a directory name.
- Browsers use forward slashes (/) as directory separators. If a call displays the HTML file at the end of the file, the call probably contains a backslash (\).

---

**Copying Xhelp Topic Files**

If you copy the shipped Xhelp to a separate directory, you can customize the content in those files, yet retain their HTML hyperlinks. You then activate your system by setting your OPA_CUSTOM_DOC_DIR registry string value to point to the duplicate directory.

**Creating Custom Help Files**

The Oracle Clinical Xhelp system provides flexibility to link to files that you create from within the online help. However, if you do create custom help files do not place them under the \html\xhelp directory. This directory may be over-written during product and/or documentation upgrades. We suggest you create another directory under \html, for example, \html\custom_xhelp.

---

**Note:** If you placed custom help files in the \html\xhelp\oc or the \html\xhelp\rdc directory trees in earlier versions of Oracle Clinical/RDC, you must create backups of these directories during the upgrade process.
This section contains the following topics:

- Configuring Default Installation Data Extract Settings on page 6-1
- Setting Values in Data Extract-Related Reference Codelists on page 6-5
- Creating Tablespaces for Data Extract Tables and Indexes on page 6-5
- Customizing Data Extract Views on page 6-6
- Generating Data Extract Views on page 6-7
- Enabling the View Builder and Converting Views on page 6-8
- Controlling Access to Data Extract Views on page 6-9

Configuring Default Installation Data Extract Settings

In the DX Installation Configuration window, you determine the default settings for new studies. To launch the Data Extract Installation Configuration window, navigate to Admin and select DX Installation Configuration.

The DX_CONFIG installation reference codelist includes exactly the same settings. When you change a setting here, the change is reflected in the reference codelist. When you change a setting there, the change is reflected here.
Configuring Default Installation Data Extract Settings

You can enable or disable the attributes described in the upper part of the window by selecting or clearing the appropriate box. The lower part of the window enables you to reduce the size of comments and the DVG long value, and to choose the default Key Template and the default Key Template domain.

The settings in the Data Extract Installation Configuration window are:

**Separate Oracle and SAS Names?**
When enabled, you can specify different names for Oracle and SAS view columns. Oracle views take the long name; SAS takes the short name. The default is deselected.

In earlier versions of SAS (such as version 6.12), the maximum length for variables (columns or views) was 8 characters, while Oracle names could be as long as 30 characters. If you wanted to keep the same names for the Oracle and SAS variables (for consistency or some other business need), you had to choose a name short enough to fit in the SAS variable length. If you wanted to have a longer Oracle name, the names had to be different. Data Extract creates the Oracle and SAS views based on the decision you make in this field.

**DCM Default Views Are Linked to Source DCM as Default Condition?**
This setting controls whether a view definition is linked to its source DCM if the view definition’s link mode is DEFAULT. If this setting is enabled, a view definition with DEFAULT link mode will be linked to its DCM, meaning that changes to the DCM will propagate to the view definition as well. If this setting is not enabled, these view definitions are not linked, so they will not change when the source DCM changes.
Enable Edit of Active Key Templates?
Enable Edit of Active Extract Macros?
Enable Edit of Active View Templates?

Each of these settings enables you to choose whether users can modify one type of active component in a view definition. Your organization may want to freeze definitions like Key Templates, extract macros, and View Templates that are used across many view definitions in the global library.

Enable Selection of Aggregate, Nondefault Key Template?

Choose this setting if you want to be able to choose alternative Key Templates for different view definitions within a study.

Include Validation Status in Default View Definition?
Include DVG Sequence Number in Default View Definition?
Include DVG Short Value in Default View Definition?
Include DVG Long Value in Default View Definition?
Include Thesaurus Term1 in Default View Definition?
Include Thesaurus Term2 in Default View Definition?
Include Thesaurus Term3 in Default View Definition?
Include Full Value Text in Default View Definition?

These settings all control attributes that you might want to include in the default view definition. All are part of what you can add through the Extended Attributes button when defining a simple question in the Global Library, or through the Template Attributes button when building a template in the Maintain View Templates window.

- **Validation Status** is an attribute of the RESPONSES table. By choosing to display another attribute, you can tell how clean your data is. The default is deselected.

- The **DVG Sequence Number** indicates the order the discrete values appear in the list of values for data entry. The **DVG Short Value** is the data as entered. The **DVG Long Value** is a longer form than the short value of the data as entered. You need at least one, but you may pick all, of the following: the DVG Long Value, the DVG Sequence Number, or data values for the DVG questions.

- The **Thesaurus Term** configuration preferences involve the same kinds of choices as for the DVG, except that data can come from several different tables. You must still choose at least one term, and you may choose all three. The default is selected.

- Oracle Clinical stores valid responses in the Value Text field, and invalid ones in the Exception Value Text field. When the response is valid, the **Full Value Text** field contains the Value Text; when it is invalid, **Full Value Text** contains the Exception Value Text.

Enable Update of SAS and Oracle Column Names?

When you bring a question from the Global Library into a View Template, the SAS column names and Oracle column names in the View Template default to the names defined in the Global Library. If this option is selected, you can change the names at the View Template level; if not selected, you cannot modify the names from their Global Library-derived defaults. The default setting is deselected.

Enable View Builder as Default in New Studies?

The View Builder enables you to automatically generate views of the data and metadata included in a single Data Collection Module (DCM). If this setting is enabled, the **VB Enabled?** setting in the Clinical Study States window is selected by default for new studies.
Use DCM Question-Specific DVG Subset for DVG Attributes?
This setting determines whether the Discrete Value Group (DVG) attributes that are included in the view come from the DVG subset that has been assigned to the DCM question or from the base DVG subset. Selecting this box makes the views include the DCM question DVG's subset information; clearing the box makes the views include the base subset information. For information on DVGs and DVG subsets, see the chapter on questions in Oracle Clinical Creating a Study.

Use DCM SAS Label as Seed for Attributes in Default View Definition?
This setting determines which SAS labels the system uses for all the attribute columns of a default view definition. By default, this setting is not enabled, so the SAS labels of the attribute columns are created using the SAS label of the question attributes in the Global Library. However, when you enable this setting, the system creates the SAS labels of the attribute columns of the View Template within the context of a view definition by using the corresponding DCM question's SAS label as the seed when the View Template Question is mapped.

Max Length of Audit Comment
The default Audit Comment length is 200 characters. You can reduce this value if you typically use no more than a few characters for this comment.

Max length of Data Comment
The Oracle Clinical default Comment length is 200 characters. You can reduce this value if you typically use no more than a few characters for this comment.

Max Length of DVG Long Value
The default DVG Long Value length is 200 characters. You can reduce this value if you typically use no more than a few characters for this comment.

You can also create the DVG Long Value column with a maximum width equal to the DVG values specified for a given question. This behavior is enabled when the maximum length of the DVG Long Value is set to zero.

Default Key Template
The default Key Template for custom and default view definitions. You can choose a new default Key Template from the list of values.

Data extract users can choose a non-default Key Template for their view definition only if the Enable Selection of Aggregate, Nondefault Key Template? box is selected.

Note that study-specific Key Templates achieve the same goal. You can supply a study-specific Key Template in the Clinical Study States window (From the Conduct menu, select Security, then choose Clinical Study States).

Key Template Domain
The Key Template Domain indicates the Global Library domain in which the default Key Template is stored. You cannot assign or change the domain of the default Key Template in this window.

Build Fast Views?
Enables building fast views.
Setting Values in Data Extract-Related Reference Codelists

Check the settings of the following reference codelists, described in Chapter 7, "Reference Codelists":

- DX_CONFIG
- DX_EXTENDED_ATTRIBUTES
- DX_INDEX_TABLESPACE
- DX_KEY_NAME
- DX_ROLES
- DX_VIEW_TABLESPACE

Creating Tablespaces for Data Extract Tables and Indexes

This section contains the following topics:

- Creating Tablespaces on page 6-5
- Entering Tablespace Names in Reference Codelists on page 6-5
- Creating Data Extract Access Accounts Using Local Tablespaces on page 6-6

Creating Tablespaces

You must create tablespaces to contain the tables and indexes required for data extract. Oracle recommends creating a separate tablespace for tables and for indexes, and for creating each tablespace:

- as locally managed
- with autoallocate on
- with automatic segment space management
- with a block size of 16 kb

Use the following command:

```
CREATE TABLESPACE dxtables DATAFILE '/u02/oracle/data/dxtables01.dbf' SIZE 500M EXTENT MANAGEMENT LOCAL AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO;
```

For further information, see the Oracle Database Administrator’s Guide.

Entering Tablespace Names in Reference Codelists

In order to enable users to create tables using the Data Extract View Builder, you must specify to which tablespace and tablespace index you want to add tables for each Study Access Account. You specify these tablespaces and index tablespaces in two installation reference codelists that store data extract tablespace information:

- DX_INDEX_TABLESPACE
- DX_VIEW_TABLESPACE
Creating Data Extract Access Accounts Using Local Tablespaces

You must create a Data Extract Access Account (under Conduct, navigate to Data Extract, then Study Access Accounts) and specify a tablespace for the account's tables and for its indexes. When the tablespace is defined as locally managed, the View Builder creates tables and indexes for that account as follows:

- The extract tables are created in a single step rather than the previous approach that created a temporary table and then, after determining the size of the table, a second permanent table. This feature capitalizes on the ability to use locally managed tablespaces with the AUTOALLOCATE feature that automatically sizes the table.

- The tables are created using the Oracle database table compression feature. Since data extract tables tend to have many repeated keys and values, this should result in significantly less space usage and more efficient data access.

- The indexes are created with leading key compression that results in significantly more compact and efficient indexes.

- Index statistics are now computed as the indexes are created, which results in faster statistics calculation.

- Repeated account maintenance of the ROLLSNAP account will take advantage of the more efficient space allocation method used in Locally Managed Tablespaces when it drops and recreates tables.

- Tables are created with the NOLOGGING attribute. This reduces the table creation time significantly by avoiding writing to the redo logs. The price of this option is that recovery from database failure using redo logs will not recreate extract tables that have not been otherwise backed up and restored. Since the tables can be recreated from the extract views at any time by rerunning account maintenance in FULL, this trade-off is usually acceptable. However, if you do not want to use the NOLOGGING behavior, you can override it by creating the locally managed tablespace with the FORCE LOGGING attribute.

Customizing Data Extract Views

Oracle Clinical ships with two scripts in the INSTALL directory that you can use to customize data extract views.

- rxcptdxvb.sql populates the data extract tables EXTRACT_KEYS and EXTRACT_MACROS and creates the standard key template.

- pop_vb_static_views.sql creates the standard view templates, which include views for responses and RDCMs.

For example, if you want to include the investigator's country in the data extract views, add a key for Investigator Name in rxcptdxvb.sql and modify the SELECT, FROM and WHERE clauses of the RDCMS_VIEW creation statement in pop_vb_static_views.sql. The scripts contain comments that show exactly what lines to add for this example.

Note: You must also remove invalid values from the reference codelists to prevent users from selecting them for study access accounts. See "DX_INDEX_TABLESPACE" on page 7-25 and "DX_VIEW_TABLESPACE" on page 7-26 for information.
Generating Data Extract Views

The gen_views utility performs the same operations as the Maintain Data Extract Views batch job within Oracle Clinical (under Conduct select Data Extract and then Data Extract Views) for all accounts in FULL maintenance mode.

Use the gen_views utility to regenerate views for all accounts in one or all studies. For example, if a change is made to a key template, all views based on that template in that study must be regenerated for all the study access accounts.

The general sequence for this task is:

- run opa_setup, which defines the RXC_TOOLS directory
- change to the RXC_TOOLS directory
- run gen_views.

Variables include:

- study enter either the name of a study or ALL for all studies
- sas_queue enter the name of the queue where SAS jobs execute or NULL (on Windows only)
- view_creation_mode valid values are: DATA_ONLY, COMBINED_VIEW, or SEPARATE_VIEW.

Operating system-specific instructions follow:

Running gen_views on UNIX Platforms

To run gen_views on a UNIX platform:

1. Log on to the server in your user account and set the environment:

<table>
<thead>
<tr>
<th>Shell</th>
<th>Command sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Shell</td>
<td>opa_setup $db_name $code_env</td>
</tr>
<tr>
<td>Bourne</td>
<td>p1 = $db_name</td>
</tr>
<tr>
<td></td>
<td>p2 = $code_env</td>
</tr>
<tr>
<td></td>
<td>. opa_setup</td>
</tr>
</tbody>
</table>

   where $db_name is a database instance name and $code_env is a code environment designation.

2. Change directories to $RXC_TOOLS.

3. Set the output directory:

<table>
<thead>
<tr>
<th>Shell</th>
<th>Command sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Shell</td>
<td>setenv RXC_LOG $usr_log_dir</td>
</tr>
<tr>
<td>Bourne</td>
<td>RXC_LOG=$usr_log_dir</td>
</tr>
<tr>
<td></td>
<td>export $code_env</td>
</tr>
</tbody>
</table>
Enabling the View Builder and Converting Views

4. Run the script. For example:

   % gen_views ALL UNIX DATA_ONLY

Running gen_views on Windows

To run gen_views on Windows:

1. Log on to the server using your local account.
2. In an MS-DOS window, set the server environment:

   set p1=db_name
   set p2=code_env
   opa_setup
   where db_name is a database instance name and code_env is a code environment designation.
3. Change directories to %RXC_TOOLS%
4. Set the output directory:

   set rxc_log=user_log_folder
5. Run the command file. For example:

   gen_views ALL NULL DATA_ONLY

Enabling the View Builder and Converting Views

The View Builder was an enhancement in Oracle Clinical Release 3.1. If you are still using pre-View Builder views, you can convert them to View Builder-style views by running two scripts, vb_pop_view and enable_vb, for each study.

Converting Views

Running vb_pop_view converts a study's existing old-style data extract views to the new view builder style. For example:

   cd $RXC_INSTALL
   sqlplus rxc/notrxc
   start vb_pop_view

   The script prompts you for the name of the study.

   Note: You can run vb_pop_view only once per study.

Enabling the View Builder in a Study

To make view builder-style the default for future views in a study, use the enable_vb.sql script. For example:

   cd $RXC_INSTALL
   sqlplus rxc/notrxc
   start enable_vb.sql
When you execute this script it performs the following actions:

- prompts you for the name of the study.
- updates the CLINICAL_STUDY_STATES table, setting VB_ENABLED to "Y".
- sets the default key template in CLINICAL_STUDY_STATES to STANDARD.

After running enable_vb, commit the changes to the database, then rerun view creation.

You can also enable view builder interactively for individual studies within the Oracle Clinical interface. However, using the command line may be more convenient.

**Controlling Access to Data Extract Views**

Access to individual views at the access account level can be controlled by granting access via roles. You can create your own company-specific roles so that appropriate choices appear in the list of values in the View Definition window.

Do the following:

1. Create as many database roles as you need; see "Creating Custom Database Roles" on page 2-6.
2. Add these roles to the DX_ROLES reference codelist. All the roles in that reference codelist appear in the list of values in the View Definition window.
3. Grant the database roles to users who need them; see "Granting Additional Database Roles to User Accounts" on page 1-15.
This section describes Oracle Clinical reference codelists. It is comprised of the following topics:

- **Overview** on page 7-1
- **Adding a Value to a Reference Codelist** on page 7-2
- **Modifying a Value in a Reference Codelist** on page 7-3
- **Running the Reference Codelists Report** on page 7-3
- **Local Reference Codelists** on page 7-7
- **Installation Reference Codelists** on page 7-19
- **Design Installation Reference Codelists** on page 7-31
- **System Reference Codelists** on page 7-31

**Overview**

Oracle Clinical uses reference codelists for a wide range of functionality. Certain codelists are used internally by the application, while others set the values the system present to users performing various tasks in the application. There are different types of reference codelists that are used by different subsystems. Some types of codelists are set during the installation of the application and others are set by users during various tasks.

Although the values of certain codelists are listed and described in this section, you can quickly view the values of any codelist by running the Reference Codelist report. Refer to the "Running the Reference Codelists Report" section on page 7-3 for instructions.

Depending on your assigned roles, you can perform the following types of reference codelist maintenance tasks:

- Add a new value to a local, installation, or design installation codelist.
- Activate or deactivate a codelist value for a local, installation, or design installation codelist.

**Accessing and Modifying Reference Codelists**

You may have authority to access only some of the types of codelist values. In any case, you cannot create new codelists; you can only add values to existing codelists.
You may modify only the following types of codelists:

- Installation codelists, such as the type codes for clinical planned events, DCIs, DCMs, and question groups.
- Design installation codelists, which are used to setup treatments and study designs, should be maintained by clinical (rather than system administration) personnel. The design installation codelists are actually a subset of the full set of installation codelists. If you choose, you can grant access to the design installation codelists for users to whom you would not grant access to the full set of installation codelists.
- Local codelists specific to your site within the company, such as the codelist for batch and print queue names.

System codelists are used internally by Oracle Clinical, and cannot be modified.

**Viewing Original Reference Codelist Settings**

To see the initial settings for a codelist shipped with Oracle Clinical, go into the Developer’s Toolkit, Maintain All Codelists, and query for a codelist. You can access all types of codelists through this window, including local, installation, design, and system codelists.

**Adding a Value to a Reference Codelist**

Follow this procedure to add values to an existing local, installation, design, or installation codelist:

1. Select **Admin**, then **Reference Codelists**. Choose one option to display the Reference Codelists window for one of these codelist types:
   - Local Codelists
   - Installation Codelists
   - Design Installation Codelists

2. To find the reference codelist you want, start a query, enter query criteria in one or more enterable fields to define the search, and then execute the query. Use the arrow keys to scroll to the codelist you want.

3. Click **Next Area** to move to the **Reference Codelist Values** block.

   If there are no records, the record lines are blank and you can go to step 5.

   If there are records, place focus in a field that is one row above the row you want to add. Select **Data**, then **Insert Record**.

4. Enter information about the new value in the following fields:
   - **Seq** – The order in which the value appears. Used for presentation and in reports.
   - **Short Value** – Shortened name of the value; used to fill in the application field when selected from a list of values.
   - **Long Value** – Complete name of the value; used if short value is truncated.
   - **Active** – For reference codelists that provide a list of values, only those entries whose Active checkbox is selected appear in the list.
Adding a Value to a Reference Codelist

- **Default** – For reference codelists that provide a list of values, the entry whose Default box is checked may be highlighted when the list appears.
- **Description** – Description of the value.

5. Click the **Save** button. The system adds the new value to the codelist.

Modifying a Value in a Reference Codelist

You cannot delete a value, but deactivating it (clearing the Active check box) means it will not appear in a list of values, and Oracle Clinical will not include it as an acceptable value during validation. Follow this procedure to modify an existing value in a reference codelist:

1. Perform steps 1 through 3 in “Adding a Value to a Reference Codelist” on page 7-2 to select a reference codelist, then place focus in the value record you want to change.

2. Make your changes. You cannot change the Short Value field. Changing the Active checkbox activates or deactivates the value. Only active values appear in the list of values for the relevant field and are used in field validation.

3. Click Save. Oracle Clinical modifies the changed codelist records in the database.

Make sure the Long Value of the entry SERVER_OS in the local reference codelist OCL_STATE is correct for your operating system—that is, set to Windows or UNIX.

---

**Note:** On UNIX systems, the Long Value of the server_name entry (also in OCL_STATE) must be in lowercase letters.

---

Running the Reference Codelists Report

Oracle Clinical includes a large number of reference codelists, many of which you can modify. In order to obtain the current values in any codelist, or to view the codelist values at any point in time, run the Reference Codelist report. This report summarizes all of the values in one or more reference codelists. You use the Report Submission window, which is depicted in Figure 7–1, to set the values of four parameters and specify the information you want to include in the report.
Adding a Value to a Reference Codelist

Figure 7–1 Report Submission Window for Reference Codelist Report

To run the Reference Codelists report:

1. Navigate to Admin, Admin Reports, and then Reference Codelists. The system opens that Report Submission window with a set of parameters specific to the Reference Codelists report.

2. In the list of parameters, set that values of the four parameters to setup the report you want to run. Only the Active Flag parameter value is mandatory.
   a. the Modification Date parameter allows you to limit the report to include codelists modified on or after a certain date; use the “DD-MON-YYYY” date format for this field
   b. use the Reference Codelist Name parameter to specify a codelist; the list of values allows you to select from the list of codelists
   c. the Reference Codelist Type parameter allows you to limit the report to certain types of codelists
   d. ensure that the Active Flag parameter is set to its default value of "Y".

3. Click Job Details. The system opens the Submission Details window.

4. In the Submission Details window:
   a. Set the Output Type, Output Format, and Printer (if applicable) fields to appropriate values.
   b. Ensure that the Mode of Execution and the Report Server fields are set correctly.

5. Click Submit Job.

Maintain/Query Reference Codelists Windows

When you query or maintain a codelist, the system displays the Reference Codelists window. The graphic in Figure 7–2, "Maintain Reference Codelists Window", depicts a typical "maintain" reference codelist window, which allows you to modify the values in a codelist.
**Access**

You access the reference codelist windows through the menu paths that are available under the **Admin, Reference Codelists** path. The available selections are:

- Local Codelists
- Qry Local Codelists
- Installation Codelists
- Qry Installation Codelists
- Design Installation Codelists
- Qry Design Installation Codelists
- Qry System Codelists

The selections that are prefixed "Qry" open the window in query mode, that is, you can view the codelists only. The other selections open the window in maintenance mode, in which the system allows you to modify the codelist. Note that you can only open the System Reference Codelists in query mode because that set of codelists cannot be modified.

**Components**

The reference codelist windows consist of two sections: an upper section that identifies the current codelist and provides information about it; and a lower section that lists the values associated with the current reference codelist. The components that
Adding a Value to a Reference Codelist

comprise the upper section of the window are listed and described in Table 7–1, "Components of the Reference Codelist section".

Table 7–1 Components of the Reference Codelist section

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Field</td>
<td>Displays the name of the current reference codelist; the field can also be used to query for a specific codelist.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box</td>
<td>Specifies if the codelist is active. This component cannot be updated.</td>
</tr>
<tr>
<td>Default</td>
<td>Field</td>
<td>Lists the default value, which is the value for which the Default checkbox in the Values section is selected.</td>
</tr>
<tr>
<td>Description</td>
<td>Field</td>
<td>A system-specified description of the current reference codelist</td>
</tr>
<tr>
<td>Type</td>
<td>Field</td>
<td>Displays the type of current codelist</td>
</tr>
<tr>
<td>Data Type</td>
<td>Field</td>
<td>Displays the data type for the values in the codelist</td>
</tr>
<tr>
<td>Max Short Len</td>
<td>Field</td>
<td>The maximum number of characters allowed for the Short Value of each reference codelist value</td>
</tr>
<tr>
<td>Max Long Len</td>
<td>Field</td>
<td>The maximum number of characters allowed for the Long Value of each reference codelist value</td>
</tr>
<tr>
<td>Application</td>
<td>Field</td>
<td>Displays the subsystem with which the codelist is associated</td>
</tr>
</tbody>
</table>

The lower portion of the window contains the values that are associated with the current reference codelist. In the maintenance windows, you use this section to modify the values. The components in this section are listed and described in Table 7–2, "Components of the Reference Codelist Values section".

Table 7–2 Components of the Reference Codelist Values section

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq</td>
<td>Field</td>
<td>Displays the sequence number of the value</td>
</tr>
<tr>
<td>Short Value</td>
<td>Field</td>
<td>Displays the short value for the value</td>
</tr>
<tr>
<td>Long Value</td>
<td>Field</td>
<td>Displays the long value for the value</td>
</tr>
<tr>
<td>Active</td>
<td>Check box</td>
<td>Specifies if the value is active, that is, if it is included when the system accesses the reference codelist</td>
</tr>
<tr>
<td>Default</td>
<td>Check box</td>
<td>Specifies if the value is the default value for the reference codelist</td>
</tr>
<tr>
<td>Description</td>
<td>Field</td>
<td>A text explanation of the value</td>
</tr>
<tr>
<td>Exit</td>
<td>Button</td>
<td>Closes the window</td>
</tr>
<tr>
<td>Save</td>
<td>Button</td>
<td>Commits all pending changes to the database</td>
</tr>
</tbody>
</table>
Usage
This section provides instructions for basic tasks you can use the reference codelists windows.

Query a reference codelist To search for a specific reference codelist:
1. With focus in the Name field, press the F7 key. This puts the system in query mode.
2. Type the name of the codelist, using the wildcard ("%") as necessary, to construct the search string.
3. Press the F8 key. The system runs the query and displays the first codelist returned in the Name field.

Navigate a list of codelists If more than one codelist is returned, there are several methods you can use to navigate to a specific entry:
- Use the Page Up/Page Down or the Up/Down arrow keys to move one entry up or down in the list.
- Use the Move, then Last Record to move to the bottom of the list.
- Use the Move, then First Record to move to the top of the list.

Add a value To add a value to the current reference codelist, you must be working in the Maintain Reference Codelist window.
1. Place focus in the row immediately above the location in which you want to add the value.
2. Select the Data, then Insert Record menu command. The system places a blank row below the current row.
3. Modify the fields for the row, as appropriate.

Local Reference Codelists
Local codelists control the behavior of some Oracle Clinical features in the selected database only. You can modify local codelists if your user role has one of the following schema: RXC_ADMIN, RXC_SUPER, or RXC_SUPER_NOGL.

Local codelists are listed here alphabetically:

BATCH QUEUE NAME
This codelist contains batch queue names to be used by the Parameterized Submission (PSUB) utility for this Oracle Clinical instance.

The reference codelist ships with a short value of RXC_BATCH_QUEUE, and a long value of a. This short value is in turn found in the OCL_JOB_PREF reference codelist, indicating that this is the default batch queue to be used by PSUB.

You can set a different default queue for a particular user by specifying any short value from the BATCH QUEUE NAME reference codelist when you create or modify the user’s account. When adding entries, the short value specifies a symbolic name for the queue, and the long value specifies a single character queue like a, d, e. b and c are excluded, as these are reserved.
You can modify the long value for the RXC_BATCH_QUEUE entry in one of 2 ways:
- change the value to another single-character queue name
- enter the value RXC_BATCH_QUEUE

If you choose the latter option, PSUB interprets this as an environment variable, whose value is set in the opa_settings file; see Appendix A, "Environment Variables and Registry Settings".

**DB_LINKS**

This codelist contains the names of database links for replication.

Replication is a "pulling" operation; that is, the database location requesting the data must initiate the action. Each database in the installation maintains its own local DB_LINKS reference codelist. There should be an entry in the Short Value field for each of the other database locations in the installation. The Long Value contains the name of the private database link to that database, owned by the Oracle user RXC_REP.

**DCF COMMENT TEXT**

This codelist contains values you can use as the initial text for the DCF comment field.

**DCF DEFAULT FOOTERS**

This codelist contains values you can use as the footer text to be inserted into the DCF Footer field. See "Defining DCF Headers and Footers" on page 3-32.

**DCF DEFAULT HEADERS**

This codelist contains values you can use as the header text to be inserted into the DCF Header field. See "Defining DCF Headers and Footers" on page 3-32.

**DCF REPORT LABELS**

This codelist contains user-configurable labels for the DCF Report.

**DISC COLS**

This codelist contains the variables that can be specified for MCU CDS Discrepancy criteria; see "Customizing Mass Changes Local Codelists" on page 4-1.

**DISC_FLEX1 and DISC_FLEX2**

See "Customizing Flexfields" on page 3-9 for information.

**DISC_FLEX1_VALUES and DISC_FLEX2_VALUES**

See "Customizing Flexfields" on page 3-9 for information.

**FLD RXCMCMCD RDCI DELETE**

This codelist contains the display and order of RDCI Delete Candidate Data Set fields; see "Customizing Mass Changes Local Codelists" on page 4-1.
FLD RXCMCMCD RDCI KEY
This codelist contains the display and order of RDCI Key Change Candidate Data Set fields; see "Customizing Mass Changes Local Codelists" on page 4-1.

FLD RXCMCMCD RDCM KEY
This codelist contains the display and order of RDCM Key Change Candidate Data Set fields; see "Customizing Mass Changes Local Codelists" on page 4-1.

FLD RXCMCMCD RESPONSE
This codelist contains the display and order of response Candidate Data Set fields; see "Customizing Mass Changes Local Codelists" on page 4-1.

IND DISC COLS
This codelist contains variables that you can specify for MCU CDS Ind discrepancy criteria; see "Customizing Mass Changes Local Codelists" on page 4-1.

MAN DISC COLS
This codelist contains variables that you can specify for MCU CDS manual discrepancy criteria; see "Customizing Mass Changes Local Codelists" on page 4-1.

MANHD DISC COLS
This codelist contains variables that you can specify for MCU CDS manual header discrepancy criteria; see "Customizing Mass Changes Local Codelists" on page 4-1.

MC CDS SORT ORDER
This codelist contains the sort order of CDS fields; see "Customizing Mass Changes Local Codelists" on page 4-1 for information.

MC COLUMNS
This codelist contains variables that you can specify for MCU CDS criteria; see "Customizing Mass Changes Local Codelists" on page 4-1.

MULTI DISC COLS
This codelist contains variables that you can specify for MCU CDS multivariate discrepancy criteria; see "Customizing Mass Changes Local Codelists" on page 4-1.

NLS_CONFIG
This codelist contains settings that control behavior in Oracle Clinical with NLS Option. See the Oracle Clinical with NLS Option User’s Guide for details.

OCL MC PREFS
This codelist enables you to set preferences for Mass Changes.
OCL_DE_CONFIG

This codelist controls Data Entry configuration settings. The entries for values are listed and described in Table 7–3.

Note: The settings with sequence numbers 1-9, 12-14, and 17 are also updateable in the Maintain Installation Configuration window under Admin, DE Admin, DE Config Settings; see “Define Data Entry Configuration Settings” on page 5-1.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2ND PASS ALERT</td>
<td>Y</td>
<td>Determines if the alert for a 2nd Pass comparison failure is enabled</td>
</tr>
<tr>
<td>2</td>
<td>DISC IN BROWSE</td>
<td>Y</td>
<td>Determines if the user is allowed to initiate a manual discrepancy when working in browse mode</td>
</tr>
<tr>
<td>3</td>
<td>DISC RES IN DE</td>
<td>Y</td>
<td>Determines if the user is allowed to resolve discrepancies during data entry.</td>
</tr>
<tr>
<td>4</td>
<td>PRIV UPDATE</td>
<td>N</td>
<td>Determines if the privileged update is enabled at the database level.</td>
</tr>
<tr>
<td>5</td>
<td>THESAURUS list of values</td>
<td>Y</td>
<td>Determines if the DVG, which populates the list of values, for thesaurus questions is enabled.</td>
</tr>
<tr>
<td>6</td>
<td>UNIVAR ALERT</td>
<td>Y</td>
<td>Determines if the univariate validation alert, which prompts the user to a validation error during data entry, is enabled.</td>
</tr>
<tr>
<td>7</td>
<td>USE DCI BOOK</td>
<td>N</td>
<td>Determines if the system initiates a data entry session using a DCI book.</td>
</tr>
<tr>
<td>8</td>
<td>UNENROLL ALERT</td>
<td>Y</td>
<td>Determines if the system alerts the user to a patient that has been unenrolled.</td>
</tr>
<tr>
<td>9</td>
<td>P2 NOT BY P1</td>
<td>N</td>
<td>Determines if the system prevents a Pass 1 data entry operator from performing Pass 2 data entry.</td>
</tr>
<tr>
<td>10</td>
<td>OCL THES DISC</td>
<td>N</td>
<td>Determines if system alerts the user to OLC Thesaurus discrepancies during data entry.</td>
</tr>
<tr>
<td>11</td>
<td>OCL THES list of values</td>
<td>N</td>
<td>Determines if the lists of values for OCL Thesaurus questions are enabled.</td>
</tr>
<tr>
<td>12</td>
<td>BROWSE ACC ONLY</td>
<td>N</td>
<td>Determines if data in accessible documents is browse only.</td>
</tr>
<tr>
<td>13</td>
<td>DEF PAGE HEIGHT</td>
<td>22</td>
<td>Sets the default height of the DCM data entry page in points, pixels, inches, or centimeters, depending on the unit of measure selected in the Maintain Installation Configuration window under Admin, DE Admin, DE Config Settings; see “Define Data Entry Configuration Settings” on page 5-1. Applies only to Oracle Clinical and to RDC Classic.</td>
</tr>
<tr>
<td>14</td>
<td>DEF PAGE WIDTH</td>
<td>78</td>
<td>Sets the default height of the DCM data entry page in points, pixels, inches, or centimeters, depending on the unit of measure selected in the Maintain Installation Configuration window under Admin, DE Admin, DE Config Settings; see “Define Data Entry Configuration Settings” on page 5-1. Applies only to Oracle Clinical and to RDC Classic.</td>
</tr>
<tr>
<td>15</td>
<td>P2 ALWAYS ALERT</td>
<td>Y</td>
<td>Determines if the system alerts the user whenever Pass 2 data differs from the corresponding Pass 1 data.</td>
</tr>
<tr>
<td>16</td>
<td>AUTO SEQ DFLT</td>
<td>Y</td>
<td>Sets the default auto sequence behavior; when a user presses tab in the last field of a DCM/CRF Section, the system opens the next DCM for data entry.</td>
</tr>
</tbody>
</table>
### Table 7–3 (Cont.) Values for the OCL_DE_CONFIG Reference Codelist

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>DCI DATE REQ</td>
<td>Y</td>
<td>Sets the default as to whether the DCI date is required in the Log-in form.</td>
</tr>
<tr>
<td>18</td>
<td>DVGSEQOVERALPHA</td>
<td>N</td>
<td>This setting is used only when a question has both an internal and an alpha DVG assigned. Enter By Sequence is enabled for the internal DVG, and both the internal and the alpha DVG have values with the same sequence number. When the user enters a sequence number that exists in both DVGs, the system records the internal DVG value if this reference codelist value is set to Y. If it is set to N, the system records the alpha DVG value with the same sequence number.</td>
</tr>
<tr>
<td>19</td>
<td>DVG List of Values STYLE</td>
<td>SEQ</td>
<td>This value determines how the system displays the list of values for DVG and Alpha DVG questions.</td>
</tr>
<tr>
<td>20</td>
<td>COMMCHGREQ_ SEQ</td>
<td>Y</td>
<td>Determines if a change reason is required for updates to RDCI comments in accessible documents.</td>
</tr>
<tr>
<td>21</td>
<td>JUSTENTERP1TIME</td>
<td>0</td>
<td>Sets the time period (in minutes) during which a Pass 1 Complete RDCI/RDCM can be modified in Pass 1 mode.</td>
</tr>
<tr>
<td>22</td>
<td>JUSTENTERP2TIME</td>
<td>0</td>
<td>Sets the time period (in minutes) during which a Pass 2 Complete RDCI/RDCM can be modified in Pass 2 mode.</td>
</tr>
<tr>
<td>23</td>
<td>SEQUENCEBUFFER</td>
<td>1000000</td>
<td>The sequence buffer that is used to prevent the sequence from reaching its maximum value. At this setting, when a sequence number is within 1,000,000 of 2,147,483,647, the system displays a warning message when a user attempts to use a relevant subsystem and the system exits the current screen.</td>
</tr>
<tr>
<td>24</td>
<td>RSTRCT LCKD CRF</td>
<td>N</td>
<td>Determines if some actions are permitted on a locked CRF. Applies to RDC Onsite only.</td>
</tr>
</tbody>
</table>

### DVG List of Values STYLE

This value determines how the system displays the list of values for DVG and Alpha DVG questions. The options for this value are SEQ and SHORT.

**Note:** If "Enter by Sequence" is not selected in the DVG definition, this reference codelist setting (SEQ or SHORT) has no effect. The list of values does not display the sequence number, sorts records by the DVG value, and displays these columns in this order:

1. Short Value, which is titled "DVG Value" in the list of values
2. Alpha Column, which contains "A" if it is an Alpha DVG value or is null for all others
3. Long Value, which is titled "Description" in the list of values.

**SEQ** If the Long Value is set to "SEQ" and "Enter by Sequence" is selected in the DVG definition, the columns in the list of values are displayed in this order:

1. Display SN (prefixed with "A" if it is an Alpha DVG value)
2. Short Value, which is titled, "DVG value" in the list of values
3. Long Value, which is titled "Description" in the list of values.

The records are sorted by DVG sequence number.
**SHORT** If the Long Value is set to "SHORT" and "Enter by Sequence" is selected in the DVG definition, the columns in the list of values are displayed in this order:

1. Short Value, which is titled "DVG Value" in the list of values
2. Display SN (prefixed with "A" if it is an Alpha DVG value)

**Note:** The sequence number is displayed only if there is an alpha DVG associated with the question.

3. Long Value, which is title "Description" in the list of values.

The records are sorted by DVG sequence number.

**COMMCHGREAS_REQ**
This value determines if the system requires a change reason when the RDCI comment is updated and the document is internally accessible. The options for this value are "Y" and "N". The default value is "Y", which causes the system to require a change reason for an update to the RDCI comment.

**JUSTENTERP1TIME**
This value defines time period (in minutes) during which a user can modify or query a previously Pass 1 Complete RDCI/RDCM in Pass 1 mode.

The initial value of this entry is "0".

**JUSTENTERP2TIME**
This value defines the time period (in minutes) during which a user can modify/query a previously Pass 2 Complete RDCI/RDCM in Pass 2 mode.

The initial value of this entry is "0".

**RSTRCT LCKD CRF**
By default, RDC Onsite restricts access to locked CRFs. You can use the RSTRCT LCKD CRF setting in the OCL_DE_CONFIG local reference codelist to allow some users to take actions on locked CRFs.

- **Y** — Specifies that users cannot update discrepancies for a locked CRF, verify a locked CRF, or approve a locked CRF unless the CRF is specifically unlocked for them.
- **N** — Specifies that any user with UPD_DISCREP privileges can work on discrepancies in a locked CRF, any user with VERIFY privileges can verify a locked CRF, and any user with APPROVE privileges can approve a locked CRF.

**OCL_DE_PREFS**
This codelist enables you to set the default data entry preferences for this instance. The default entries are displayed and described in Table 7–4, "Values for the OCL_DE_PREF reference codelist". You can also set almost all of these values in the Maintain Installation Preferences under Admin, DE Admin, DE User Prefs. The exceptions, which you can set only here, are: AUTO NEXT FORM and AUTO SEQ DEFLT.
**Table 7–4  Values for the OCL_DE_PREF reference codelist**

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AUTO SKIP</td>
<td>Y</td>
<td>Determines if auto skip is enabled</td>
</tr>
<tr>
<td>2</td>
<td>AUTO FILL</td>
<td>Y</td>
<td>Determines if auto fill is enabled</td>
</tr>
<tr>
<td>3</td>
<td>UNIVAR BEEP</td>
<td>Y</td>
<td>Determines if the system sounds a beep when a recorded response value generates a validation error</td>
</tr>
<tr>
<td>4</td>
<td>COMPARISON FAILURE</td>
<td>Y</td>
<td>Determines if the system sounds a beep when a response value generates a comparison failure.</td>
</tr>
<tr>
<td>5</td>
<td>END FORM BEEP</td>
<td>Y</td>
<td>Determines if the system sounds a beep when the user navigates from the last field in a form.</td>
</tr>
<tr>
<td>6</td>
<td>DATE ENTRY FMT</td>
<td>US</td>
<td>Determines the default format for dates that the system assumes during data entry</td>
</tr>
<tr>
<td>7</td>
<td>DATE DISPLAY FMT</td>
<td>STANDARD</td>
<td>Determines the format the system uses to present dates in the display</td>
</tr>
<tr>
<td>8</td>
<td>RDCI ORDER</td>
<td>PATIENT</td>
<td>The “order-by” the system uses for RDCI queries.</td>
</tr>
<tr>
<td>9</td>
<td>AUTO NEXT FORM</td>
<td>Y</td>
<td>Determines if the system automatically displays the next data entry form in the sequence.</td>
</tr>
<tr>
<td>10</td>
<td>AUTO SEQ DEFLT</td>
<td>Y</td>
<td>Determines if auto-sequence is set by default.</td>
</tr>
</tbody>
</table>

**OCL_JOB_PREF**

This codelist sets default values in the Oracle Clinical PSUB job window.

Oracle Clinical ships default system-wide values in the local reference codelist OCL_JOB_PREF. Each entry refers to a specific row in another reference codelist, which must be updated with correct values for your installation. The long value in this reference codelist is populated from the short value in the other reference codelist. Table 7–5 lists the defaults you can set.

**Table 7–5  Entries in the OCL_JOB_PREF Reference Codelist**

<table>
<thead>
<tr>
<th>Short Value</th>
<th>Description</th>
<th>Refers to reference codelist</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSUB_PRINTER</td>
<td>Default PSUB Printer</td>
<td>PRINT QUEUE NAME</td>
</tr>
<tr>
<td>DFLT_SUBSCH_RS</td>
<td>Default for PSUB Scheduling, generating DCI Forms, and previewing</td>
<td>REPORT_SERVER</td>
</tr>
<tr>
<td>DFLT_RS_PRINTER</td>
<td>Default Reports Server Printer</td>
<td>PRINT QUEUE NAME</td>
</tr>
</tbody>
</table>

For example, to set a printer as the default for reports at an installation:

1. Navigate to Admin, then Reference Codelists, and Local Codelists.
2. Insert a record in the PRINT QUEUE NAME codelist:
   Short Value – Name of the printer, for example, boston09
   Long Value – Printer specification, for example, \ocldsn1\boston09
   Description – Information about the printer that may be helpful to the end user

   (Note that all three values display in the Submission Details screens.)

3. In the OCL_JOB_PREF codelist, update the Long Value of the DFLT_RS_PRINTER entry with the Short Value of the printer from Step 3.

OCL MC PREFS

This codelist’s single value, MAX_CDS_RECORDS, sets the default maximum number of records to change using the Mass Changes utility. Its initial value is 1000; see "Customizing Mass Changes Local Codelists" on page 4-1.

OCL_MENU_ACCESS

This codelist contains settings that support configuration of menu options.

Table 7–6 OCL_MENU_ACCESS Default Values

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PRPJ</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>ORUN</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>REGION</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>STUDY</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>FACTOR</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>STRATA</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>ACSU</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>DRUG</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>REGIMEN</td>
<td>Y</td>
</tr>
<tr>
<td>10</td>
<td>SITES</td>
<td>Y</td>
</tr>
<tr>
<td>11</td>
<td>INVEST</td>
<td>Y</td>
</tr>
<tr>
<td>12</td>
<td>PATTEN</td>
<td>Y</td>
</tr>
</tbody>
</table>

OCL_STATE

OCL_STATE tracks information about the database, such as its name, operating system, and RDBMS version, which Oracle Clinical requires to complete certain processes. Some of the short values are described in more detail following the table.

Table 7–7 Values for the OCL_STATE Reference Codelist

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DB_NAME</td>
<td>name of database</td>
<td>Name of the database</td>
</tr>
<tr>
<td>2</td>
<td>SERVER_OS</td>
<td>Server operating</td>
<td>system</td>
</tr>
<tr>
<td>3</td>
<td>LOCATION_CODE</td>
<td></td>
<td>Client name for the current location</td>
</tr>
<tr>
<td>Seq</td>
<td>Short Value</td>
<td>Long Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>PRINTER_TYPE</td>
<td>Default type of printer used - ASCII</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DB_VERSION</td>
<td>The database version, this is used by Data Extract to determine optimization.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>INVOKE_IMAGE</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BDL_R_SE</td>
<td>Specifies the rollback segment for Batch Data Load processes. Set to active and enter a long value of Y if you have created a database rollback segment for use in batch data load.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DFLT_PAGE_TRACK</td>
<td>N</td>
<td>Specifies the default value of the “Page Tracking Enabled” study level setting.</td>
</tr>
<tr>
<td>9</td>
<td>FILE_VIEWER</td>
<td>Y</td>
<td>Used by PSUB to view the log and output files</td>
</tr>
<tr>
<td>10</td>
<td>SERVER_NAME</td>
<td>Y</td>
<td>Used by PSUB to open FTP sessions</td>
</tr>
<tr>
<td>11</td>
<td>DB_BLOCK_SIZE</td>
<td>The database block size from init.ora</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>TEMP_TABLES</td>
<td>Temporary Tablespace for use in new account creation activities</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>DM_PROMPT</td>
<td>Prompt for the DM window.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>INVOKE_WORKFLOW</td>
<td>N</td>
<td>Used by the Login form to determine if a workflow system can be used.</td>
</tr>
<tr>
<td>15</td>
<td>USR_SAVE_OCPASS</td>
<td>Y</td>
<td>Determines if the user can save the OS password in the database; see &quot;UNIX Passwords” on page 1-21.</td>
</tr>
<tr>
<td>16</td>
<td>PROC_LAB_ALIAS</td>
<td>LAB</td>
<td>The name of the lab variable for the standard Procedure question group prompt.</td>
</tr>
<tr>
<td>17</td>
<td>DISC_DCM_PROMPT</td>
<td>QUESTION_NAME</td>
<td>The single-record mode of the Maintain Discrepancy Database form's Characteristics panel has a button that toggles the three long value settings. Choose the value to display by default: question_name (displays the name of the question associated with the discrepancy),</td>
</tr>
<tr>
<td>18</td>
<td>DCF_TEXT_SYNC</td>
<td>N</td>
<td>Edits comments and resolutions text DCFs propagates to disc mgmt</td>
</tr>
<tr>
<td>19</td>
<td>DISC_AUTO_HDFT</td>
<td>Y</td>
<td>Auto Generation of Header/Footer Text</td>
</tr>
<tr>
<td>20</td>
<td>DISC_AUTO_CRFPG</td>
<td>Y</td>
<td>Auto Generation of CRF Page Number</td>
</tr>
<tr>
<td>21</td>
<td>BOOK_USAGE</td>
<td>IGNORE</td>
<td>(Applies only to RDC Classic) If set to IGNORE (the default) the system allows the user to use any book for the patient, regardless of the patient's book assignment. If set to PROMPT, the system allows the use of any book but gives a warning. If set to ENFORCE, the system does not allow using any book other than the one to which the patient is assigned.</td>
</tr>
<tr>
<td>23</td>
<td>BOOK_CHANGE</td>
<td>ALLOW</td>
<td>Change of DCI Book for patient: ALLOW, DISALLOW, or DISALLOW IF DATA</td>
</tr>
</tbody>
</table>
SERVER_NAME
Set the SERVER_NAME value in this codelist to the database/PSUB server.

LOCATION_CODE
For replication, the specific piece of information required is the LOCATION_CODE value, which is the client name for the current location. The system collects and stores this value during database creation.

DB_BLOCK_SIZE
This entry is used by the DX table code to obtain the block size, which is used for space calculations. It is used for the dictionary-managed tablespace algorithm.

TEMP_TABLES
This entry specifies the name of the temporary tablespace the system users when creating DX study_access_accounts.

DM_PROMPT
This entry is used by the Discrepancy Management form as the prompt to use for Study site.

DISC_DCM_PROMPT
Specifies the source of the question in a discrepancy. The options for the Long Value are:
- DEFAULT_PROMPT
- QUESTION_NAME
- SAS_LABEL

PROC_LAB_ALIAS
This entry is used during procedure generation to determine the alias of the standard RDCM Question LAB in the Procedure Question Group declarations. The system sets

---

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>BOOK_ASSIGN</td>
<td>N</td>
<td>Assign DCI book to patient on first data entry: Y or N</td>
</tr>
<tr>
<td>25</td>
<td>TMS_FAIL_BV_ACT</td>
<td>FAIL</td>
<td>Determines whether Batch Validation continues when TMS derivation fails during Batch Validation: WARN or FAIL.</td>
</tr>
<tr>
<td>26</td>
<td>DB_HOST</td>
<td></td>
<td>Host name for the database, as entered in tnsnames.ora file</td>
</tr>
<tr>
<td>27</td>
<td>DB_PORT</td>
<td>1532</td>
<td>Port number for the database as entered in tnsnames.ora file</td>
</tr>
<tr>
<td>28</td>
<td>DMGR RDC ACCESS</td>
<td>YES</td>
<td>&quot;Granting Automatic Access in RDC to Studies Granted in Oracle Clinical&quot; on page 1-13</td>
</tr>
<tr>
<td>29</td>
<td>UPD_FV_INCREM</td>
<td>N</td>
<td>Y or N - Y means do only incremental PDR Templates and HTML Forms gen</td>
</tr>
</tbody>
</table>

---

Table 7–7 (Cont.) Values for the OCL_STATE Reference Codelist
this to a default value of "LAB". This default value for the alias will conflict with the Question QGalias.LAB if you have a study question named "LAB".

If there is a conflict with the study question named LAB, you should redefine PROC_LAB_ALIAS to a different name, such as, "RDCM_LAB", to avoid errors during procedure generation. In this case, any references to the standard Received DCM Lab question must be changed to RDCM_LAB, or a generation error will occur. When you do this, procedure references to QGalias.LAB will relate to the study question LAB and references to QGalias.RDCM_LAB will relate to the standard Received DCM Lab.

**BOOK.Assign**

If set to Y, the system automatically assigns the DCI Book defined as the default for the study (in DCI Books or Enhanced DCI Books, under Definition) to each patient when data is first entered for the patient.

**TMS_FAIL_BV_ACT**

Use this entry to indicate whether batch validation should continue or stop after a TMS-related error is encountered. The two values are 'WARN' and 'FAIL,' which is the default value. It causes batch validation to fail when there is a TMS failure; for example, when either an invalid dictionary or an invalid domain is defined. Irrespective of the value for TMS_FAIL_BV_ACT, if a TMS-related error occurs, the batch validation status is always failure and batch validation completes the process with a TMS warning.

**UPD_FV_INCREM**

This setting pertains to the upgrade utility provided with Oracle Clinical/RDC 4.5.3 and above that allows you to migrate from PDF to HTML data entry forms. By default, the Upgrade utility generates the HTML data entry forms and the PDR templates for all DCI Form Versions in the study that have PDF data entry forms generated.

There may be times when you have problems with the form or template generation. For example, perhaps some images used in the form layout are no longer in the correct location so the Upgrade utility does not generate a few of the DCI Form versions. For such cases, you can temporarily change the utility to run in incremental mode. In incremental mode, the Upgrade utility only creates the HTML data entry forms and the PDR templates if they do not already exist.

Set the long value to Y or N:

- **Y** — Forces the Upgrade utility to run in incremental mode. In incremental mode, the utility will not regenerate HTML data entry forms and PDR templates for form versions that already exist.
- **N** (default) — Forces the Upgrade utility to generate all form versions.

This setting applies only to running the Upgrade utility for existing DCI Form versions.

**Note:** Running the Upgrade utility in incremental mode is for resolving problems. Oracle recommends that you do not continue to run the utility in incremental mode. Be sure to change the UPD_FV_INCREM value back to N after you generate the forms you need.
PRINT QUEUE NAME

This codelist populates the list of printers that you can use when you submit a batch job or a reports job and when you set up a user account.

Oracle Clinical users can select a printer name from a list of values when they submit a job. You specify that list by defining entries in this local reference codelist. Both the long and short values appear in the list of values.

The short value is a code or abbreviation for the printer. The long value of the printer name is the printer specification. You can use either an absolute path or an environment variable for the long value. For example:

- `\opaprtsrv\walprt9` where walprt9 is the printer name
- `%RXC_PRINTER%` where RXC_PRINTER is the environment variable

The Default setting in the reference codelist has no effect. The OCL_JOB_PREF reference codelist determines the default printer. You can override the default when you set up a user account with any of the values in the PRINT QUEUE NAME reference codelist, and a user can override his or her own default when he or she submits a job.

PUBLIC_DB_LINKS

This codelist provides a list of database links for replication; it is used by RXA_DES.

Each database in the installation maintains its own PUBLIC_DB_LINKS local reference codelist. There should be an entry in the Short Value field for each of the other database locations in the installation. The Long Value contains the name of the public database link to that database.

REPORT_SERVER

This codelist defines the list of reports servers from which the user can select when setting up to run a report.

SAS_QUEUE

This codelist specifies the value(s) in the BATCH QUEUE NAME codelist that is required to run SAS.

SQL FUNCTIONS

This codelist provides a list SQL functions that you can then use when building queries.

TMS_OPTIONS

This local reference codelist contains TMS-specific options. It is populated only if TMS is installed in the Oracle Clinical database. When TMS is installed, currently the only option defined is FIRST_REVIEW. If the Long Value for this option is set to Y, and the Review Before TMS flag is also set to Y in the question set, then the first review for a thesaurus omission happens in the Oracle Clinical discrepancy management system, rather than in TMS. For more information on the interaction between the Oracle Clinical discrepancy management system and TMS, see the "Defining a question set" topic in the TMS User's Guide.
UNI DISC COLS
This codelist contains variables that you can specify for Mass Change Utility CDS Univariate Discrepancy criteria.

Installation Reference Codelists
Installation codelists control Oracle Clinical behavior installation-wide; that is, across an installation of multiple Oracle Clinical databases. They are listed here alphabetically:

APPLICATION AREA CODE
This codelist contains settings for areas on the body where the medication can be given.

APPLICATION SYSTEM NAME
This codelist contains a list of application systems used by Oracle Clinical.

APPROVE VERIFY RETAIN CODE
Enter one or more reasons to supply if CRF approvals or verifications are retained during DCI Form version migration.

You can limit the options available in a particular study in the DCI Form Study Settings window. If User Override to Reverse Approvals? or User Override to Reverse Verifications? is set to N in the DCI Form Local Database Settings window, the user running the migration cannot change the default value.

1. In the row with the short value DFLT_RETAIN_R, enter the text you want to appear as the initial default value as the long value.

2. If you want to allow the user to select other reasons for retaining approvals, enter an appropriate short and long value for each reason. Both values appear in the list of values.

3. Set each value to Active.

APPROVE VERIFY REVERSE CODE
Enter one or more reasons to supply if CRF approvals or verifications are reversed during DCI Form version migration.

You can limit the options available in a particular study in the DCI Form Study Settings window. If User Override to Reverse Approvals? or User Override to Reverse Verifications? is set to N in the DCI Form Local Database Settings window, the user running the migration cannot change the default value.

1. In the row with the short value DFLT_REVERSE_R, enter the text you want to appear as the initial default value as the long value.

2. If you want to allow the user to select other reasons for retaining approvals, enter an appropriate short and long value for each reason. Both values appear in the list of values.

3. Set each value to Active.
BLIND TYPE CODE
This codelist contains a list of types of blinding that are available in a study.

CLINICAL PHASE
This codelist list the clinical phases that are available in the design of a study.

COUNTRIES
This codelist contains the list of valid countries.

CRF PAGE NUMBERING SCHEME
This codelist contains the list of valid page tracking statuses that is used by the CRF Page Tracking feature.

CRF PAGE STATUS CODES
This codelist contains the list of valid CRF page statuses that is used in Maintain Page Status Tracking form.

CRF PAGE STATUS QUERY
This codelist contains the list of valid statuses for CRF page types that is used in Log-In Query window.

DATA CHANGE REASON TYPE CODE
This codelist contains the list of valid data change reason type codes for data update. The system uses it to provide a default value and to validate user-supplied response change reasons for Oracle Clinical data entry and mass changes (response changes), RDC, and DCAPI.

The default value is presented initially in each user session. If the user selects a different change reason, the system presents the user’s last selection as the default. Subsequent changes use the value that was selected last.

To set the default data change reason for a user group, put the group name in the Long Value field of that reason code (Short Value column). You can also put a comma-separated list of roles in a single field.

To prevent the reason from appearing at all in RDC Onsite HTML, enter NOTRDC as the long value.

Note: Changing the Long Value of a DATA CHANGE REASON TYPE CODE entry has no effect on data update in Oracle Clinical DE forms. This feature is affects RDC data entry only.

You can customize this codelist. However, the following values are reserved by the system and cannot be used as custom values: 'PASS1', 'PASS2', 'BATCH', 'UPDATE', 'REMOVED', 'TRANSLATION', 'KEY CHANGE', 'BROWSE', 'RECONCILIATION'.
Table 7–8  Values for the DATA CHANGE REASON TYPE CODE Reference Codelist

<table>
<thead>
<tr>
<th>Short value</th>
<th>Long value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRA CORR</td>
<td>CRA Correction</td>
<td></td>
</tr>
<tr>
<td>DATA ENTRY ERR</td>
<td>Error during data entry</td>
<td></td>
</tr>
<tr>
<td>BATCH</td>
<td>Batch-loaded data</td>
<td></td>
</tr>
<tr>
<td>CRA CORR-INV</td>
<td>CRA corrected, after consulting with the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Investigator</td>
<td></td>
</tr>
<tr>
<td>CRA CORR-SRC</td>
<td>CRA correction, after consulting the source data</td>
<td></td>
</tr>
<tr>
<td>INV CORR</td>
<td>Investigator correction</td>
<td></td>
</tr>
<tr>
<td>STUDY ASSUM</td>
<td>Study assumption</td>
<td></td>
</tr>
<tr>
<td>THES CLARIF</td>
<td>Thesaurus clarification</td>
<td></td>
</tr>
<tr>
<td>ANALYSIS CORR</td>
<td>Analysis correction</td>
<td></td>
</tr>
<tr>
<td>REMOVED</td>
<td>RDC removed</td>
<td></td>
</tr>
<tr>
<td>VAL STATUS CHG</td>
<td>Validation status changed</td>
<td></td>
</tr>
<tr>
<td>DATA ENTRY MODE</td>
<td>Data entry mode</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Set the LONG VALUE of one the reference codelist values in this list or in DATA CHANGE REASON2 TYPE CODE, to "DCAPIDEL" and another to "DCAPIINV" (comma-separated, if there are other values in the LONG VALUE).

**DATA CHANGE REASON2 TYPE CODE**

This codelist is used to provide a default value and for validation in DCAPI. It works in conjunction with the DATA CHANGE REASON TYPE CODE. It is not used by Oracle Clinical or RDC, but provides a method for you to provide an additional list of values of change reasons that are used by your external applications.

There are no default values for this codelist. Although you can customize this codelist, note that the following set of values are reserved by the system and cannot be used as custom values: 'PASS1', 'PASS2', 'BATCH', 'UPDATE', 'REMOVED', 'TRANSLATION', 'KEY CHANGE', 'BROWSE', 'RECONCILIATION'.

**DCF LOCK CONDITIONS**

This codelist contains statuses that refer to particular actions that should be limited; see 'DCF LOCK CONDITIONS' on page 3-29 for information.

**DCF OPTIONAL_STATUS_CODES**

This codelist contains optional status codes for DCFs; see "DCF OPTIONAL STATUS CODES" on page 3-29.

**DCF STATUS CODES**

This codelist contains status codes for DCFs; see "DCF STATUS CODES" on page 3-29.
DCIF CHECKBOX SIZE

This codelist controls the allowable sizes for checkboxes in DCI Forms. These values are available in the DCI Form Local Database Settings window; see "Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report" on page 5-10.

The values refer to point sizes. The values are:

- 10
- 12
- 15
- 20

DCIF FONT TYPESIZE

This codelist contains the list of valid font type sizes that can be used in DCI Form graphic layouts in prompts and fields; see "Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report" on page 5-10.

The default values refer to point sizes. The values are:

- 8
- 9
- 10
- 11
- 12
- 14

DCIF PAGE DEFINITION

This codelist contains the list of page definitions that can be used by the DCI form; see "Setting DCI Form Default Values for RDC Data Entry and the Patient Data Report" on page 5-10. The four values that are installed are:

- US Portrait
- US Landscape
- A4 Portrait
- A4 Landscape

The long value specifies the height width, and binding offset. With 0 Binding offset, the form layout template appears in the middle of the page. If you specify a non-zero (positive or negative) offset, the FLT moves left/right (for portrait) or up/down (for landscape).

DCM DCI QG TYPE CODE

This codelist contains the values you use to categorize DCMs, DCIs, and Question Groups. The values appear in the lists of values in the DCI Type field in the Maintain DCIs window, the Type field in the Maintain DCMs window, and the QG Type field of the Maintain Question Groups window.

DISCREPANCY ACTIONS ROLE Codelists

See "Configuring the Actions Allowed on Discrepancies" on page 3-23 for information.
DISCREPANCY MESSAGES
This codelist's function has been replaced by the Standard Text Maintenance form under Admin, Discrepancy Mgmt Maintenance. See "Reason Codes and Descriptions for Univariate Discrepancies" on page 3-12 for information.

DISCREPANCY RESOLU TYPE CODE
See "Defining Resolution Reasons for Discrepancies" on page 3-16 for information.

DISCREPANCY REV STATUS CODE
See "Defining the Possible Review Statuses for Discrepancies" on page 3-14 for information.

DISCREPANCY STATUS ROLE Codelists
See "Configuring Discrepancy Display by User Role" on page 3-18 for information.

DISCREPANCY NO OTHER UPDATE
See "Preventing Update to OTHER Discrepancies" on page 3-25 for information.

DISCRETE VAL GRP TYPE CODE
This codelist lists the severity codes for DVGs.

DISC_STDST_VALUES
This codelist contains discrepancy Study Site Values.

DOSE FORM TYPE CODE
This codelist contains Dosage Form Types.

DOSE FREQUENCY TYPE CODE
This codelist contains frequencies of dose administration.

DX_CONFIG
This codelist contains default settings for data extract.
The DX installation Configuration window includes exactly the same settings. When you change a setting here, the change is reflected there. When you change a setting there, the change is reflected here. For additional explanation of each setting, see "Configuring Default Installation Data Extract Settings" on page 6-1.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KEY_TEMPLATE</td>
<td>STANDARD</td>
<td>Default Key Template Name</td>
</tr>
<tr>
<td>2</td>
<td>DOMAIN</td>
<td>STANDARD</td>
<td>Domain of Default Key Template</td>
</tr>
<tr>
<td>3</td>
<td>SEPARATE_SAS_YN</td>
<td>N</td>
<td>Separate Oracle and SAS names?</td>
</tr>
</tbody>
</table>
### DX_EXTENDED_ATTRIBUTES

This codelist contains settings for Global Library Questions extended attribute creation.

Oracle Clinical offers the option of enabling additional question or response attributes for use in building views and view templates. The extended attributes are enabled at database creation when a population script, `vb_pop_que_attr.sql`, reads the installation reference codelist `DX_EXTENDED_ATTRIBUTES`.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>DCM_LINK_YN</td>
<td>Y</td>
<td>DCM default views are linked to source DCM as default conditions?</td>
</tr>
<tr>
<td>5</td>
<td>ACTIVE_KTEDT_YN</td>
<td>N</td>
<td>Enable edit of active key templates?</td>
</tr>
<tr>
<td>6</td>
<td>ACTIVE_XMEDT_YN</td>
<td>Y</td>
<td>Enable edit of active extract macros?</td>
</tr>
<tr>
<td>7</td>
<td>FREE_KT_SEL_YN</td>
<td>N</td>
<td>Enable selection of nonaggregate, nondefault key template?</td>
</tr>
<tr>
<td>8</td>
<td>INC_VALSTAT_YN</td>
<td>N</td>
<td>Include validation status in default view definition?</td>
</tr>
<tr>
<td>9</td>
<td>DVGDEFAULT_SN</td>
<td>N</td>
<td>Include DVG sequence number in default view definition?</td>
</tr>
<tr>
<td>10</td>
<td>DVGDEFAULT_SV</td>
<td>Y</td>
<td>Include DVG short value in default view definition?</td>
</tr>
<tr>
<td>11</td>
<td>DVGDEFAULT_LV</td>
<td>N</td>
<td>Include DVG long value in default view definition?</td>
</tr>
<tr>
<td>12</td>
<td>THESAURUS_TERM1</td>
<td>Y</td>
<td>Include thesaurus term1 in default view definition?</td>
</tr>
<tr>
<td>13</td>
<td>THESAURUS_TERM2</td>
<td>Y</td>
<td>Include thesaurus term2 in default view definition?</td>
</tr>
<tr>
<td>14</td>
<td>THESAURUS_TERM3</td>
<td>Y</td>
<td>Include thesaurus term3 in default view definition?</td>
</tr>
<tr>
<td>15</td>
<td>SAS_ATTREDT_YN</td>
<td>N</td>
<td>Enable update of SAS and Oracle column names?</td>
</tr>
<tr>
<td>16</td>
<td>INC_FULL_VALTXT</td>
<td>N</td>
<td>Include Full Value Text in default view definition?</td>
</tr>
<tr>
<td>17</td>
<td>MAX_AUDIT_LEN</td>
<td>200</td>
<td>Maximum length of Audit Comment</td>
</tr>
<tr>
<td>18</td>
<td>MAX_DATCOM_LEN</td>
<td>200</td>
<td>Maximum length of Data Comment</td>
</tr>
<tr>
<td>19</td>
<td>DVG_LNG_LEN</td>
<td>200</td>
<td>Maximum length of DVG long value</td>
</tr>
<tr>
<td>20</td>
<td>ACTIVE_VTEDT_YN</td>
<td>N</td>
<td>Enable edit of active view templates?</td>
</tr>
<tr>
<td>21</td>
<td>ENABLE_VB</td>
<td>N</td>
<td>Enable View Builder as default in new studies?</td>
</tr>
<tr>
<td>22</td>
<td>VB_DVG_SUBSET</td>
<td>N</td>
<td>Use DCM Question-specific DVG attributes?</td>
</tr>
<tr>
<td>23</td>
<td>USE_DCM_SAS</td>
<td>N</td>
<td>Use DCM SAS Label as seed for attributes in default view definition?</td>
</tr>
<tr>
<td>24</td>
<td>FAST_VIEWS</td>
<td>N</td>
<td>Build fast views?</td>
</tr>
</tbody>
</table>
If you change the settings in this reference codelist, run the script vb_def_que_attr.sql again. VALIDATION_STATUS and DVG_SHORT_VALUE are enabled by default.

### Table 7–10  Extended Attributes for Questions

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDIT_COMMENT_TEXT</td>
<td>Audit comment text from the response.</td>
</tr>
<tr>
<td>DATA_CHANGE_REASON_TYPE_</td>
<td>Code value for the reason the data was changed.</td>
</tr>
<tr>
<td>CODE</td>
<td></td>
</tr>
<tr>
<td>DATA_COMMENT_TEXT</td>
<td>Data comment text from the response.</td>
</tr>
<tr>
<td>DISCREPANCY_INDICATOR</td>
<td>Discrepancy indicator for the response.</td>
</tr>
<tr>
<td>DVG_LONG_VALUE</td>
<td>DVG long value, if this is a DVG question.</td>
</tr>
<tr>
<td>DVG_NUMBER</td>
<td>DVG sequence number, if this is a DVG question.</td>
</tr>
<tr>
<td>DVG_SHORT_VALUE</td>
<td>DVG short value, if this is a DVG question.</td>
</tr>
<tr>
<td>VALUE_TEXT</td>
<td>Value of the response, if not.</td>
</tr>
<tr>
<td>EXCEPTION_VALUE_TEXT</td>
<td>Exception value text for the response.</td>
</tr>
<tr>
<td>FULL_VALUE_TEXT</td>
<td>Response value, or, if error, the exception text.</td>
</tr>
<tr>
<td>VALIDATION_STATUS</td>
<td>Validation status of the response</td>
</tr>
<tr>
<td>TERM_COL1</td>
<td>Thesaurus DVG first term column</td>
</tr>
<tr>
<td>TERM_COL2</td>
<td>Thesaurus DVG second term column</td>
</tr>
<tr>
<td>TERM_COL3</td>
<td>Thesaurus DVG third term column</td>
</tr>
</tbody>
</table>

### DX_INDEX_TABLESPACE

This codelist contains a list of index tablespaces for use with the Data Extract View Builder. The long values appear in the list of values for the Index Tablespace field in the Maintain Study Access Accounts window under Conduct, then Data Extract.

You must:

- Uncheck the **Active** check box for the long values RXC_APP_IDX_TSPA and RXC_DCD_IDX_TSPA. These values are not valid and should not be available to users.
- Enter the names of each tablespace you create for the purpose of containing data extract indexes, and check its **Active** check box. See "Creating Tablespaces for Data Extract Tables and Indexes" on page 6-5 for more information.

### DX_KEY_NAME

This codelist contains the column alias for a data extract view keys

### DX_ROLES

This codelist contains the list of default roles for accessing data extract views. Values set here appear in a list of values in the View Definition window; users with the selected role have access to the view. You can add roles; see "Controlling Access to Data Extract Views" on page 6-9 for more information.
**DX_VIEW_TABLESPACE**

This codelist contains a list of view tablespaces for use with the Data Extract View Builder. The long values appear in the list of values for the Tablespace field in the Maintain Study Access Accounts window under Conduct, Data Extract.

You must:
- Uncheck the **Active** check box for the long values RXC_APP_TSPA and RXC_DCD_TSPA. These values are not valid and should not be available to users.
- Enter the names of each tablespace you create for the purpose of containing data extract tables, and check its **Active** check box. See "Creating Tablespaces for Data Extract Tables and Indexes" on page 6-5 for more information.

**EXECUTION SCHED CODE**

This codelist contains the execution schedule, which is used in MODULE_EXECUTIONS.

**EXP DESIGN TYPE CODE**

This codelist contains experimental design types for studies.

**EXTERNAL_TRANS_TYPE**

This codelist provides transaction types for the Data Capture API function SetExternalContext.

**LAB RANGE SUBSET CODE**

This codelist contains the set of Lab Range Subset Codes.

**MANUAL SOURCE TYPE CODE**

See "Reason Codes and Descriptions for Manual Discrepancies" on page 3-11 for information.

**MAPPING_TYPE**

This codelist contains the list of user codes for mapping type. Refer to Chapter 11, "Setting Up File Viewing."

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HTTP</td>
<td>HTTP</td>
<td>HTTP Transfer protocol</td>
</tr>
<tr>
<td>2</td>
<td>FTP</td>
<td>FTP</td>
<td>FTP Transfer protocol</td>
</tr>
<tr>
<td>3</td>
<td>HTTPS</td>
<td>HTTPS</td>
<td>Secure-socket layer HTTP protocol</td>
</tr>
<tr>
<td>4</td>
<td>UNC</td>
<td>UNC</td>
<td>Universal Naming Convention protocol</td>
</tr>
</tbody>
</table>

**MEDICAL EVAL TYPE CODE**

This codelist contains the values you use to categorize Questions. The values appear in the lists of values in the Medical Evaluation Type field in the Maintain Questions window.
OBJECTIVE TYPE CODE

This codelist contains the list of types of Clinical Planned Objective.

OCL_DOMAINS

This codelist contains the installation-wide list of global library domains. For information, see Oracle Clinical Creating a Study.

OCL_INSTALLATION

This codelist defines the location code of the database that owns the Global Library. The system requests and stores this value during database creation. Oracle Clinical checks the local reference codelist OCL_STATE to determine if the location code specified for the current database matches the one for the Global Library stored in OCL_INSTALLATION. If these values match, Oracle Clinical allows updates to the Global Library.

OCL_OPTIONS_TYPE_CODE

This codelist contains a list of Optional Subsystems.

Table 7–12  Default Values for OCL_OPTIONS_TYPE_CODE Reference Codelist

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TMO_INSTALLED</td>
<td>Y</td>
<td>TMO Installed (obsolete).</td>
</tr>
<tr>
<td>2</td>
<td>TMS_INSTALLED</td>
<td>Y</td>
<td>Set to Y if TMS installed; to N if not.</td>
</tr>
<tr>
<td>3</td>
<td>VAL_STATUS</td>
<td>Y</td>
<td>If set to Y, Batch Validation skips documents that have had a validation status change since</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the last validation and are currently open (locked) in OC data entry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If set to NONE, Batch Validation processes such documents.</td>
</tr>
<tr>
<td>4</td>
<td>SR_INSTALLED</td>
<td>Y</td>
<td>Set to Y if Symmetric Replication installed.</td>
</tr>
</tbody>
</table>

OPA_MENU_ROLES

This codelist contains the list of menu roles available in your Oracle Clinical installation. Any custom roles you create are not available until you enter them in this reference codelist; see "Adding a Custom Role to OPA_MENU_ROLES" on page 2-8 for information.

PATIENT STATUS CODE

This codelist describes patient statuses such as enrolled or terminated.

PLAN STUDY INT TYPE CODE

This codelist describes types of Planned Study Intervals, such as baseline, dosing, and qualifying.

PROCEDURE TYPE CODE

This codelist contains the list of user code for PROCEDURES.
QUESTION CATEGORY TYPE CODE

This codelist contains the list of user codes for QUESTION_CATEGORY_RELATIONS.

RDCI CHANGE REASON TYPE CODE

RDCI change reason code for Oracle Clinical data entry and mass changes (key changes and soft deletes), RDC, and DCAPI applications during key data update and when deleting an accessible CRF. The values in this codelist are listed and described in Table 7–13, "Initial values for the RDCI CHANGE REASON TYPE CODE Reference Codelist”.

This following subsystems use this reference codelist during processing to provide a change reason for audited changes:

1. Patient Transfer
2. Lab Assignment Criteria
3. Mass Changes
4. Data Entry Login

The user is always prompted for a change reason during items 1 through 3. During data entry login, the user is prompted for a change reason when the document is internally accessible. For each case in which a user-supplied change reason is required, the system prompts the user and presents the list of values in this codelist as the set of options from which the user chooses.

The initial set of values in this codelist are copied from the DATA CHANGE REASON TYPE CODE reference codelist.

You can choose which values should not appear in RDC HTML Data Entry by adding 'NOTRDC' as the long value. HTML Data Entry displays all active values in the reference codelist that do not have ‘NOTRDC’ specified as the long value.

Although you can customize this codelist, note that the following set of values are reserved by the system and cannot be used as custom values: 'PASS1', 'PASS2', 'BATCH', 'UPDATE', 'REMOVED', 'TRANSLATION', 'KEY CHANGE', 'BROWSE', 'RECONCILIATION'.

<table>
<thead>
<tr>
<th>Short value</th>
<th>Long value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRA CORR</td>
<td>CRA Correction</td>
<td></td>
</tr>
<tr>
<td>DATA ENTRY ERR</td>
<td>Error during data entry</td>
<td></td>
</tr>
<tr>
<td>BATCH</td>
<td>Batch-loaded data</td>
<td></td>
</tr>
<tr>
<td>CRA CORR-INV</td>
<td>CRA corrected, after consulting with the investigator</td>
<td></td>
</tr>
<tr>
<td>CRA CORR-SRC</td>
<td>CRA correction, after consulting the source data</td>
<td></td>
</tr>
<tr>
<td>INV CORR</td>
<td>Investigator correction</td>
<td></td>
</tr>
<tr>
<td>STUDY ASSUM</td>
<td>Study assumption</td>
<td></td>
</tr>
<tr>
<td>THES CLARIF</td>
<td>Thesaurus clarification</td>
<td></td>
</tr>
<tr>
<td>ANALYSIS CORR</td>
<td>Analysis correction</td>
<td></td>
</tr>
<tr>
<td>REMOVED</td>
<td>RDCI removed</td>
<td></td>
</tr>
</tbody>
</table>
Installation Reference Codelists

Table 7–13 (Cont.) Initial values for the RDCI CHANGE REASON TYPE CODE Reference Codelist

<table>
<thead>
<tr>
<th>Short value</th>
<th>Long value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAL STATUS CHG</td>
<td>Validation status changed</td>
<td></td>
</tr>
<tr>
<td>DATA ENTRY MODE</td>
<td>Data entry mode</td>
<td></td>
</tr>
</tbody>
</table>

RDCI CHANGE REASON2 TYPE CODE
An additional set of RDCI change reason codes that are used by DCAPI. This codelist allows you to provide an additional list of valid change reason values that can be used by your external applications.

There are no default values for this codelist. Although you can customize this codelist, note that the following set of values are reserved by the system and cannot be used as custom values: 'PASS1', 'PASS2', 'BATCH', 'UPDATE', 'REMOVED', 'TRANSLATION', 'KEY CHANGE', 'BROWSE', 'RECONCILIATION'.

REGION TYPE CODE
This codelist contains types of region such as state, country, or continents.

RETIREMENT REASON TYPE CODE
This codelist contains the list of user codes for DCIS, DCMS, DVGS, PROCEDURES, QUES, QUES_GROUPS.

ROUTE OF ADMIN TYPE CODE
This codelist contains routes of drug administration such as oral, intravenous, and inhalation.

SAS_FORMATS
This codelist contains a list of optional subsystems.

SINGLE DCI TYPES
This codelist contains DCI types that do not allow duplicates. That is, the user is not permitted to add DCIs in this codelist as unplanned pages to a visit. Also, if the user adds an unplanned visit to the study, only those DCIs that are not listed in this codelist are included in the new visit.

The short value must correspond to a short value in the DCM DCI QG TYPE CODE installation reference codelist because DCIs can only be created of DCI types defined in that codelist.

The default values are listed in Table 7–14, "Reference Codelist SINGLE DCI TYPES".

Table 7–14 Reference Codelist SINGLE DCI TYPES

<table>
<thead>
<tr>
<th>Seq</th>
<th>Short Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DEMOGRAPHY</td>
</tr>
<tr>
<td>2</td>
<td>COMPLETION</td>
</tr>
</tbody>
</table>
SOURCE LOCATION CODE
This codelist contains the unique location code for each location in the installation.

The Global Library-owning location maintains this codelist. The Long Value field stores the offset in hours of that location from Greenwich Mean Time (GMT). Locations in a time zone east of Greenwich, England have a positive offset and those to the west have a negative offset. This reference codelist is replicated to the other locations as part of Global Library replication.

STANDARDS AFFIL TYPE CODE
This codelist contains the user code for STANDARDS_AFFILIATIONS.

STUDY STATUS TYPE CODE
This codelist contains types of Clinical Study Status.

TMS_CONFIGURATION
This codelist contains settings that control various default behaviors in the Oracle Thesaurus Management System. See the Oracle Thesaurus Management System User’s Guide for details.

TMS_OMISSION_ASSIGNMENT
This codelist controls the default behavior for omission assignment in the Oracle Thesaurus Management System (TMS). See the Oracle Thesaurus Management System User’s Guide for details.

TMS_OMISSION_STATUS
This codelist controls the default behavior for TMS Omission Status.

TMS_QUERY_TYPE
This codelist populates the Query fields in the Oracle Thesaurus Management System (TMS) with default choices for query type. See the Oracle Thesaurus Management System User’s Guide for details.

TMS_X_SEARCH
This codelist populates the list of values for the Search Type field in Oracle Thesaurus Management System (TMS). It currently has only one value, Cross Search. See the Oracle Thesaurus Management System User’s Guide for details.

TREAT CHG REASON TYPE CODE
This codelist contains the list of valid Treatment Change Reason Type Codes.

UNITS_OF_MEASURE_TYPE_CODE
This codelist contains units of measure that are available for use in studies in the database.
USER GROUP ROLES

See "Mapping Database Roles to User Roles" on page 3-1.

USER GROUPS

See "Specifying User Roles for the Oracle Clinical Discrepancy Database" on page 3-2 for information on this reference codelist.

VALIDATION FAILURE TYPE CODE

User code for DCM_QUESTIONS, PROCEDURE_DETAILS, QUES, QUES_GROUP_QUES

Design Installation Reference Codelists

Design Installation codelists are a subset of the Installation Reference Codelists relating to study design. Using a database role with access to the appropriate menu item, you can grant a study designer access to either of the Design Installation Codelists windows (Query or normal), and enable the designer to maintain this subset of the reference codelists without providing access to change all of the installation reference codelists.

Each of the design installation codelists is documented in the main Installation Codelists section:

- APPLICATION AREA CODE
- BLIND TYPE CODE
- CLINICAL PHASE
- DOSE FORM TYPE CODE
- DOSE FREQUENCY TYPE CODE
- EXP DESIGN TYPE CODE
- OBJECTIVE TYPE CODE
- PLAN STUDY INT TYPE CODE
- REGION TYPE CODE
- ROUTE OF ADMIN TYPE CODE
- STUDY STATUS TYPE CODE
- TREAT CHG REASON TYPE CODE

System Reference Codelists

System reference codelists provide values for Oracle Clinical’s internal use only. You can browse or query for system codelist values, but you cannot change them. System codelists provide standard values across distributed environments.

To access the System codelists, navigate to Admin, Reference Codelists, and then Qry System Codelists. Alternatively, you can run a Reference Codelist report for one or all system codelists. Refer to the "Running the Reference Codelists Report" section on page 7-3 for instructions.
Part II
Oracle Clinical Administration Tasks

This section includes tasks involved in running Oracle Clinical:

- Chapter 8, "Managing Batch Jobs"
- Chapter 9, "Partitioning and Indexing"
- Chapter 10, "Utilities"
- Chapter 11, "Setting Up File Viewing"
- Chapter 12, "Enabling Image Viewing"
Managing Batch Jobs

The Oracle Clinical parameterized job and report submission facility (PSUB) submits jobs to execute either on the Reports Server or on the back end server. Oracle Clinical uses Oracle Reports for reports, job sets and scheduling. The back end server, also called the PSUB server, handles only jobs implemented in PL/SQL or a third-generation language (3GL). You cannot alter which server is used for a given job; this is determined by the way in which each PSUB job has been coded.

When you take an action runs a report or PSUB job, the job submission form displays the type of job (either PSUB or Reports Server) at the top of the form.

Use the information in this section to manage PSUB jobs, such as:
- reviewing the installation
- diagnosing and recovering from installation failures, and
- understanding how other system changes affect the consistency of the installation.

**Note:** For information on managing Oracle Reports jobs, see the Oracle Application Server Reports Queue Manager documentation.

This section includes the following topics:
- **User Account Requirements for Batch Jobs** on page 8-1
- **How PSUB Handles a Request** on page 8-2
- **Managing the PSUB Process** on page 8-3
- **Batch Job Reference Codelists** on page 8-13

### User Account Requirements for Batch Jobs

Before submitting batch job requests, a user should have the following:
- an OPS$ database account
- an account on the PSUB server
- a record in the Oracle Accounts table
- a log directory on the PSUB server
- a log directory for reports output

For instructions on setting up user accounts, see "Setting Up User Accounts" on page 1-1.
How PSUB Handles a Request

When a client issues a PSUB request to run a job or report, it sends a message, via a database pipe to the PSUB process, that includes the batch job ID and the primary key into the RXC.BATCH_JOBS table.

When the process receives this message from the database pipe, it reads the job information from the RXC.BATCH_JOBS table and submits the job on the user's behalf with the PSUB Launcher (PSLAUNCH).

In submitting the job, the process creates the following files, where $nnnn$ represents the batch job ID reported to the user in a dialog box.

The system creates these files in the user's RXC_LOG directory, which you can see in the Oracle Accounts form:

<table>
<thead>
<tr>
<th>Filename format</th>
<th>Description of file contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$nnnn$.log</td>
<td>Log file of a user's PSUB job. Note that the initial character is the letter L, not the number one.</td>
</tr>
<tr>
<td>$nnnn$.out</td>
<td>Output from a report job.</td>
</tr>
<tr>
<td>$nnnn$.log</td>
<td>Log file for a print request from the Submitted Batch Jobs form.</td>
</tr>
</tbody>
</table>

Asynchronous PSUB Requests

The DBMS_PIPE mechanism is an asynchronous protocol, so messages remain in the pipe until read by the process. This means that if a user makes a PSUB request and the process is not running, the request is read when the process is started again. Also, the DBA can safely stop and restart the process in a production environment, provided the delay is not too long. All nonblocking requests are processed, and all blocking requests start when the process comes up or times out.

Blocking and Nonblocking Jobs

Blocking jobs are usually of short duration and run in a mode where the system does not allow the user to proceed before their completion. Job completion status (SUCCESS or FAILURE) can be reported to the user immediately after the job completes. Blocking jobs in Oracle Clinical include:

- Default layout generation
- Moving a data entry screen to production
- Generating a Validation Procedure

Note: On Windows only:

- All Oracle Clinical users must also have operating system-level accounts on the Windows server running the PSUB service.
- The Windows server that runs the PSUB service must belong to the same domain as the Windows server that runs the Oracle database.

This Oracle database security feature prevents unauthorized users from logging in over a network connection. For information on database security, see your Oracle database documentation.
When an application server submits a blocking job, it sends information through a common database pipe and then waits on an application server-specific receiving pipe for completion (execution) information, which it receives from PSUB. The pipe is maintained by the application and is named with a unique session name. Time-out on the receiving pipe for the application server occurs after 5 minutes, at which point the following message appears: "Failed to get response from server - Batch Job Id <nnnn>". Timing out happens for a variety of reasons, including:

- The rxcprod user is not set up properly.
- Job execution is slow, so more than 5 minutes elapse before completion.
- The PSUB process is not running.
- Too many jobs in the batch queue cause the current job to be pending; or it started after the other jobs were completed.
- The server machine is too slow for the current load.
- The job is executing but is waiting for some resource.
- The client is not connected to an OPS$ Oracle account; or the client does not have an operating system account on the server.
- The queue is in a stopped state.

Nonblocking jobs include all reports, all jobs launched from the PSUB submission screen, and randomization.

**Checking a Nonblocking Batch Job**

Users are not notified when nonblocking batch jobs complete. To check the job’s status, they can execute a query in the Submitted Batch Jobs form, accessible via:

- the Job Status button in the Submission of module window
- the Batch Jobs item from the Action in-form menu
- this command, entered from the rxcprod or system manager account:
  
$ at -l

**Managing the PSUB Process**

This section contains the following topics:

- Starting and Stopping PSUB Manually in UNIX on page 8-4
- Starting and Stopping PSUB Automatically in UNIX on page 8-4
- Starting and Stopping PSUB Manually in Windows on page 8-6
- Starting PSUB Automatically in Windows on page 8-7
- Changing PSUB Job Number Sequencing on page 8-9
- Viewing the Status of a Submitted Batch Job on page 8-9
- Removing the PSUB Service on page 8-9
- Viewing Log and Output Files on the Screen on page 8-10
- Using Job Sets to Control Execution Order on page 8-11
- Tracking PSUB Processes on page 8-12
Starting and Stopping PSUB Manually in UNIX

This section contains the following topics:

- Starting PSUB Manually in UNIX on page 8-4
- Stopping PSUB Manually in UNIX on page 8-4

Starting PSUB Manually in UNIX

To start the process on UNIX:

1. Log on as rxcprod to the UNIX back end server computer.
2. Run the startup script.

The startup script has the following syntax:

```
start_psub instance environment verbose
```

where `instance` refers to a generic instance name, and `environment` to a code environment. This starts up PSUB in verbose mode, which means additional information is placed in the process log file to help in debugging PSUB problems.

Log in as rxcprod to start PSUB. For example:

```
$ start_psub prod 46 verbose
```

If the process is already running, the script exits with an error message.

The `start_psub` shell script is run interactively as part of the user’s current shell. This script runs the executable `rxcpsdps` as a background process, which is the PSUB process. Creating the background process should take only a few seconds.

Search for an instance of the `rxcpsdps` process.

For example:

```
$ ps -ef | grep -i rxcpsdps
```

```
rxcpod  0 12750  1 0 15:52:43 ? 0:00 rxcpsdps verbose dev 40
rxcpod  0 22142  1 0 18:50:39 ? 0:01 rxcpsdps verbose test 40
```

If the PSUB process fails to start, check for errors in `rxcpsd_instance_environment_1.log` and `rxcpsd_instance_environment_2.log` in the `$RXC_CENTRAL_LOG` directory.

Stopping PSUB Manually in UNIX

The preferred way to stop the PSUB process, because it ensures a graceful exit, is with the following utility, from either the opapps or rxcprod account, after setting the correct environment:

```
stop_psub instance environment rxc_password
```

Starting and Stopping PSUB Automatically in UNIX

On UNIX systems, you can automate the process of starting and stopping PSUB.

Starting PSUB Automatically in UNIX

The following example shell scripts for Sun Solaris show how to make the process start automatically at system startup:

```
# File: /etc/init.d/dbora
ORA_HOME=/u01/app/oracle/product/11.1.0.7.0
```
ORA_OWNER=oracle
if [ ! -f $ORA_HOME/bin/dbstart -o ! -d $ORA_HOME ]
then
  echo 'Oracle startup: cannot start'
  exit
fi
case "$1" in
  'start')
    echo 'Starting Oracle...'
    su - $ORA_OWNER -c $ORA_HOME/bin/dbstart
    su - $ORA_OWNER -c 'lsnrctl start'
    su - rxcprod -c start_psub
    ;;
  'stop')
    echo 'Stopping Oracle...'
    su - rxcprod -c stop_psub
    su - $ORA_OWNER -c $ORA_HOME/bin/dbshut
    ;;
esac

# File: start_psub
# Start database 1
start_psub venus 46
# Start database 2
start_psub pluto 46

**Stopping PSUB Automatically in UNIX**

With an OPS$ Oracle account you can automate the shutdown of the PSUB process on UNIX so that it does not require the entry of a password. For example, you can grant the OPS$RXCPROD user access to shut down the PSUB process:

```bash
$ sqlplus rxc/password
SQL> grant execute on stop_psub_daemon to ops$rxcprod
```

The PSUB process may then be shut down with the command:

```bash
$ rxcpstop.sh /
```

The entire process of starting up and shutting down Oracle Clinical instances can be automated. The following example shell scripts for Sun Solaris show how.

```bash
# File: /etc/init.d/dbora
ORA_HOME=/u01/app/oracle/product/11.1.0.7.0
ORA_OWNER=oracle
if [ ! -f $ORA_HOME/bin/dbstart -o ! -d $ORA_HOME ]
then
  echo 'Oracle startup: cannot start'
  exit
fi
case "$1" in
  'start')
    echo 'Starting Oracle...'
    su - $ORA_OWNER -c $ORA_HOME/bin/dbstart
    su - $ORA_OWNER -c 'lsnrctl start'
    su - rxcprod -c start_psub
    ;;
  'stop')
    echo 'Stopping Oracle...'
    su - rxcprod -c stop_psub
    su - $ORA_OWNER -c $ORA_HOME/bin/dbshut
    ;;
```
esac
   # File: stop_psub
   # Stop database 1
   stop_psub venus 46
   # Stop database 2
   stop_psub pluto 46

Starting and Stopping PSUB Manually in Windows

This section contains the following topics:

- Installing PSUB in Windows on page 8-6
- Installing PSUB in Windows on page 8-6
- Stopping PSUB Manually in Windows on page 8-7

Installing PSUB in Windows

On Windows, you must first install the PSUB process as a service:

1. Log in to the Windows server as Administrator.
2. Open a Command Prompt window and enter the following commands:
   
   set p1=database-connect-string
   set p2=code-environment
   opa_setup
   cd %RXC_BIN%
   rxcpsdps -install database-connect-string database-instance-name

3. Open the computer’s Services control panel.
4. In the Services dialog box, select the PSUB service and open the Properties window.
5. In the General tab window, set the Startup Type to Manual.
6. In the Log On tab window, click This Account, and choose rxcprod. The system prompts you for the password for the rxcprod user.
7. Enter and confirm the password, and click OK.
8. Log out from this Administrator session.

Starting PSUB Manually in Windows

After installing the PSUB service and setting its parameters as Administrator, you start it as rxcprod:

1. Log in as rxcprod.
2. Open the computer’s Services control panel.
3. In the General tab window, select the PSUB Service and enter values for the Start Parameters as follows:

   database-connect-string  code-environment [verbose | noverbose] value-of-rxc_root

   For example:
   
   db2x2 46 verbose c:\opapps\oc\46

   Note that if you need a backslash (\) in the text box, you must double it (\\).
4. Click the Start button.
5. Click OK to close the PSUB Service Properties window.

**Stopping PSUB Manually in Windows**

Do not use Enterprise Manager to stop rxcprod sessions on Windows; instead, use the Control Panel on the appropriate local machines.

To stop the PSUB service, follow these steps:

1. Log in as rxcprod.
2. Open the Services control panel.
3. In the Services dialog box, select the PSUB service for the particular database and open the Properties window. The PSUB service for db2x2, for example, might appear as:
   ```
   psub service db2x2
   ```
4. In the Properties window, click Stop.

**Starting PSUB Automatically in Windows**

This section contains the following topics:

- Creating a System Environment Variable on page 8-7
- Creating a Batch File on page 8-8
- Scheduling the Batch File and Testing the Setup on page 8-8
- Adding a Shortcut on page 8-8

You must first install PSUB; see "Installing PSUB in Windows" on page 8-6.

The batch file is required in large databases that take so much time to come up during a server reboot so that the system tries to start PSUB before the database is fully up. In this case the PSUB process does not start and an error like the following appears in the PSUB log file (found in the drive:\opapps\oc\46\log directory):

```
ERROR:Daemon error while connecting:/@devoc
ORA-1033: ORACLE initialization or shutdown in progress
```

**Creating a System Environment Variable**

You can specify that the PSUB service starts automatically when the Windows PSUB server re-boots. The service parameters are read from system environment variable with name `PSUBSERVICE` concatenated with database name, such as:

```
PSUBSERVICESUN6X2
```

where "SUN6x2" is the database name.

To create a system environment in Windows:

1. Select System in the Control Panel.
2. Select the Advanced tab, select Environment Variables.
3. Define the System Environment variable in the lower portion of the window, using the format: `database id code environment verbose RXC_ROOT`

   For example, for the variable "PSUBSERVICESUN6x2":
   ```
   sun6x2 46 verbose t:\opapps\oc\46
   ```
Creating a Batch File

Create a batch file called 'psub_start1.bat' in the %RXC_ROOT%/log directory (for example, D:\opapps\oc\46\log) with the following contents.

```
cmd /c echo Current Date/time= %DATE% %TIME% > psub_start1.log
cmd /c echo Starting Time Delay > psub_start1.log
ping localhost -n 180 > nul
cmd /c net start "PSUB Service database_id" > psub_start1.log
```

Notes:
- "PSUB Service database_id" is the PSUB service name that appears in the Services window (under Administrative Tools from the Control Panel).
- You can repeat the following command for different databases if needed:
  
  ```
  cmd /c net start "PSUB Service database_id" > psub_start1.log
  ```
- The 'ping localhost' command introduces a time delay to ensure that the database is up before starting PSUB. You can increase this value—set to 180 seconds (3 minutes) in the example above—if required.

Scheduling the Batch File and Testing the Setup

To schedule batch file execution:

1. Make sure that the Task Scheduler service is started in the Services Window (under Administrative Tools from the Control Panel).
2. Navigate to Scheduled Tasks in the Control Panel.
3. Open the Add Scheduled Task wizard and click Next. The Scheduled Task Wizard window appears.
4. Click on Browse. The Browse window appears.
5. Go to the directory where 'psub_start1.bat' is saved and then select psub_start1.bat. The task name is displayed as psub_start1. You can modify the name.
6. Under 'Perform this task, select 'When my computer starts' and click Next.
7. On the next screen ensure that the ADMINISTRATOR username/password is used.
8. Click Next and click Finish.
   
   To test, shut down the service if necessary (see "Stopping PSUB Manually in Windows" on page 8-7) and double-click on file 'psub_start1.bat' to test that it starts the PSUB service. Verify that the log file 'psub_start1.log' is created in the same directory unless a different path was specified.
9. In the Services Window under Administrator Tools in the Control Panel, change the Startup Type of the PSUB Service to Manual.
   
   To test, restart the computer and check the Services window to see if the PSUB service has started. If it has, submit a PSUB job such as Batch Validation and check if it runs.

Adding a Shortcut

For convenience add a shortcut for psub_start1.bat on the desktop to manually start PSUB by double-clicking the icon.
Changing PSUB Job Number Sequencing

A sequence generator numbers Oracle Clinical submitted batch jobs. By default, at each submission the generator increments by 10 the database seed number you provide during back end installation. Change the default by running alter_psub_seq.sql, found in the INSTALL directory, which lists current settings and asks for:

- start value – number to append to the initial job ID
- increment – the value to add to the job ID for each subsequent job.

For example:

<table>
<thead>
<tr>
<th>Start Value</th>
<th>Increment</th>
<th>Job ID Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>1, 11, 21, 31...</td>
</tr>
<tr>
<td>21</td>
<td>100</td>
<td>21, 121, 221, 321...</td>
</tr>
</tbody>
</table>

The value entered for the "start value" for the PSUB batch job number does not need to be the same as the database seed.

Tips: If users are accessing multiple databases, keep the batch job numbers generated by each database unique so that the log files do not collide.

Keep the increment of the batch job numbers as small as possible so that batch job numbers do not grow too large.

Viewing the Status of a Submitted Batch Job

You view the status of submitted batch jobs in the Submitted Batch Jobs window, which you access by selecting: Admin, then PSUB/Reports Jobs, and Batch Jobs. This window provides information about the batch jobs, including logs, output file names, and stop jobs you have submitted. The most recently submitted jobs are listed first.

You cannot use this window to update fields. If you are viewing this form while the job is executing, requery the form periodically to see the statuses change, or press the Auto Refresh button. Press the button again to turn off the auto query.

Batch job execution statuses:

- ENTERED = User has requested submission of the job.
- SUBMITTED = Job has been submitted to the batch queue.
- SUBMIT_FAILED = Job failed to be submitted to the batch queue.
- STARTED = Job is currently executing.
- SUCCESS = Job has completed successfully.
- FAILURE = Job has completed unsuccessfully. Reason displayed in Failure Text field.
- STOPPED = Job has been stopped by the Stop button.
- STOP_FAILED = Job has not responded to the Stop button.

Removing the PSUB Service

On Windows, you can remove (uninstall) the PSUB service as follows:

1. Log in as Administrator.
2. Open a Command Prompt window and enter the following commands:

```plaintext
set p1=database-connect-string
set p2=code-environment
opa_setup
cd %RXC_BIN%
rxcpsdps -remove database-connect-string database_instance_name
```

3. Log out from this Administrator session.

### Viewing Log and Output Files on the Screen

This section includes the following topics:

- Viewing Logs That Concern Execution of PSUB Processes on page 8-10
- Viewing Logs for Individual PSUB Jobs on page 8-10

#### Viewing Logs That Concern Execution of PSUB Processes

The PSUB process writes to log files pointed to by the variable RXC_CENTRAL_LOG. RXC_CENTRAL_LOG is defined as $rxc_root/log (on UNIX), or %rxc_root%\log (on Windows). Do not redefine this variable to any other values.

**Note:** On UNIX systems, you must stop the PSUB process to view the contents of this log file, because it will be locked.

However, the following file can be read while the PSUB process is running:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Example path/filename for log file</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX</td>
<td><code>$RXC_CENTRAL_LOG/rxcpsd_instance_environment_1.log</code></td>
</tr>
<tr>
<td>Windows</td>
<td><code>%RXC_CENTRAL_LOG%/rxcpsd_instance_environment_1.log</code></td>
</tr>
</tbody>
</table>

If the process is running in verbose mode more information is written to the log file. You cannot switch from non-verbose to verbose modes while the process is running; you must stop and restart the process to switch mode.

#### Viewing Logs for Individual PSUB Jobs

Log and output files are always placed in the log directory in your home directory on the server on which PSUB is running. You cannot update log and output file names. The names of the files are structured such that the unique batch job ID for the job is given an:

- "l" prefix for the log file
- "o" for the output file.

For example, a batch job ID of 12345 would have an output file name of `$RXC_LOG/o12345.out` (for UNIX), or `%RXC_LOG%\o12345.out` (for Windows), and a log file name of `$RXC_LOG/l12345.log` (for UNIX), or `%RXC_LOG%\l12345.log` (for Windows).

To view the output or log file on the screen, click the View Output or View Log button and enter the relevant information in the window.
Printing Output or Log Files
To print the output or log file, press the Print Output or Print Log button. In the pop-up box, specify the printer.

For printing, PSUB uses the standard `lp` command (on UNIX) or print functions (on Windows). No other environment variables control printing.

Printing from PSUB jobs on a UNIX server to a Windows spooler is not supported.

Using Job Sets to Control Execution Order
You can create a job set to control the execution order when you want jobs that depend on other jobs to execute only if the jobs on which they depend have successfully executed. If any of the jobs in a job set exits with a status of SUBMIT-FAILED or STOPPED, the whole job set aborts with a status of FAILURE. The basic steps are:

1. Create a Job Set
2. Submit a Job Set

---

**Note:** All jobs included in a job set must have a saved parameter set. No job included in a job set can have a LOCAL_IMMEDIATE mode of execution.

---

Create a Job Set

1. Save a parameter set for each job to be included in the job set, if that has not been done already.
2. Select Admin, then PSUB Jobs, and Job Sets.
3. Enter a name for the job set (alphabetic characters only).
4. In the Job Label field, enter a short name for the first job in the sequence. You use this name in the fields under JOB LABEL to the right.

---

**Note:** No jobs included in a job set can have a LOCAL_IMMEDIATE mode of execution.

---

5. Enter the name of the saved parameter set for the job. list of values available.
6. Set the timeout limit for the job in minutes. Default: 720 (12 hours).
7. The next three fields set the execution order and conditional branching. Enter the job label for the job you want to run next if the current job runs successfully, fails, or times out. If you leave one of these fields blank, the entire job set will stop executing if the current job has the corresponding result. For example, if you leave the Failure field blank and the current job fails, Oracle Clinical stops executing the entire job set.

Oracle Clinical enters the task name for the parameter set you entered.

Repeat this process for each job in the job set.
Submit a Job Set
1. Select Admin, then PSUB Jobs, and Submit Job Set.
2. Enter a job set name. An list of values is available.
3. Click Submit Job.

A job set controls the execution order for a specified set of jobs, so that jobs that depend on other jobs will execute only if the jobs on which they depend have been successfully executed. If any of the jobs in a job set exits with a status of SUBMIT-FAILED or STOPPED, the whole job set aborts with a status of FAILURE. You create job sets by selecting Admin, then PSUB, and Job Set.

---

Note: It is not necessary to wait until Oracle Clinical has executed one job before you submit another job.

---

Tracking PSUB Processes

Before starting a PSUB process, you may want to know the answer to such questions as, "Where was PSUB last running?", or "Where was PSUB running on such-and-such a date?" You can query the table RXC.PSUB_PROCESS_LOG to find out, for a given database, the instance, the environment, and the time a PSUB process was started and the time it was stopped.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSUB_PROCESS_LOG_ID</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td>HOST</td>
<td>VARCHAR2(64)</td>
<td></td>
</tr>
<tr>
<td>SERVER_OS</td>
<td>VARCHAR2(8)</td>
<td>The type of operating system on the server where PSUB runs</td>
</tr>
<tr>
<td>START_TS</td>
<td>DATE</td>
<td>The time stamp when the PSUB process started</td>
</tr>
<tr>
<td>CODE_ENVIRONMENT</td>
<td>VARCHAR2(20)</td>
<td></td>
</tr>
<tr>
<td>VERBOSE</td>
<td>VARCHAR2(1)</td>
<td></td>
</tr>
<tr>
<td>STOP_TS</td>
<td>DATE</td>
<td>The time stamp when the PSUB process stopped</td>
</tr>
</tbody>
</table>

For example, this query will give you the host and code environment of the last time PSUB was started against the database:

```
SQL> select start_ts, host, code_environment
  2  from psub_process_log
  3  where start_ts = (select max(start_ts) from psub_process_log);
```

This query will list all starts and stops, in time order:

```
SQL> select start_ts, stop_ts, host, code_environment
  2  from psub_process_log
  3  order by 1;
```
Batch Job Reference Codelists

Oracle Clinical manages batch jobs through entries in these reference codelists, refer to Chapter 7, "Reference Codelists" on page 7-1 for additional information:

- BATCH QUEUE NAME on page 7-7
- PRINT QUEUE NAME on page 7-18
- OCL_JOB_PREF on page 7-13
- REPORT_SERVER on page 7-18
- SAS_QUEUE on page 7-18
Oracle Clinical modifies the required index structure for the Responses table (as well as several other tables), and enables the use of partitioning for the Responses table. The baseline Release 4.6 upgrade does not change the index or partition structure of the Responses table because these changes can be time-consuming and require additional planning. You decide when to upgrade to the new indexes, and whether and when to upgrade for partitioning.

This section covers the following topics:
- guidelines for planning the upgrade process
- background information on partitioning
- procedures for performing the upgrades
- guidelines for maintaining partitioned Oracle Clinical databases.

This does not replace the Oracle technical documentation regarding partitioning or indexes. Oracle strongly recommends that the DBA in charge of the Oracle Clinical database be familiar with partitioning before attempting to convert an existing Oracle Clinical database to a partitioned Responses table.

For detailed and authoritative information on partitioning and on maintaining partitioned tables, refer to the Oracle database documentation, specifically the Concepts manual and the Administrator’s Guide.

**Introduction**

There are two basic sets of decisions you must make to plan the upgrade:
- Whether and when to convert to a partitioned Responses table
- If you are not partitioning, when and how to upgrade the index structure of the Responses table

These two decisions are inter-related — the conversion to partitioning subsumes the index upgrade. If you decide that you will be partitioning shortly after upgrading, you might want to skip the separate index upgrade. Conversely, if you are postponing partitioning you might want to do the separate index conversion soon after upgrade. If you are not converting to partitioning, you need to consider the path to index upgrade (see Figure 9–1).

If you have a fresh installation of Oracle Clinical, you do not need the index upgrade. You do need to decide about partitioning.
Timing Considerations and Deferral of Index or Partition Upgrade

The index upgrade and partitioning both require impact analysis in advance and both can take a significant amount of time to implement. Assuming that the impact analysis is done prior to upgrading and the upgrade path chosen, the primary consideration for when to upgrade is the elapsed time needed to perform the actual upgrades. While either could be done at the same time as the upgrade, deferring the index or partition upgrade to a separate time might reduce the risk and time pressure. The index upgrade is the least time-consuming and is easily performed overnight even on a large database. Partitioning is more time-consuming and, while it can easily be accomplished over a weekend on even the largest Oracle Clinical database, it needs more careful advanced planning.

If you decide to defer both partitioning and the index upgrade, you can force Oracle Clinical to continue to create validation and derivation procedures and data extract views that are optimized for the pre-4.0-style index structure (see "Enforcing Pre-Version 4.0 Optimization" on page 9-3).

The upgrade for Responses should be done at the same time as either partitioning or the Responses table index upgrade.

The index upgrade steps are documented in the "Upgrading Indexes" section on page 9-17. This upgrade can be done at any time that is convenient.
Enforcing Pre-Version 4.0 Optimization

If you decide to postpone both the partitioning and indexing upgrades, you can force Oracle Clinical to continue to create validation and derivation procedures and data extract views that are optimized for the pre-4.0-style index structure.

You do this by inserting a record in the local OCL_STATE reference codelist with the short value of USE RESP_DCMQG and a long value of YES.

Later, when you are ready to convert to the new index scheme or partitioning, change the long value to NO before you regenerate views and procedures. The absence of an entry is equivalent to NO.
About Partitioning

Partitioning is an Oracle database capability that allows you to divide a single Oracle table into separate physical partitions, each with its own storage characteristics. The indexes on a table can be partitioned as well. When a table is partitioned, each partition functions physically like a separate table while the table, as a whole, can still be treated as a single table for purposes of data access through SQL. Partitions can be managed like independent tables. They can be reorganized, imported, exported, taken off line and even dropped.

This section provides an overview of partitioning as it has been used for the Responses table in Oracle Clinical. For complete information on partitioning, see the Oracle Concepts manual and Administrator’s Guide.

How Has the Responses Table Been Partitioned?

The response table is range partitioned on clinical_study_id. This means that the data is placed into partitions based upon ranges of the internal clinical_study_id identifier. The two remaining indexes on the Responses table are equipartitioned with the Responses table on clinical_study_id. This means they share the same partitioning scheme as defined for the Responses table. This equipartitioning guarantees partition independence, that is, operations on one partition or its indexes do not affect any other partition or its indexes. Equipartitioning also means that the index partitions are automatically maintained during most partition maintenance operations on the Response table partitions. It is important to note that the Response table partitions and the index partitions have separate storage specifications, so it is still possible to place the index partitions on separate tablespace/physical devices from the table partitions.

Why Partition?

In Oracle Clinical Version 3.1, the Responses table together with its indexes is 20 times larger than the next largest table, Received DCMs. Partitioning the Responses table and reorganizing the indexes accomplish a number of goals:

- Improves Database and System Management
- Removes Single Point of Failure
- Improves Performance
- Reduces Space Requirements

Improves Database and System Management Partitioning improves database and system management in several ways. First, by dividing the Responses table into many physical segments, you can manage the physical growth and structure of the table at a finer level of granularity. Individual partition segments can be rebuilt independently to consolidate space and improve performance. Partition segments can be placed on separate tablespaces on separate physical devices to improve performance; for example, by moving studies that will be used for intensive reporting onto separate disks from studies with active entry. Studies that are no longer active can be periodically moved to a read-only tablespace on a physical device that need only be backed up after the periodic maintenance and not as part of nightly backups. Similarly, if table export is used as a backup mechanism, only active partitions need to be exported.
Removes Single Point of Failure  Prior to partitioning, a single bad index block would require rebuilding the entire index and a bad data block might require rebuilding the entire responses table. Since all of the partitioned indexes are local indexes, with partitioning, a bad index only affects the particular partition and a bad data block only requires the rebuild of that particular partition; the remaining partitions can remain in use while the one with problems is restored.

Improves Performance  Both transactional and retrieval performance are improved by partitioning. Transactional performance is improved in two ways: by reducing contention for the data associated with a particular study and reducing transaction overhead. Contention is reduced by distributing inserts, updates, and deletes over different data blocks for each study and by allowing data blocks that are actively being accessed for reporting to be separated from those that are actively being modified by data entry.

Retrieval performance is improved in two ways as well: by physically grouping study data and by allowing physical reorganization on a study basis. By allowing data for each study to be stored their own data and index blocks instead of interspersed with data from many other studies, retrieval by study performs significantly less disk I/O and, in fact, a study can usually be buffered entirely in the SGA. This can produce dramatically improved extract view performance. In addition, when a study is about to be heavily accessed for reporting, the particular partition can be reorganized to consolidate space used by the index and data blocks, thus reducing I/O, and even to place the partition on a separate, perhaps faster, physical device to optimize access.

Reduces Space Requirements  The most dramatic reduction in space requirements comes from the re-optimization of Oracle Clinical to do away with the need for three of the indexes on the Responses table. Combined with the Oracle database leading key compression on one of the remaining indexes, this drops the space required by the Responses table and its indexes by over 50%. This benefit can be realized whether or not you partition the Responses table. In addition, the ability to rebuild partitions individually means that space can be recovered from partitions that are no longer being modified by rebuilding them with reduced free space. Since freshly rebuilt indexes are frequently two-thirds the size of an actively growing index, and the free space requirement can be reduced by 10%, the net improvement over time could be an additional 30-40% reduction in space requirements.

Planning for Partitioning Upgrade

Before partitioning the Responses table you must decide how you want to partition it, both initially and on an ongoing basis. To do this, you need to understand how the partitioning is implemented and how this impacts your partition strategy.

Oracle recommends that you place all but the smallest active studies in separate partitions. Small studies that happen to have contiguous clinical_study_ids can share a single partition without impacting performance significantly. Inactive studies with contiguous clinical_study_ids can be merged into a single partition (see "Using Read-Only Partitions to Minimize Backup" on page 9-17).

Other partitioning schemes are possible and give differing degrees of performance and problem isolation benefits. One alternate approach is to segment the table into larger partitions that loosely correspond to time-slices based upon sequential clinical_study_id allocation. This approach minimizes partition maintenance and gives some problem isolation benefit, but is not likely to give much performance benefit because active studies will tend to share the same partitions.
Pre-upgrade Steps

To assist in the planning process, Oracle Clinical includes several listings and utilities to create the initial partitioning structure. A forms-based user interface is used to perform the actual partition definition prior to generating the partitioned Response table creation script.

Decide Partitions Needed

The following SQL statement provides a listing of all studies with data, the number of responses per study, the most recent response creation in the study, and the flag that indicates whether the study is frozen:

```
SELECT /*+ ordered */ cs.study, cs.clinical_study_id, css.frozen_flag,
       a.resp_count, a.max_date
FROM
  (SELECT count(*) resp_count, clinical_study_id,
       to_char(max(response_entry_ts), 'DD-MON-YYYY') max_date
   FROM responses
   GROUP BY clinical_study_id ORDER BY clinical_study_id) a,
  clinical_studies cs,
  clinical_study_states css
WHERE a.clinical_study_id = cs.clinical_study_id
AND cs.clinical_study_id = css.clinical_study_id
AND css.current_flag = 'Y'
ORDER BY clinical_study_id
```

Analyze the resulting list to determine which ranges of clinical_study_id can be combined into single partitions and which should be in their own partition. Among the issues to consider are:

1. Identify studies that will never have changes to their data and that are unlikely to have intensive reporting requirements in the future (see "Using Read-Only Partitions to Minimize Backup" on page 9-17).
   **Action:** Consolidate as much as possible and locate in read-only tablespace (see "Using Read-Only Partitions to Minimize Backup" on page 9-17).

2. Identify studies that are complete, but may have continued reporting requirements.
   **Action:** Keep in separate partitions, as appropriate by size, but allocate minimal free space. Consider placing tablespace on fast storage devices.

3. Identify contiguous ranges of small studies that will not grow to be large studies.
   **Action:** Consider consolidating into single partitions. This is especially relevant if there are studies that were created but will never contain data.

Define Partition Structure

Once you are prepared to define the initial partition structure, you use the following script to populate the actual table:

```
SQL> @populate_part_map_table.sql log_file_name.log
```

This script populates the mapping table with one row per study with storage clauses based upon the number of responses. The storage clauses are designed to result in a range of 1 to 15 extents for a given table size.
Implement Partition Structure

Once you have executed the script, you can use the Maintain Partition Mapping Tables form, shown in Figure 9–2 (navigate to Admin, then Partition Admin), to implement the partitioning scheme determined above.

You can delete rows to merge partitions. You can edit the storage clauses to reflect consolidation or to adjust the storage because you are aware of factors that affect the size, such as planned rapid growth of a partition. Bear in mind that you can always rebuild a partition later, independent of the upgrade process.

Figure 9–2  Maintain Partition Mapping Tables Window

<table>
<thead>
<tr>
<th>Partition Size</th>
<th>Max Responses</th>
<th>Table Extent Size</th>
<th>Index Extent Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Small</td>
<td>&lt;10K</td>
<td>256K</td>
<td>128K</td>
</tr>
<tr>
<td>Small</td>
<td>&lt;50K</td>
<td>512K</td>
<td>256K</td>
</tr>
<tr>
<td>Medium</td>
<td>&lt;100K</td>
<td>1M</td>
<td>512K</td>
</tr>
<tr>
<td>Large</td>
<td>&lt;500K</td>
<td>4M</td>
<td>2M</td>
</tr>
<tr>
<td>Very Large</td>
<td>500K+</td>
<td>16M</td>
<td>8M</td>
</tr>
</tbody>
</table>

Table 9–1  Default Storage for Oracle Clinical Responses Partitions

Note: You always specify the maximum clinical_study_id included in a partition. The minimum is implicitly one more than the maximum of the previous partition.
Planning for Partitioning Upgrade

Generating Table and Index Creation SQL
Once you have completed modifying the partition mapping, you use the script:

```
SQL> @gen_create_part_table.sql part_table_ddl.sql
```

to generate the partitioned table creation script (see Example 9–1) and:

```
SQL> @gen_create_part_index.sql part_index_ddl.sql
```

to generate a creation script for both partitioned indexes (see Example 9–2 and Example 9–3). Note that the CREATE statement for RESPONSE_RDCM_NFK_ID uses the tablespace but the index storage clause (see "Sample Responses Foreign Key Index Creation Statement" on page 9-9), because this large concatenated index is approximately 70-80% as large as the data segment while RESPONSE_PK_IDX is 40-50% as large.

Example 9–1  Sample Responses Table Creation Statement

```
CREATE TABLE RESPONSES ( 
  RESPONSE_ID                    NUMBER(10,0)         NOT NULL,
  RESPONSE_ENTRY_TS              DATE                 NOT NULL,
  ENTERED_BY                     VARCHAR2(30)         NOT NULL,
  RECEIVED_DCM_ID                NUMBER(10,0)         NOT NULL,
  DCM_QUESTION_ID                NUMBER(10,0)         NOT NULL,
  DCM_QUESTION_GROUP_ID          NUMBER(10,0)         NOT NULL,
  CLINICAL_STUDY_ID              NUMBER(10,0)         NOT NULL,
  REPEAT_SN                      NUMBER(3,0)          NOT NULL,
  END_TS                         DATE                 NOT NULL,
  VALIDATION_STATUS              VARCHAR2(3)          DEFAULT 'NNN' NOT NULL,
  SECOND_PASS_INDICATOR          VARCHAR2(1)          NULL,
  VALUE_TEXT                     VARCHAR2(200)        NULL,
  DISCREPANCY_INDICATOR          VARCHAR2(1)          NULL,
  DATA_CHANGE_REASON_TYPE_CODE   VARCHAR2(15)         NULL,
  DATA_COMMENT_TEXT              VARCHAR2(200)        NULL,
  AUDIT_COMMENT_TEXT             VARCHAR2(200)        NULL,
  EXCEPTION_VALUE_TEXT           VARCHAR2(200)        NULL) 
TABLESPACE RXC_RESP_TSPA 
STORAGE (INITIAL 1M NEXT 1M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) 
PARTITION BY RANGE (CLINICAL_STUDY_ID) ( 
  PARTITION RESP_CSID_LE_1 VALUES LESS THAN (2) 
  STORAGE (INITIAL 512K NEXT 512K MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) 
  , PARTITION RESP_CSID_LE_101 VALUES LESS THAN (102) 
  STORAGE (INITIAL 1M NEXT 1M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) 
  , PARTITION RESP_CSID_LE_1001 VALUES LESS THAN (1002) 
  STORAGE (INITIAL 1M NEXT 1M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) 
  , PARTITION RESP_CSID_LE_1201 VALUES LESS THAN (1202) 
  STORAGE (INITIAL 4M NEXT 4M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) 
  , PARTITION RESP_CSID_LE_10000802 VALUES LESS THAN (10000803) 
  STORAGE (INITIAL 1M NEXT 1M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) 
  , PARTITION RESP_CSID_LE_10001202 VALUES LESS THAN (10001203) 
  ... 
  , PARTITION RESP_CSID_LE_10242201 VALUES LESS THAN (10242202) 
  STORAGE (INITIAL 256K NEXT 256K MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) 
  , PARTITION RESP_CSID_LE_9999999999 VALUES LESS THAN (9999999999) 
  STORAGE (INITIAL 1M NEXT 1M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) 
) ENABLE ROW MOVEMENT;
```
Example 9–2  Sample Responses Primary Key Index Creation Script

CREATE UNIQUE INDEX RESPONSE_PK_IDX ON RESPONSES (RESPONSE_ID, RESPONSE_ENTRY_TS, CLINICAL_STUDY_ID) TABLESPACE RXC_RESP_IDX_TSPA

COMPUTE STATISTICS
NOLOGGING
STORAGE (INITIAL 1M NEXT 1M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
LOCAL {
  PARTITION RESP_CSID_LE_1
  STORAGE (INITIAL 256K NEXT 256K MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  , PARTITION RESP_CSID_LE_101
  STORAGE (INITIAL 512K NEXT 512K MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  , PARTITION RESP_CSID_LE_1001
  STORAGE (INITIAL 512K NEXT 512K MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  , PARTITION RESP_CSID_LE_1201
  STORAGE (INITIAL 2M NEXT 2M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  ...
  ...
  , PARTITION RESP_CSID_LE_10242201
  STORAGE (INITIAL 128K NEXT 128K MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  , PARTITION RESP_CSID_LE_9999999998
  STORAGE (INITIAL 512K NEXT 512K MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
}

Once these scripts are generated you should review them and manually make any changes that can not be made through the Maintain Partition Mapping Table form. For instance, if you want to modify the percent free default, for the table as a whole or for particular partitions, you must currently do this by editing the generated creation script. The script generation currently ignores the partition-specific tablespace specification, so you must manually amend the scripts to add this after the partition name and before the storage clause if you want to place a partition in a tablespace other than the one specified for the table or index as a whole.

In addition, you should amend the index creation scripts to add the clauses COMPUTE STATISTICS and NOLOGGING at the position indicated in bold in Examples 9–2 and 9–3. COMPUTE STATISTICS allows the cost-based statistics to be computed as the new indexes are created and avoids the time-consuming extra step of separately computing the statistics. NOLOGGING avoids the overhead of writing to the Oracle redo log file, which is unnecessary since the database will be backed up after the upgrade (see "Step 6. Create Indexes on Partitioned Responses Table" on page 9-12 and "Maintenance of Cost-Based Statistics" on page 9-15).

Example 9–3  Sample Responses Foreign Key Index Creation Statement

CREATE INDEX RESPONSE_RDCM_NFK_IDX ON RESPONSES (CLINICAL_STUDY_ID, RECEIVED_DCM_ID, DCM_QUESTION_GROUP_ID, DCM_QUESTION_ID, END_TS, RESPONSE_ENTRY_TS, REPEAT_SN, RESPONSE_ID, VALUE_TEXT, VALIDATION_STATUS, EXCEPTION_VALUE_TEXT, DATA_COMMENT_TEXT) TABLESPACE RXC_RESP_IDX_TSPA

COMPUTE STATISTICS
NOLOGGING
STORAGE (INITIAL 1M NEXT 1M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0) COMPRESS 6
LOCAL {
  PARTITION RESP_CSID_LE_1
  STORAGE (INITIAL 512K NEXT 512K MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  , PARTITION RESP_CSID_LE_101
  STORAGE (INITIAL 1M NEXT 1M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  , PARTITION RESP_CSID_LE_1001
Partitioning the Responses Table

Once you have prepared the partition table and index creation scripts, you are ready to perform the partition upgrade. The upgrade consists of the following steps:

1. **Back Up the Database.**
2. **Export the Responses Table.**
3. **Prepare the Database:**
   a. **Drop Responses Table**
   b. **Reorganize Tablespaces and System File Space**
   c. **Create Partitioned Responses Table**
4. **Import Responses Data.**
5. **Compute Statistics on the Responses Table.**
6. **Compile Invalid Objects and, Optionally, Restore Database Trigger.**
7. **Back Up the Database** (can be deferred to after Step 8).
8. **Regenerate Procedures and Data Extract Views.**
9. **Back Up the Database.** (Optional, can be deferred to normal backup schedule.)

**Note:** Conduct all SQL activities while connected to the RXC account, from the tools directory for Oracle Clinical, unless otherwise specified.

---

**Step 1. Back Up the Database**

Since the upgrade process involves dropping and recreating the Responses table and since there is a finite chance that the export of the Responses table could be corrupted, Oracle highly recommends that you perform a full database backup before you start the upgrade.

**Step 2. Export the Responses Table**

The migration of data from the non-partitioned Responses table to a partitioned Responses table is done through the exporting of data into a dump file using the Oracle Export utility and then importing the data into the partitioned table using the Oracle Import utility. During this migration process the Responses table is not available to users.
The Export utility reads the data blocks or rows from non-partitioned Responses table and writes into a dump file (or files). It uses a parameter file exp_resp_param.dat which is shipped with Oracle Clinical and located in the tools directory. The DBA in charge of migration at your location should review and make any changes to the parameters they feel are required.

Please note that by default the export includes grants on the Responses table and that the import restores these grants. The upgrade process relies upon this to restore the grants on the newly reconstructed Responses table, so you should not modify the export or import to exclude grants.

exp rxc/password parfile=exp_resp_param.dat

The exp_resp_param.dat file contains the following information:

FILE=responses1.dmp,responses2.dmp,responses3.dmp,responses4.dmp
FILESIZE=1G
DIRECT=Y
TABLES=(RXC.RESPONSES)
COMPRESS=y
CONSTRAINTS=N
INDEXES=N
LOG=exp_resp.log
#RECORDLENGTH=64K
STATISTICS=none

**FILE** The names of the files to create. Use multiple files, as shown, but extend or shorten this list to handle the maximum estimated size of your exported database.

**FILESIZE** As a rough guideline, test exports of a multi-gigabyte Responses table resulted in a total export size approximately 130% of the size of the unpartitioned Responses table’s data segment. For full information on Export and its parameters, see the Oracle Utilities manual.

### Step 3. Prepare the Database

Prior to creating the new Responses table, you should consider restarting the database specifying NOARCHIVELOG. Since you will back up the database at the completion of the upgrade and it is easier to restart a failed upgrade than to attempt to recover it from redo logs, there is no reason to pay the overhead of archiving the redo logs.

**Drop Responses Table**

Drop the existing, unpartitioned Responses table and its indexes:

SQL> drop table responses;

**Reorganize Tablespaces and System File Space**

If you are changing the use of tablespaces, for instance, to allocate different partitions to different tablespaces, you might want to drop the existing RXC_RESP_TSPA and RXC_RESP_IDX_TSPA and associated database files and recreate the new tablespace structure for the partitioned response table and indexes. This is not required, however.

**Create Partitioned Responses Table**

Create an empty partitioned Responses table by executing the table creation script generated earlier (see Example 9–1, "Sample Responses Table Creation Statement" on page 9-8). Do not create the indexes.

SQL> @part_table_ddl.sql
Step 4. Import Responses Data

The Import utility reads the dump file (or files) created by the export step and populates the partitioned Responses table. It uses a parameter file imp_resp_param.dat which is shipped with Oracle Clinical and located in the tools directory. The DBA in charge of migration at your location should review and make any changes to the parameters which they feel are required.

imp rxc/password PARFILE=imp_resp_param.dat

The imp_resp_param.dat file contains the following information:

- FILE=responses1.dmp, responses2.dmp, responses3.dmp
- FILESIZE=1G
- IGNORE=Y
- ANALYZE=N
- #BUFFER
- COMMIT=Y
- TABLES=RESPONSES
- INDEXES=N
- LOG=imp_resp.log
- #RECORDLENGTH
- ANALYZE=N

FILE The names of the files to create. Make sure you edit this list to match the files created by the Export.

For full information on Import and its parameters, see the Oracle Utilities manual.

Step 5. Compute Statistics on the Responses Table

Once the import completes successfully, you should compute the statistics that are used by the cost-based optimizer by executing the ANALYZE command:

SQL> analyze table responses compute statistics;

While you can perform the analyze step with a small sample size, as illustrated below, the time saved here is relatively small. Because the indexes do not yet exist, the entire ANALYZE is performed by full scans of each partition, which are very efficient.

SQL> analyze table responses estimate statistics sample 5 percent;


Step 6. Create Indexes on Partitioned Responses Table

Before creating the indexes on the Responses table, amend the index creation script to add the clauses COMPUTE STATISTICS and NOLOGGING, as shown in Examples 9–2 and 9–3, above.

Create the indexes by executing the index creation scripts generated and edited earlier (see "Generating Table and Index Creation SQL" on page 9-8).

SQL> @part_index_ddl.sql

These scripts take significant time to execute, proportionate to the size of the Responses table. You can monitor progress by querying the USER_SEGMENTS dictionary view from the RXC account because each index partition is committed separately.
Step 7. Compile Invalid Objects and, Optionally, Restore Database Trigger

Dropping and recreating Responses invalidates some packages and views. Execute the script compile_all_invalid.sql to compile all invalid objects in the database.

To run the compile_all_invalid.sql script:
1. Set up the environment to point to the correct database and code environment.
2. cd $RXC_INSTALL
3. Open a SQL*Plus session as system.
4. Run compile_all_invalid.sql:
   
   SQL> start compile_all_invalid.sql

If you use pre-version 3.1 procedures and replicate study data using a database trigger on the Responses table to ensure that certain data changes occurring during batch validation are replicated, then execute the script resptrig.sql in the install directory.

   SQL> @resptrig.sql

Step 8. Back Up the Database

After the upgrade is complete, perform your normal full database backup. Re-enable redo log archiving before releasing Oracle Clinical for production use. This backup is mandatory because logging was disabled during the index rebuild and archive logging disabled during the import. You can wait until after Step 8, if you do that immediately after Step 6 and before production use of the database.

Step 9. Regenerate Procedures and Data Extract Views

Once the Response table is partitioned, you must regenerate all procedures and data extract views for active studies. If you have disabled the generation of Version 4-style procedure and view optimization (see "Enforcing Pre-Version 4.0 Optimization" on page 9-3), you must enable it before regenerating procedures and views.

To regenerate procedures and views, follow the instructions in Chapter 10, "Utilities". For procedure regeneration you should select the parameters FULL, GENERATE and ALL. Note that in Releases 4.0 and above, procedure regeneration with GENERATE no longer causes discrepancies to be closed and reopened.

Step 10. Back Up the Database

If you performed Step 7, you can defer this backup to your normal schedule.

If you skipped Step 7, do a full backup now. Remember to re-enable redo log archiving before releasing Oracle Clinical for production use.

Partition Maintenance

This section includes the following topics:

- Prospective Allocation of clinical_study_ids for Partitioning
- Strategy for Ongoing Partition Maintenance
- Instructions for Partition Maintenance
- Rebuild Indexes after Partition Maintenance
Using Read-Only Partitions to Minimize Backup

Prospective Allocation of clinical_study_ids for Partitioning
1. Ensure that distributed studies do not intersperse clinical_study_ids arbitrarily.
   If you intend to collapse studies into single partitions or to consolidate dormant studies into read-only partitions, you must ensure that new studies do not get created whose clinical_study_ids lie between existing studies. This can occur in distributed installations of Oracle Clinical when studies are created at multiple locations using database seeds that differ only in their trailing digits. To address this you should either share clinical_study_id sequence across locations by replicating OCL_STUDIES and creating studies centrally or, alternatively, you should ensure that sequences do not overlap by seeding at non-overlapping starting seed values, not just separate trailing digits. For instance, start location 1 at id 101 and location 2 at id 1000102 rather than 102.
2. Preallocate planned studies to achieve grouping of small studies.
   If you want to consolidate small studies in single partitions, you should make sure that they have contiguous ranges. For instance, for a new project, create placeholder planned studies for all phase 1 and phase 2 studies so that they have contiguous ids. This allows them to be grouped into a single partition.

Strategy for Ongoing Partition Maintenance
There are three types of partition maintenance activities:
- Activities that rebuild partitions
- Activities that split or merge partitions
- Activities that change ongoing characteristics of partitions
Activities that rebuild partitions must be performed manually. These include rebuilding the indexes on a partition, moving the data in a partition to a different tablespace, and restructuring the data in a partition to reduce the number of extents or change the PCTFREE storage parameter. See the Oracle documentation for instructions on how to carry out these activities.

Oracle Clinical includes the partition mapping form (see "Maintain Partition Mapping Tables Window" on page 9-7) and a utility script generator that support SPLIT and MERGE operations and changes to ongoing storage characteristics.

Note that all SQL activities are carried out while connected to the RXC account.

Rebuild Partitions Periodically, at weekly or monthly intervals depending on growth rate, review the current partition growth by querying the USER_SEGMENTS view:

```
SQL> select segment_name, partition_name, extents
SQL> from user_segments
SQL> where extents > 20
SQL> and segment_name like 'RESPONSE%';
```

This shows you any partitions that might need to be rebuilt.
In addition, you might want to initiate a manual process for identifying studies that are about to enter periods of intense reporting. The partitions for these studies could then be rebuilt, both to coalesce index space by rebuilding the indexes and, perhaps, to decrease the PCTFREE storage parameter if data is no longer changing.

See Example 9–6 for instructions on rebuilding partition indexes.
Managing New Studies Unless there is a study that will grow extremely rapidly, for instance, due to batch loading, there is no need to anticipate the creation of studies and preallocate their partitions. Since new studies are created with incrementally higher clinical_study_ids, they are automatically added to the maximum partition. As long as this partition has a storage clause with large enough space allocations, these studies can start there without significant performance impact. At planned intervals you should review this partition to determine how these studies should be partitioned and use the partition maintenance script ("Instructions for Partition Maintenance" on page 9-16) to split out the new partitions.

Use the script list_study_resp_cnt_part.sql to obtain a count by study of studies in a particular partition:

```
list_study_resp_cnt_part.sql partition_name spool_file_name
```

The following example lists the studies in the last partition, that is, the new ones:

```
SQL>@list_study_resp_cnt_part.sql RESP_CSID_LE_9999999998 new_studies.lis
```

Reviewing for Partitions that Need Splitting or Merging As with new studies, existing partitions that contain multiple studies should be periodically reviewed and split as needed.

You may also want to merge partitions for studies that become inactive or that did not grow as planned. In particular, you might want to use the approach of using read-only tablespaces for inactive studies discussed in "Using Read-Only Partitions to Minimize Backup" on page 9-17.

Maintenance of Cost-Based Statistics Whenever a partition is merged or split, or when the data volume in a partition has changed significantly, the cost-based statistics for that partition need to be refreshed. The command to refresh the statistics for a partition is:

```
ANALYZE TABLE Responses PARTITION (partition_name) COMPUTE STATISTICS;
```

If the partition is large, you can use the statistics estimation with a small sample size. However, since you do not need to analyze the whole table, but just partitions with changes, this may no longer be necessary:

```
SQL> analyze table responses partition (partition_name)
```

One approach to maintaining the statistics is to use a table to hold the partition statistics from the previous partition maintenance and then drive the creation of the analyze statements from that table and the current statistics. The code fragment in Example 9–4 illustrates this approach. It triggers the ANALYZE when the data volume in a partition increases by more than 50%.

**Example 9–4 Using a Table to Compute Statistics**

```
CREATE TABLE resp_part AS
SELECT partition_name, bytes
FROM user_segments
WHERE segment_name = 'RESPONSES';
spool analyze_responses.sql
SELECT 'ANALYZE TABLE RESPONSES PARTITION ('||
u.partition_name||') COMPUTE STATISTICS'
FROM resp_part r, user_segments u
WHERE u.segment_name = 'RESPONSES'
AND u.partition_name = r.partition_name
```
Partitioning the Responses Table

AND u.bytes/1.5 > r.bytes
UNION ALL
SELECT 'ANALYZE TABLE RESPONSES PARTITION ('||
  u.partition_name||') COMPUTE STATISTICS'
FROM user_segments u
WHERE u.segment_name = 'RESPONSES'
AND not exists (select null from resp_part r
  where r.partition_name = u.partition_name)

Instructions for Partition Maintenance

When you determine that you need to merge or split partitions, first use the Maintain Partition Mapping form (see Figure 9–2 on page 9-7). Insert records for partitions that are to be split or delete records for partitions that are to be merged. You can also alter the storage clauses for partitions, although this affects only new storage allocations — it does not rebuild the existing partition space.

You then use the utility script gen_alter_partition.sql to generate the SQL partition maintenance commands.

gen_alter_partition output_file.sql

For example:

SQL> $gen_alter_partition.sql alter.sql

The generation script compares the partition definition in the Oracle Clinical Partition Mapping table with the actual partition structure in the database. It then generates the SQL DDL statements to merge or split the partitions and to alter the storage clauses (see Example 9–5).

You should review the generated script and make any necessary modifications before you run it.

Example 9–5  Sample Output from the Partition Maintenance Script

REM Split case 4: Split RESP_CSID_LE_10002201 into RESP_CSID_LE_10001901 at 10001902
ALTER TABLE responses SPLIT PARTITION RESP_CSID_LE_10002201 AT (10001902) INTO
  (PARTITION RESP_CSID_LE_10001901 , PARTITION RESP_CSID_LE_10002201
  STORAGE (INITIAL 4M NEXT 4M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  );
ALTER INDEX RESPONSE_PK_IDX MODIFY PARTITION RESP_CSID_LE_10001901
  STORAGE (NEXT 2M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  ;
ALTER INDEX RESPONSE_RDCM_NFK_IDX MODIFY PARTITION RESP_CSID_LE_10001901
  STORAGE (NEXT 4M MINEXTENTS 1 MAXEXTENTS 99 PCTINCREASE 0)
  ;
REM Merge case 2: Merge RESP_CSID_LE_10001102 into RESP_CSID_LE_10001202
ALTER TABLE responses MERGE PARTITIONS RESP_CSID_LE_10001102,
  RESP_CSID_LE_10001202 INTO PARTITION
  RESP_CSID_LE_10001202;
REM Merge case 2: Merge RESP_CSID_LE_10001101 into RESP_CSID_LE_10001202
ALTER TABLE responses MERGE PARTITIONS RESP_CSID_LE_10001101,
  RESP_CSID_LE_10001202 INTO PARTITION
  RESP_CSID_LE_10001202;
Rebuild Indexes after Partition Maintenance

After you run the partition maintenance script (see Example 9–5), some of the associated index partitions may be unusable. Use the SQL code shown in Example 9–6 to detect the affected partitions. It generates a script (rebuild.sql) that you run to rebuild the index partitions.

Example 9–6  SQL Code for Rebuilding Index Partitions

```sql
set pagesize 1000
set verify off
set feedback off
set heading off
spool rebuild.sql
select distinct 'ALTER TABLE RESPONSES MODIFY PARTITION ' || partition_name || ' REBUILD UNUSABLE LOCAL INDEXES /
' from all_ind_partitions
where index_name in ('RESPONSE_PK_IDX', 'RESPONSE_RDCM_NFK_IDX')
and status = 'UNUSABLE'
/
spool off
set verify on
set feedback on
set heading on
```

Using Read-Only Partitions to Minimize Backup

Oracle enables you to designate tablespaces as read-only. If a tablespace is designated read-only, any attempt to modify data in that tablespace causes an Oracle error. Since the tablespace can not be modified, it can be created using data files on a separate physical device that only needs to be backed-up after maintenance activity during which the tablespace is modifiable. Over time, a significant portion of the Responses table data volume could be migrated to such read-only tablespaces, dramatically reducing the data volume that needs to be backed up by nightly backups.

To implement this strategy, you would set up the tablespace using data files on a separate physical device. A manual process of identifying studies that will no longer have modifications must be implemented. At periodic intervals, quarterly or semi-annually, you would perform partition maintenance to rebuild the partitions for those studies. During the maintenance, you alter the tablespace to make it writable. You move the data by exporting the partition, dropping the partition, and recreating the partition using the new tablespace. Once all studies have been moved, you alter the tablespace to be read-only once again and perform a backup of the device.

Upgrading Indexes

Oracle Clinical has been retuned to require only two indexes on the Responses table:

- **RESPONSE_PK_IDX** – primary key index on RESPONSE_ID and RESPONSE_ENTRY_TS
- **RESPONSE_RDCM_NFK_IDX** – an expanded, concatenated index on RECEIVED_DCM_ID

The following indexes are no longer required:

- **RESPONSE_CS_NFK_IDX** – on CLINICAL_STUDY_ID
- **RESPONSE_DCMQG_NFK_IDX** – on DCM_QUESTION_GROUP_ID
RESPONSE_UK_IDX – on DCM_QUESTION_ID

This, combined with the use of Oracle database leading key compression, results in up to 50% savings in space allocated to the Responses table and its indexes, and a significant reduction in runtime overhead to maintain the indexes. These savings occur whether you partition or not.

Index Upgrade Process

To upgrade the indexes you must modify and execute the script rebuild_resp_index_non_part.sql in the install directory while connected to the RXC account. You should modify the script to provide storage parameters appropriate for the size of your Responses table. The foreign key index is approximately 70% the size of the Responses table when created and the primary key index is approximately 40%. In addition, if you are dropping the obsolete indexes, you should un-comment the drop index statements and move them to before the index creation statements to ensure that their space is freed before the new indexes are created.

SQL> @rebuild_resp_index_non_part.sql

Resizing the Storage Clause for Indexes

If you upgraded without resizing (redefining) the storage clause of indexes using ocl46indexchg.sql, you can do it at any time by manually running the script. Modifying these indexes improves performance in queries executed from the RDC application.

To run ocl46indexchg.sql in UNIX, from the command line, enter:

    opa_setup database 46
    cd $RXC_INSTALL
    sqlplus /nolog @ocl46indexchg.sql

To run ocl46indexchg.sql in Windows, from the command line, enter:

    set p1=database
    set p2=46
    opa_setup
    cd %RXC_INSTALL%
    sqlplus /nolog @ocl46indexchg.sql

___

**Note:** Resizing the storage clause and running the script is an optional step in every upgrade path, including from 4.5.2 to 4.5.3, so you may not have done it yet.

Response Index Upgrade

If you have other applications that directly access the Responses table, these may need to be retuned before you perform the index upgrade (due to the new leading CLINICAL_STUDY_ID key on the RESPONSE_RDCM_NFK_IDX) or before you drop the optional indexes (see "Query Tuning Guidelines" on page 9-20). If you are not partitioning you can retain one or more of these indexes until your other applications are retuned.
If you are partitioning and want to retain one or more of the old indexes, you have to manually recreate them as non-partitioned global indexes after the partitioning upgrade.

Before upgrading the indexes, you must analyze your disk space requirements if you are preserving the existing indexes. Due to leading key compression the new, concatenated RESPONSE_RDCM_NFK_IDX index is significantly smaller than the version 3.1 concatenated RESPONSE_DCMQG_NFK_IDX it functionally replaces. However, if you are preserving all pre-existing indexes, there is a small (20-30%) net increase in the size of the pre-existing RESPONSE_RDCM_NFK_IDX. Dropping any of the pre-existing indexes, including the largely superfluous RESPONSE_CS_NFK_IDX, releases at least that amount of space.

Even if you are not preserving existing indexes, since you will be dropping and recreating all of the indexes, the index upgrade is a good time to replan the use of physical storage for the Response table indexes. Placing the indexes on a tablespace on a separate physical device from the Responses table is a good practice.

**Regenerate Procedures and Data Extract Views**

Once the Response table indexes are rebuilt, you must regenerate all procedures and data extract views for active studies. If you have disabled the generation of Version 4-style procedure and view optimization (see "Enforcing Pre-Version 4.0 Optimization" on page 9-3), you must enable it before regenerating procedures and views.

To regenerate procedures and views, follow the instructions for using the procedure and view regeneration utilities documented in Chapter 10, "Utilities". For procedure regeneration you should select the parameters FULL, GENERATE and ALL.

**ResponsesT (Test Database) Index Upgrade**

A script, rebuild_respt_index.sql in the install directory, rebuilds the indexes on the ResponsesT, or test database Responses table. This script should be run whether or not you are partitioning the Responses table. The upgrade for ResponsesT should be done at the same time as either partitioning or the Responses table index upgrade.

**RECEIVED_DCI and RECEIVED_DCM Index Upgrades**

To take advantage of the Oracle index compression feature, you can optionally rebuild selected indexes on the RECEIVED_DCMs and RECEIVED_DCIs tables. Since these can be large tables, this script is not automatically run as part of the upgrade to Oracle Clinical V4. The upgrade of the RECEIVED_DCMs and RECEIVED_DCIs indexes can be done at any time that is convenient, independently of either partitioning or Responses table index upgrade. This change reduces the size of these indexes and, therefore, improves performance due to fewer disk accesses.

To rebuild the indexes on the RECEIVED_DCMs and RECEIVED_DCIs tables, execute the script oclupg32to40opt5.sql in the install directory:

```
SQL> @oclupg32to40opt5.sql
```
Query Tuning Guidelines

Query Tuning Guidelines

For both partitioning and index upgrade, Oracle Clinical has been retuned to use only the primary key index RESPONSES_PK_IDX with the Response ID leading key or the concatenated RESPONSES_RDCM_NFK_IDX with the clinical_study_id and RECEIVED_DCM_ID leading keys. You should review any applications that directly access the Responses table to ensure that they are correctly optimized for the new index structure.

The basic guidelines for re-optimization are:

1. Include clinical_study_id in all access to the Responses table.

   There are two reasons for this. Primarily, because Responses are in a partition based on clinical_study_id, the query optimizer can restrict its search to the proper partition if the query contains clinical_study_id. This is called 'Partition Pruning'. In order of preference this reference to clinical_study_id can be a constant, a bind variable in an equi-join, or a join from another table.

   Secondly, the concatenated index is prefixed with clinical_study_id in order to force Responses from different studies in the same partition to be physically grouped together and to optimize certain partition accesses.

2. Redirect all previous queries on DCM_QUESTION_ID or DCM_QUESTION_GROUP_ID to use a join through RECEIVED_DCMs so that they can use the concatenated index.

   Since there are no longer indexes with DCM_QUESTION_ID or DCM_QUESTION_GROUP_ID as leading keys, these are no longer efficient access paths. Much of the access involving these keys is already done in the context of a RECEIVED_DCM, so the query tuning is usually minimal. In some cases, it might be necessary to add joins to DCM_QUESTIONS or DCM_QUESTION_GROUPS, then through RECEIVED_DCMs via DCM_ID. For instance, the query shown in Examples 9–7 and 9–8 selects responses to a particular Question where the patient position is owned locally.

Example 9–7  Query Before Redirection

```
SELECT response_id, to_char(response_entry_ts, 'DD-MON-YYYY HH24:MI:SS'),
       received_dcm_id
FROM responses r
WHERE r.dcm_question_id = :dcm_question_id
  AND EXISTS
    (SELECT NULL FROM received_dcms rd, patient_positions papo
     WHERE rd.patient_position_id = papo.patient_position_id
       AND papo.owning_location = :current_location
       AND rd.received_dcm_id = r.received_dcm_id)
```

Example 9–8  Query After Redirection

```
SELECT response_id, to_char(response_entry_ts, 'DD-MON-YYYY HH24:MI:SS'),
       received_dcm_id
FROM responses r
WHERE r.dcm_question_id = :dcm_question_id
  AND
    (clinical_study_id, received_dcm_id, dcm_question_group_id) IN
    (SELECT rd.clinical_study_id, rd.received_dcm_id, dq.dcm_question_group_id
     FROM received_dcms rd, patient_positions papo, dcm_questions dq
     WHERE dq.dcm_question_id = :dcm_question_id
       AND dq.dcm_que_dcm_subset_sn = 1
       AND dq.dcm_que_dcm_layout_sn = 1
```
AND rd.dcm_id = dq.dcm_id
AND rd.patient_position_id = papo.patient_position_id
AND papo.owning_location = :current_location)
Oracle Clinical provides a set of utilities for performing tasks that are easier to accomplish from a command line or that cannot be done from within the application. These utilities are described in this section. The activities covered by these utilities include:

- **Computing the Validation Status of All Responses** on page 10-1
- **Generating Validation Procedures** on page 10-2
- **Deleting Inactive Procedures** on page 10-4

Information on using data extract within the Oracle Clinical application is available in the *Oracle Clinical Creating a Study* and *Oracle Clinical Conducting a Study* manuals. Information on Procedures is also in *Oracle Clinical Creating a Study*.

### Computing the Validation Status of All Responses

Use the cnvstatus utility to compute a validation status for all responses. The utility populates a column in the RESPONSES table that contains the validation status of each stored response. Before populating the response field VALIDATION_STATUS, you might want to add Discrepancy Resolution subtypes to distinguish various types of resolutions. You do this by entering values in the Long Value field of the reference codelist DISCREPANCY RESOLU TYPE CODE. This is an installation codelist you access from within Oracle Clinical which maintains user-defined discrepancy statuses.

You must select the values from the following list: NULL, CONFIRMED, IRRESOLVABLE, SUPERSEDED, or NOT DISCREPANT. The last two values are used only for manual discrepancies; they indicate that the discrepancy applied to a previous version of the response, or that the discrepancy was never really a problem with the data, but just a comment.

When the process is complete, examine the log, $RXC_LOG/cnvstatus.log, for errors.

### Running cnvstatus on UNIX Platforms

To run cnvstatus on a UNIX platform:

1. Log on to the server in your user account and change the directory to $RXC_TOOLS.
2. Set the environment:

<table>
<thead>
<tr>
<th>Shell</th>
<th>Command Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Shell</td>
<td>opa_setup db_name code_env</td>
</tr>
</tbody>
</table>
Generating Validation Procedures

With the gen_procs utility you can convert existing Validation Procedures to 3.1-style, and regenerate them, on a per-study basis. Its use is not required, or necessarily recommended, for upgrades or new installations of Oracle Clinical.

This utility has the following syntax:

```
gen_procs { ALL | study_name } { FULL | INC } { CONVERT | GENERATE | PARSE } { 31 | 30 | ALL }
```

<table>
<thead>
<tr>
<th>Shell</th>
<th>Command Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bourne</td>
<td><code>p1 = db_name</code></td>
</tr>
<tr>
<td></td>
<td><code>p2 = code_env</code></td>
</tr>
<tr>
<td></td>
<td><code>. opa_setup</code></td>
</tr>
</tbody>
</table>

where `db_name` is a database instance name and `code_env` is a code environment designation.

3. Set the output directory:

<table>
<thead>
<tr>
<th>Shell</th>
<th>Command Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bourne</td>
<td><code>RXC_LOG=usr_log_dir</code></td>
</tr>
<tr>
<td></td>
<td><code>export code_env</code></td>
</tr>
</tbody>
</table>

4. Run the script. For example:

```
% cnvstatus ALL | Study_Name
```

Where "ALL" is all studies in the database and `Study_Name` is the name of one study.

Running cnvstatus on Windows

To run cnvstatus on Windows:

1. Log on to the server using your local account.
2. Open a DOS window, change directory to `%RXC_TOOLS`, and set the server environment:

```
set p1=db_name
set p2=code_env
opa_setup
```
3. Set the output directory:

```
set rxc_log=user_log_folder
```
4. Run the command file. For example:

```
cnvstatus ALL | Study_Name
```

Where "ALL" is all studies in the database and `Study_Name` is the name of one study.
Choose one option from each set of qualifiers:

**ALL | study_name** — specifies the study you want to apply to. Enter either an individual study name or ALL to include all studies. This qualifier is not case-sensitive.

**FULL | INC** — specifies whether to perform full or incremental replication. You select FULL when running from the command line. INC is used when replication runs this command. This qualifier is case-sensitive.

**CONVERT | GENERATE | PARSE** — specifies the action you want to take. CONVERT works only for pre-3.1 Procedures; it converts the Procedure to 3.1-style, as well as generating and parsing as part of processing. PARSE works only with 3.1-style Procedures; it parses and recreates the package. GENERATE works for 3.0, 3.1, or ALL procedures; it also parses each package. PARSE and GENERATE are used primarily when the utility is called for replication.

**31 | 30 | ALL** — specifies the version of the Procedures to process.

The system creates a file named ora_errors.err in the user’s RXC_LOG directory.

**Running gen_procs on UNIX Systems**

To run gen_procs on a UNIX platform, follow these steps:

1. Log on to the server as the opapps user or a user who has write permission.
2. Set the environment:
   
   ```
   opa_setup db_name code_env
   ```
   
   where `db_name` is a database instance name and `code_env` is a code environment designation.
3. Change the directory to `$RXC_TOOLS`.
4. Set the output directory:
   
   ```
   setenv RXC_LOG usr_log_dir
   ```
5. Run the script. For example:
   
   ```
   gen_procs ALL FULL GENERATE ALL > gen_procs.log
   ```

Oracle Clinical creates `gen_procs.log` in `$RXC_TOOLS`. Files `ora_errors.err` and `genprocs.log` are created in the RXC_LOG directory.

**Running gen_procs on Windows Systems**

To run gen_procs on Windows, follow these steps:

1. Log on to the server using your local account.
2. Open a DOS window and set the server environment. Enter:
   
   ```
   set p1=db_name
   set p2=code_env
   opa_setup
   cd /d %RXC_BIN%
   ```
3. Set the output directory:
   
   ```
   set RXC_LOG=user_log_folder
   ```
4. Run the command file. For example:

   gen_procs ALL FULL GENERATE ALL > gen_procs.log

   Oracle Clinical creates genprocs.log in the current directory (%RXC_BIN%). Files
   ora_errors.err and genprocs.log are created in the RXC_LOG directory.

Deleting Inactive Procedures

Oracle Clinical lets you delete unneeded Procedures from within the application. However, this does not actually delete the database packages that contain the Procedures, which may cause unwanted Procedures to accumulate. To delete them, go
to the RXC_TOOLS directory, log in as RXC_PD, and execute the SQL script
rxcdelproc.
You need to set up file viewing in Oracle Clinical so that users can:

- View report outputs and log files
- View log files from Parameterized Submission (PSUB) batch jobs
- View HTML previews of DCI Forms Layout definitions

This section contains the following topics:

- File Viewing Setup Overview on page 11-1
- Editing formsweb.cfg to Change File Viewing Protocol on page 11-2
- Setting Directory Mappings on page 11-2
- Setting Up Image Viewing for DCI and DCM Form Layouts HTML Preview Feature on page 11-3

**File Viewing Setup Overview**

To set up file viewing:

1. Install the Oracle RDBMS and specify the file viewing protocol you want to use: FTP, HTTP, HTTPS, or UNC; instructions are in the *Oracle Clinical Installation Guide*.
   - If your system uses a UNIX Database Server, we recommend that you choose FTP file viewing protocol. However, if your administration policies do not permit FTP use in your system, we recommend you choose HTTP.
   - If your system uses a Windows Database Server, we recommend that you choose the UNC file viewing protocol.

2. Complete the server setup required for the protocol you use:
   - For UNC, set up a shared directory on the database server that is accessible from your WAN.
   - For FTP, set up an FTP server on your database server.
   - For HTTP, setup an HTTP server on the database server.

3. Install the Oracle Clinical Reports Server and create a Reports Server root directory; instructions are in the *Oracle Clinical Installation Guide*.

4. Create one or more log directories for report outputs; see "Setting Up a Report Server Log Directory" on page 1-2.
5. Create one or more log directories for batch job (PSUB) outputs; see "Creating a PSUB Log Directory" on page 1-3.

For the UNC or HTTP file viewing protocol, setup the correct permissions. Refer to "Creating a PSUB Log Directory" on page 1-3.


7. If you use a different file viewing protocol than the default for the operating system, modify the formsweb.cfg file to match the type of file viewing protocol you use. See "Editing formsweb.cfg to Change File Viewing Protocol" on page 11-2 for information.

**Editing formsweb.cfg to Change File Viewing Protocol**

If you use a different file viewing protocol than the default for the operating system, modify the formsweb.cfg file to match the type of file viewing protocol you use.

This file is located in the `ORACLE_AS10GR2MT_HOME/forms/server/` directory.

For example, to change configure the application server for a change from FTP to HTTP file viewing protocol, modify the `opa_file_viewing` parameter in formsweb.cfg from:

```
opa_file_viewing=PSUBMAP=FTP RSMAP=UNC
```

to:

```
opa_file_viewing=PSUBMAP=HTTP RSMAP=UNC
```

**Note:** The RSMAP setting does not affect file viewing.

**Setting Directory Mappings**

Use this form to map the PSUB log directory specified for each user to a format that can be used to enable the user to view PSUB output files on the database server. This tasks is required only if you are using HTTP or HTTPS. It is not necessary if you are using UNC or FTP.

To map file viewing specifications:

1. Navigate to Admin, then Directory Mappings, then Directory Mappings. The system opens the Maintain Directory Mappings window.

2. Execute Query (F8), to see all records.

3. In the Mapping Code field select the file viewing protocol you are using on the database server.

   (You can add or remove values for this list of values in the MAPPING_TYPE installation reference codelist.)

4. In the Original Directory field, enter the root directory portion of the log directory for PSUB or the Reports Server, using the appropriate syntax for the operating system. See Table 11–1, "UNIX Directory Mapping Example" and Table 11–2, "Windows Directory Mapping Example".
5. In the **Mapped Directory** field, enter the UNC or the URL for the PSUB root directory. See Table 11–1, "UNIX Directory Mapping Example" and Table 11–2, "Windows Directory Mapping Example".

### Table 11–1  UNIX Directory Mapping Example

<table>
<thead>
<tr>
<th>Mapping Code</th>
<th>Original Directory</th>
<th>Mapped Directory</th>
<th>Where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>user01/home/logs</td>
<td><a href="http://www.pharma.com/oc_output">http://www.pharma.com/oc_output</a></td>
<td>shared directory named oc_output</td>
</tr>
</tbody>
</table>

### Table 11–2  Windows Directory Mapping Example

<table>
<thead>
<tr>
<th>Mapping Code</th>
<th>Original Directory</th>
<th>Mapped Directory</th>
<th>Where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>E:\opa\users</td>
<td><a href="http://www.pharma.com/oc_output">http://www.pharma.com/oc_output</a></td>
<td>shared directory name oc_output</td>
</tr>
</tbody>
</table>

### Setting Up Image Viewing for DCI and DCM Form Layouts HTML Preview Feature

If your CRFs contain a company logo or other images and you want to be able to view the images using the HTML Preview feature of the Oracle Clinical DCI Forms Layout Editor, configure the J2EE application used in file viewing as follows:

1. Open the Windows Explorer on middle tier and navigate to `ORACLE_AS10GR2MT_HOME\j2ee\opa\application-deployments\ocrdcclassic\ocrdcclassic`

2. Take a backup of the file `orion-web.xml`.

3. Open the original version in WordPad or a text editor.

4. Insert the following lines.

    ```xml
    <virtual-directory virtual-path="/crfimages" real-path="/Drive:/<image folder path>" />
    ```

    For example, if your images folder is at `C:\opapps46\html\rdc\dcif_images`, the file would look like:

    ```xml
    <?xml version="1.0"?>
    <orion-web-app deployment-version="10.1.2.3.0"
        jsp-cache-directory="./persistence"
        jsp-cache-tlds="on"
        temporary-directory="./temp">
        <!-- Uncomment this element to control web application class loader behavior. -->
        <web-app-class-loader search-local-classes-first="true"
            include-war-manifest-class-path="true" />
        -->
        <virtual-directory virtual-path="/crfimages"
            real-path="/C:\opapps46\html\rdc\dcif_images" />
        <web-app></web-app>
    </orion-web-app>
    ```
Note:  *real-path* can be a network location. If it is, make sure that the Oracle Process Manager Service on the middle tier is started up as a user with access to the specified network location; see "Setting Up a Network User to Run the Oracle Process Manager Service" on page 12-2.

If you have configured your Oracle Clinical and RDC installations to use a single repository for all images (see "Setting Up a Central Image Repository" on page 12-1), make sure that the path is to that location.

5. Save and close the file.

6. Log in as the network account you set up to start the Oracle Process Manager Service.

7. Restart the RDC OC4J instance.
This section contains the following topics:

- **Setting Up a Central Image Repository** on page 12-1—applies to RDC data entry and to Oracle Clinical DCI and DCM layout HTML previews
- **Copying Image Files to All Installations** on page 12-3—applies to RDC data entry only

To enable image viewing on CRFs during RDC data entry, you can choose either to set up a central repository or to copy images to all RDC Onsite installations.

### Setting Up a Central Image Repository

You can configure your OC and RDC installations to use a single repository for all images. After the initial setup, this approach is more efficient because edited and added images do not need to be copied to all installations.

There are also disadvantages:

- If the share location goes down for any reason, none of the links to the shared directory will work. The applications should continue to run, but the images will not appear on the data entry CRF or in the layout editor HTML preview.
- If there is considerable network latency between the Oracle Clinical and RDC installations and the share location, performance may suffer.

To set up a single repository:

1. Create a shared directory on a computer that is accessible from all Oracle Clinical and RDC Onsite installations, and put the image files in the shared directory.

2. To use the central image repository in Oracle Clinical, add a line to orion-web.xml at each Oracle Clinical installation for virtual folder mapping to the central repository. See "Setting Up Image Viewing for DCI and DCM Form Layouts HTML Preview Feature" on page 11-3.

3. To use the central image repository in RDC Onsite during data entry, add a line to orion-web.xml at each RDC Onsite installation for virtual folder mapping to the central repository. See "Setting Up Image Viewing During Data Entry" on page 12-2.

4. Repeat Step 2 on every Oracle Clinical installation and repeat Step 3 on every RDC Onsite installation.
Setting Up a Network User to Run the Oracle Process Manager Service

On each Oracle Clinical and RDC installation, the Oracle Process Manager Service must be started up as a network user to allow access to shared image files on a network location.

1. Open Windows Service Manager.
2. Select Oracle AS Process Manager.
3. Right-click and select Properties.
4. Go to the Log On tab.
5. Select This Account.
6. Enter username/password of the network account and click OK.
7. Reboot the computer.

Setting Up Image Viewing During Data Entry

If your CRFs contain a company logo or other images and you want users to be able to see during data entry in RDC, configure the J2EE application used in file viewing as follows:

1. Open the Windows Explorer on middle tier and navigate to ORACLE_AS10GR2MT_HOME\j2ee\rdc\application-deployments\olsardc\rdconsite\orion-web.xml
2. Take a backup of the file orion-web.xml.
3. Open the original version in WordPad or a text editor.
4. Insert the following lines.
   `<virtual-directory virtual-path="/de/crfimages" real-path="<shared folder path>">
   </virtual-directory>`

   For example, if your images folder is at \sharemachine\images, the file would look like:

   `<orion-web-app deployment-version="10.1.2.3.0" 
   jsp-cache-directory="./persistence" 
   jsp-cache-tlds="on" 
   temporary-directory="./temp"> 
   <virtual-directory virtual-path="/de/crfimages" 
   real-path="\sharemachine\images" /> 
   <!-- Uncomment this element to control web application class loader behavior. 
   <web-app-class-loader search-local-classes-first="true" 
   include-war-manifest-class-path="true" /> 
   --> 
   </web-app></orion-web-app>

5. Save and close the file.
6. Log in as the network account you set up to start the Oracle Process Manager Service.
7. Restart the RDC OC4J instance.
Copying Image Files to All Installations

You can create an images directory at every RDC installation and copy all image files to each installation. If the images are updated or if new ones are added, you must copy them to all installations.

Create an images directory at the following location at each installation:

`ORACLE_AS10GR2MT_HOME\j2ee\rdc\applications\olsarcd\rdconsite\de\crfimages`

**Note:** If you choose to copy files to all RDC installations, do not edit RDC Onsite's `orion-web.xml` file.
This section includes reference information.

- Appendix A, "Environment Variables and Registry Settings"
- Appendix B, "SAS_VIEW Directory Tree"
- Appendix C, "Troubleshooting"
- Appendix D, "Oracle Clinical Tablespaces"
- Appendix E, "Routine Server Administration"
Environment Variables and Registry Settings

This appendix describes the environment variables and registry settings that Oracle Clinical uses to condition the environment on compute servers, and to condition a user’s interactive environment to point at a particular database and code environment. These variables are created as part of the Oracle Clinical installation process; most are also set to a default value. Information is provided here so that you can set them up and modify them, if necessary.

This section discusses and describes the following topics:

- Summary of the Oracle Clinical Setup on page A-1
- Windows Registry Settings on page A-10

Summary of the Oracle Clinical Setup

The Oracle Clinical server environment definition and setup system, known as opa_setup, consists of a set of scripts (on UNIX systems) or command files (on Windows systems).

The four types of setup files are:

- **Definition** – Edit this file, if necessary, to set or change values on a system-wide or database-specific level.
  - opa_settings – UNIX
  - opa_settings.bat – Windows

- **Initialization** (login) – This file is optional, edit it to set or change values for an individual user.
  - .profile – UNIX platforms, Bourne/Korn shells
  - .cshrc – UNIX platforms, C shell
  - .login – UNIX platforms, C shell
  - login.bat – Windows

- **Selection** – Run this file to choose a configuration from among those defined in the settings file.
  - opa_setup – UNIX platforms, Bourne/Korn shells
  - copa_setup – UNIX platforms, C shell
  - opa_setup.bat – Windows

- **Supplemental** – For compatibility with previous versions of Oracle Clinical; belongs in the user’s home directory on the PSUB server machine.
You can edit and maintain the Definition scripts for your installations. However, do not modify the Selection scripts, which call the definition script. The selection script is used in the following contexts:

- When you start or stop the PSUB process for a database, you log in to the operating system as RXCPROD. The selection script is called to set the environment to that database.
- Each time a PSUB job is requested, PSUB executes the setup script to condition its environment to the correct version of the Oracle Clinical code, running against the correct database.
- You may execute the selection script at the command line to define server environments for various purposes, such as connecting to a particular database or running SQL scripts under RXC_TOOLS.

Administrators are not required to create or modify users’ installation scripts to enable users to submit back end jobs through the client interface. However, you must add entries to the initialization scripts of users who need to run opa_setup, SAS, or SQL*Plus from the back end command line. See "Setting Up Power Users" on page 1-17 for details.

**Editing opa_settings.bat**

During installation of the server code, the Installer creates the file opa_settings.bat, located in the \opapps\bin directory. File opa_settings.bat contains the commands to set environment variables at startup and execution of the PSUB process. Edit this file, and change the following assignments (in bold type), if necessary:

```bash
set NLS_DATE_FORMAT=DD-MON-RRRR
```

NLS_DATE_FORMAT determines the format in which client applications running on the Windows server transfer date information to and from the database. The format must specify the year as RRRR in order to be Year 2000 compliant.

```bash
set NLS_LANG=american_america.utf8
```

NLS_LANG determines which language settings Oracle uses when it reads and writes values into the database. The NLS_LANG entry in your registry for your iSuites Oracle Home must be consistent with the NLS_LANG setting in the Oracle Home and your databases.

If you don’t have a UTF8 character set database, you can use these character sets:

- american_america.us7ascii
- american_america.we8iso8859p1

**Note:** Do not create new databases with the default character set (AL32UTF8) by the Assistant.
Setting Up UNIX Environments

On UNIX systems, you run the selection script, which checks the arguments you provide to define a configuration against the settings file. If the arguments are valid, the script applies the appropriate values to the corresponding environment variables in the current shell. The syntax for calling the selection script depends on whether you use the C shell or Bourne/Korn shells. For all shells, the selection script accepts at least one argument and an optional second:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>database</td>
<td>Indicates the database to be used. This can be:</td>
</tr>
<tr>
<td></td>
<td>- the Oracle SID of a database on the PSUB server</td>
</tr>
<tr>
<td></td>
<td>- connect string of a database instance on a machine other than the PSUB server</td>
</tr>
<tr>
<td></td>
<td>- (minus sign). The script sets the code environment, but preserves the existing database context, if any.</td>
</tr>
<tr>
<td>code_env</td>
<td>Optional. An Oracle Clinical code environment designator that must refer to a code environment defined in the opa_settings file on the PSUB server.</td>
</tr>
</tbody>
</table>

Note that using a - (minus) for the first argument neither updates nor creates a database context.

<table>
<thead>
<tr>
<th>Arguments specified</th>
<th>Resulting behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>database</td>
<td>With no code_env specified, the script sets only Oracle-level environment variables needed for applications to access database.</td>
</tr>
<tr>
<td>- (minus) code_env</td>
<td>With no database specified, the script sets only environment variables needed for jobs to run. That is, it sets PATH to include RXC_PSUB and RXC_BIN, and defines the RXC_* environment variables.</td>
</tr>
<tr>
<td>database code_env</td>
<td>With both database and code_env specified, the script executes both sets of commands.</td>
</tr>
</tbody>
</table>

- In Bourne and Korn shells:
  
  p1=database  
  p2=code_env  
  . opa_setup
  
  For example:
  
  $ p1=test  
  $ p2=400  
  $ . opa_setup

- In C shell:
  
  copa_setup database code_env
  
  For example:
  
  %copa_setup test 46
Changing Configuration Settings on UNIX Database Servers

The configurations are defined in the opa_settings file. The Oracle Universal Installer creates all necessary entries in this file during installation of software and creation or upgrade of databases. The most common reason to modify the opa_settings file is to customize the values set for various environment variables during execution of back end jobs. You may also need to modify the file to delete databases that are no longer available and enable the use of additional code environments against a database.

Each line in the file defines a particular type of environment information:

```
record_type_key: field_1[field_2]... 
```

Starting with an identifier of the type of information (record), the line also contains a colon (:) separator, followed by fields that contain the information for that record, each separated by colons. Table A–1 lists and describes each record type.

<table>
<thead>
<tr>
<th>Record Type Key</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>oratab_filespec</td>
<td>Location of the file oratab on the server.</td>
<td>oratab_filespec:/etc/oratab</td>
</tr>
<tr>
<td></td>
<td>Field 1: Fully specified path to the OPA directory</td>
<td></td>
</tr>
<tr>
<td>tnsnames_filespec</td>
<td>Location of the file tnsnames.ora on this server. Ensure that this file has</td>
<td>tnsnames_filespec:/etc/tnsnames.ora</td>
</tr>
<tr>
<td></td>
<td>an entry for each connect string (that is, a reference to a remote database)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>that is required for OPA applications. The record provides information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>about accessing the database over the network.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 1: Fully specified path to the file tnsnames.ora</td>
<td></td>
</tr>
<tr>
<td>opa_home</td>
<td>Location of Oracle Health Sciences (formerly known as Oracle Pharmaceutical</td>
<td>opa_home:/pharm/home/opapps</td>
</tr>
<tr>
<td></td>
<td>Applications) products on the server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 1: Fully specified path to the OPA directory</td>
<td></td>
</tr>
<tr>
<td>remote_db_home</td>
<td>Location of an available remote database, to which ORACLE_HOME should be</td>
<td>remote_db_home:hpx1:/u01/app/oracle/product/9.2.0</td>
</tr>
<tr>
<td></td>
<td>set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 1: Net8 connect string of the remote database.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 2: ORACLE_HOME value that is used while accessing the current database</td>
<td></td>
</tr>
</tbody>
</table>
Defaulting, Adding, and Customizing Values

Use db_env_setting entries to define the value you want environment variables to assume during execution of back end jobs. You can add an entry for any environment variable you want to define; the definition will be in effect for any database if you set field 1 to _DEFAULT_. If you want to limit the environment variable setting so that it affects only those jobs associated with a particular database, use the database’s SID as the value for field 1.

The environment variable settings in opa_settings affect all users. If you want to set a value for an environment variable for just one user, place a statement in that user’s .oclrc script—for example, RXC_DEBUG=TRUE; export RXC_DEBUG.

If you want an environment variable setting to affect all jobs that run against a particular database, add or modify a database-specific entry in opa_settings—for example, db_env_setting:TEST:SASORA:V8.

Finally, if you want the setting to affect all jobs run against any database, add or modify a _DEFAULT_ entry for that environment variable—for example, db_env_setting:_DEFAULT_:RXC_SAS_BATCH_QUEUE:b.

<table>
<thead>
<tr>
<th>Record Type Key</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>code_environment</td>
<td>Location of the code for a version of an OPA application.</td>
<td>code_env:oc46:/pharm/home/opapps/46</td>
</tr>
<tr>
<td></td>
<td>Field 1: A code environment designator, for example, OC46 for the Oracle Clinical 4.6 code.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 2: The fully specified path to the root directory for the version of the application software.</td>
<td></td>
</tr>
<tr>
<td>db_code_pair</td>
<td>Indicates that a particular code environment can be used with a particular database.</td>
<td>db_code_pair:prod:oc46</td>
</tr>
<tr>
<td></td>
<td>Field 1: The system identifier (SID) of a local database instance, or the connect string of a remote database instance.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 2: A code environment designator.</td>
<td></td>
</tr>
<tr>
<td>db_env_setting</td>
<td>Provides either a default or database-specific setting for an environment variable. The following environment variables must have at least default settings:</td>
<td>db_env_setting:_DEFAULT_SASORA:V7</td>
</tr>
<tr>
<td></td>
<td>NLS_DATE_FORMAT</td>
<td>db_env_setting:TEST:SASORA:V8</td>
</tr>
<tr>
<td></td>
<td>NLS_LANG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RXC_BATCH_QUEUE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RXC_NOW_STRING</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SASORA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>These settings are assigned default values at install time.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 1: The database SID, or connect string, if this is a database-specific setting for the environment variable; or <em>DEFAULT</em>, if this is a default setting across databases for this environment variable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 2: Name of an environment variable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Field 3: Value to be assigned to the environment variable</td>
<td></td>
</tr>
</tbody>
</table>
The last example in Table A–1 shows how to override a system-wide default setting with a database-specific setting. The environment variable SASORA is defaulted to V7 for all databases with the exception of the TEST database, where it is defaulted to V8.

---

**Note:** The default settings for all databases or the specific settings for a particular database, such as NLS_LANG, must be correct in the opa_settings file.

---

**Constraints on the opa_settings File**

Oracle recommends that you use the defaults where possible, and add overrides only as needed. In addition, note the following constraints if you edit this file:

- In the opa_settings file, there should be exactly one each of these record types:
  - `opa_home`
  - `oratab_filespec`
  - `tnsnames_filespec`

- For each database instance appearing in a `db_code_pair` record, a value must be defined for each of the database environment settings (record type key `db_env_setting`). The setting may be made either through a generic `_DEFAULT_` record, or through a database-specific record.

**Checking for Errors in the opa_settings File**

If you modify the opa_settings file, run the script `~/opapps/bin/check_opa_settings.sh` to check the settings file for errors. The syntax is:

`check_opa_settings.sh [-nowarn] settings-file-name`

The script generates an error message if it finds any duplicate `record_type` key values. These would cause an error if present when opa_setup is run. (In fact, opa_setup calls check_opa_settings.sh to preclude this. However, opa_setup does not check for warnings. See below).

Unless the `-nowarn` argument is provided, check_opa_settings.sh will also generate a warning for multiple `db_code_pair` entries for a single database. While multiple `db_code_pair` entries are not invalid, they may represent a condition you do not want to allow. For instance, if you upgrade database `x` from 4.5 to 4.6, opa_settings would include:

```
db_code_pair:x:45
db_code_pair:x:46
```

In this case, check_opa_settings.sh warns you. Remove the line enabling the 3.2 code environment against the 4.0 database, so you don’t mistakenly start up a PSUB process in that configuration.

You might also want to disregard other warnings. For example, if you had a code tree for testing patches, as well as a production code tree. Then opa_settings might have:

```
db_code_pair:x:46
db_code_pair:x:46patchtest
```

You would disregard the warning check_opa_settings.sh would give, since both pairs are valid.
Changing Environment Variables on Windows

On Windows, the values to which you want environment variables to be set during back end job execution are maintained in opa_setup.bat. Below is a list of the Oracle Clinical environment variables that must be defined in opa_settings.bat.

- NLS_DATE_FORMAT
- NLS_LANG
- RXC_MAA_TAB_SPACE
- RXC_SAS_VIEW
- SASORA

To change or add environment variable settings active during back end job execution, edit the opa_settings.bat file with a text editor. Each line must be in the following format:

```
set variable_name=value
```

Settings that Affect Back-End Job Execution

This section lists environment variables that affect execution of jobs on back end servers. You can define values for the environment variables that apply to all back end jobs by defining their settings in opa_settings (UNIX) or in opa_settings.bat (Windows). On UNIX you can limit the scope of the environment variable setting to a single instance or to a single user. See “Defaulting, Adding, and Customizing Values”, above, for details.

- NLS_DATE_FORMAT – National Language Support
- NLS_LANG – National Language Support
- RXC_BATCH_QUEUE – Batch queue for non-blocking PSUB jobs (UNIX only)
- RXC_BDL_DIR – Spool directory for batch data load
- RXC_DEBUG_BUFFER_SIZE – Output buffer size for executing procedures
- RXC_IMMED_QUEUE – Batch queue for blocking PSUB job (UNIX only)
- RXC_LOG – Location for log files
- RXC_MAA_TAB_SPACE – Data extract temporary tablespace
- RXC_NOW_STRING – Time when PSUB job is executed (UNIX only)
- RXC_PRINTER – Default printer for Oracle Clinical
- RXC_SAS_BATCH_QUEUE – Default PSUB batch queue for SAS jobs (UNIX only)
- RXC_SAS_CONNECT – SAS connect string (UNIX only)
- RXC_USER – Root directory for creating data extract files
- SASORA – Version of Oracle compatible with SAS
- TEMP – Default location for FTP
- USER_BV_JOB – Script run after batch validation (back end server)

NLS_DATE_FORMAT

This variable specifies the format used for displaying dates and converting characters to dates. This is a Windows registry setting that the Installer sets.
The default value is “DD-MON-RRRR”.
You can modify this variable using \texttt{db_env_setting} records in the \texttt{opa_settings} file.

**NLS_LANG**
This variable specifies the language setting used by Oracle RDBMS to read from and write to the database. Set NLS_LANG to the appropriate language and character set.
The default value for this variable is:
\texttt{db_env_setting:\_DEFAULT\_\_NLS_LANG:american\_america.utf8}
You can modify this variable using \texttt{db_env_setting} records in the \texttt{opa_settings} file.

\begin{quote}
\textbf{Note:} The default settings for all databases or the specific settings for a particular database, such as NLS_LANG, must be correct in the \texttt{opa_settings} file.
\end{quote}

**RXC_BATCH_QUEUE**
This the batch queue for nonblocking PSUB jobs, on UNIX only.
If you want PSUB to use a batch queue other than the default for running user requests, redefine the setting for \texttt{rxc_batch_queue}. You can define it globally for all users, or individually by placing the command in the user’s login script.
Default is a.

**RXC_BDL_DIR**
This is the spool directory for batch data load.
When a user requests \textbf{Prepare to Completion} for a given data file group, and does not specify otherwise in the submission form, any resulting reloadable data files are written to the directory specified by RXC_BDL_DIR. If there are no reloadable files, the completed files are placed in RXC_LOG.

**RXC_DEBUG_BUFFER_SIZE**
This sets the output buffer size for executing procedures.
RXC_DEBUG_BUFFER_SIZE controls the size of the buffer space used for running a Validation or Derivation Procedure in debug mode. The installed default value is 200000; you may want to increase it to 100000.

**RXC_IMMED_QUEUE**
Batch queue for blocking PSUB jobs, on UNIX only
If you want PSUB to use a batch queue other than the default to process user requests for blocking jobs (such as default layout and generate procedure), redefine the setting for RXC_IMMED_QUEUE. It may be necessary to send blocking jobs to another batch queue so that they are not held up by other system activity, such as long-running reports.
Set this variable through \texttt{db_env_setting} records in the \texttt{opa_settings} file.

**RXC_LOG**
The directory where the system saves the log files of various processes.
**RXC_MAA_TAB_SPACE**
Oracle Clinical's Data Extract functionality requires a privileged Oracle user so that Oracle schemas can be created to hold Data Extract Views. The Oracle account for this purpose is RXC_MAA (Maintain Access Accounts).

RXC_MAA_TAB_SPACE specifies the name of the Oracle tablespace defined by RXC_MAA as the temporary tablespace when these schemas are created. During installation RXC_MAA_TAB_SPACE is set to TEMP1 with a size of 10Mb.

Set through `db_env_setting` records in the `opa_settings` file.

**RXC_NOW_STRING**
Defines the string for "now" that is used by the `at` command in the local language environment. This is effective only for 3GL and PL/SQL jobs submitted in immediate mode to run on UNIX back end servers. Default value is "now". To see your current "now" string, enter:

```
% echo $LANG
```
If LANG is undefined or is equal to "C", you have finished. The RXC_NOW_STRING is simply "now". Otherwise, do this:

```
% cd /usr/lib/nls/$LANG
dumpmsg at.cat
```
The string for "now" is the third item in the third set of output.

Set through `db_env_setting` records in the `opa_settings` file.

**RXC_PRINTER**
This is the environment variable to which PSUB refers when the user chooses RXC_PRINTER from the list of values for printing a PSUB job. It refers to the default printer for Oracle Clinical.

**RXC_SAS_BATCH_QUEUE**
This references to the default PSUB batch queue for SAS job, on UNIX only.

If you want PSUB to use a batch queue other than the default for running users' SAS requests, redefine the setting for RXC_SAS_BATCH_QUEUE, globally for all users, or individually by placing the command in the user's initialization file.

**RXC_SAS_CONNECT**
RXC_SAS_CONNECT defines the Oracle connect string to connect SAS to an Oracle database. The following examples assume an environment pre-set for a particular database.

For more information on connect strings, see your operating system-specific installation manual for SQL*Net.

The example below assumes SAS connects though the pipe driver. This is only possible when SAS and Oracle are installed on the same computer.

Sites using TCP/IP to connect from SAS to Oracle would replace the @p with an @t.

**Bourne shell**

```
RXC_SAS_CONNECT='oracle(path='''@p:'''')';
export RXC_SAS_CONNECT
```
C shell

setenv RXC_SAS_CONNECT 'oracle(path='''@p:'''')'

---

**Note:** On UNIX, if SAS (and SQL*Plus and PSUB) reside on the same computer as the Oracle Clinical database, and **RXC_SAS_CONNECT** is not defined, then you can set **remote_os_authent** to false. On Windows, **remote_os_authent** must be set to false.

Refer to the "Troubleshooting PSUB Based on the Batch Job's Execution Status" on page C-11 for additional information.

---

**RXC_USER**

This is the root directory for creating SAS files during data extract. For example, if **RXC_USER** is defined as /u01/oc, and **ORACLE_SID** is **prod**, then the data extract files go in /u01/oc/prod/...

---

**Note:** It is possible to set a different value for **RXC_USER** for each database, if you wish, overriding this default.

---

**SASORA**

This environment variable must be defined when you run SAS Access against an Oracle database. The default value shipped in opa_settings is V7, which is appropriate for SAS 6.12 running against either an Oracle 7.3 or 8i database. The appropriate value for SAS 8.2 running against either an Oracle 8i or 9i database is V8.

---

**TEMP**

This is the default temporary directory for FTP processes.

---

**USER_BV_JOB**

This environment variable specifies the name and location of a user-defined script to be executed as the last step of batch validation.

For example:

**UNIX (in .oclrc):**

```
USER_BV_JOB=/dir1/dir2/dir3/filename
export USER_BV_JOB
```

The full pathname of the file must be specified.

At batch validation run time, the environment variable is evaluated and the corresponding script is submitted for execution via PSUB. The script is called with two arguments: **clinical_study_id** and **clinical_study_version_id**.

---

**Windows Registry Settings**

This section lists the Windows registry settings used by Oracle Clinical. It describes the settings for the each of the following:

- **OPA Front End** on page A-11
- **Online Help** on page A-11
- **Oracle Clinical Front End** on page A-12
OPA Front End

These registry variables apply across products and are located on the SOFTWARE\ORACLE branch of the registry. Each value is set by the Installer.

**Table A–2  OPA Front End Registry Variables and Example Values**

<table>
<thead>
<tr>
<th>Registry variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMS_PATH</td>
<td>c:\opapps46\opa</td>
<td>The path that is searched to find forms.</td>
</tr>
<tr>
<td>FORMS_TIMEOUT</td>
<td>10</td>
<td>The Forms time-out period</td>
</tr>
<tr>
<td>OPA_CONFIG</td>
<td>opa46</td>
<td>The OPA configuration name</td>
</tr>
<tr>
<td>OPA_RQM_URL</td>
<td><a href="http://server.domain/dev60cgi/rwcg60.exe">http://server.domain/dev60cgi/rwcg60.exe</a></td>
<td>The URL for the Reports Queue Manager</td>
</tr>
<tr>
<td>OPA_HOME</td>
<td>c:\opapps46</td>
<td>The top-level OPA products directory</td>
</tr>
<tr>
<td>OPA_HOME_DIR</td>
<td>c:\opapps46\opa</td>
<td>The Oracle Health Sciences product directory</td>
</tr>
<tr>
<td>OPA_SERVER</td>
<td>server.domain</td>
<td>The full server name</td>
</tr>
<tr>
<td>OPA_PORT</td>
<td></td>
<td>This should be set to NULL, that is, &quot;blank&quot;, to facilitate either HTTP or HTTPS operations.</td>
</tr>
<tr>
<td>NLS_DATE_FORMAT</td>
<td>DD-MON-YYYY</td>
<td>The default NLS date format</td>
</tr>
<tr>
<td>NLS_LANG</td>
<td>AMERICAN_AMERICA.W</td>
<td>The NLS language</td>
</tr>
<tr>
<td>OPA_JARS</td>
<td>f60all_jinit.jar, opaicons.jar</td>
<td>The names of the OPA jar files to be downloaded to the client. This is used by the OUI to coordinate between product changes to the opa46 config section in the formsweb.cfg file.</td>
</tr>
<tr>
<td>OPA_PHYSICAL_MAP</td>
<td>c:\opapps46\html\repout</td>
<td>The physical mapping to the repout directory</td>
</tr>
<tr>
<td>OPA_VIRTUAL_MAP</td>
<td>/OPA_REPOUT/</td>
<td>The virtual mapping to the repout directory</td>
</tr>
<tr>
<td>OPA_DEV_VERSION</td>
<td>60</td>
<td>The version of Developer</td>
</tr>
</tbody>
</table>

Online Help

These registry variables are located on the SOFTWARE\ORACLE branch of the registry. Each value is set by the Installer.

**Table A–3  Xhelp Registry Variables and Example Values**

<table>
<thead>
<tr>
<th>Registry variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>opa_doc_dir</td>
<td><a href="http://server.domain/opa46">http://server.domain/opa46</a></td>
<td>The URL for standard online help and documentation location</td>
</tr>
<tr>
<td>opa_custom_doc_dir</td>
<td><a href="http://server.domain/opa46">http://server.domain/opa46</a></td>
<td>The URL for custom online help and documentation</td>
</tr>
<tr>
<td>opa_xhelp_dir</td>
<td>c:\opapps\html\xhelp</td>
<td>The location of the online help directories</td>
</tr>
</tbody>
</table>
### Oracle Clinical Front End

These registry variables are located on the SOFTWARE\ORACLE branch of the registry. Each value is set by the Installer.

**Table A–4  Oracle Clinical Front End Registry Variables and Example Values**

<table>
<thead>
<tr>
<th>Registry variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMS_PATH</td>
<td>c:\opapps46\oc</td>
<td>The Forms60 path that is searched to find forms</td>
</tr>
<tr>
<td>forms60_repformat</td>
<td>HTML</td>
<td></td>
</tr>
<tr>
<td>forms60_userexits</td>
<td>c:\opapps46\oc\f60xtb.dll; c:\oc\rxcdcl.dll</td>
<td>User Exits referenced from Oracle Clinical/RDC</td>
</tr>
<tr>
<td>forms60_defaultfont</td>
<td>MS Sans Serif 1.0</td>
<td>Oracle Clinical default font</td>
</tr>
<tr>
<td>oc_home_dir</td>
<td>c:\opapps46\oc</td>
<td>The Oracle Clinical top level directory</td>
</tr>
<tr>
<td>OC_DE_TEXTFONT</td>
<td>Arial.8</td>
<td>The default font for data entry field prompts and boilerplate text</td>
</tr>
<tr>
<td>OC_DE_FIELDFONT</td>
<td>Arial.8</td>
<td>The default font for response fields in data entry</td>
</tr>
<tr>
<td>OPA_JARS</td>
<td>f60all_jinit.jar, opaicons.jar,ociicons.jar,pharmaocjle.jar,pharmaocgle.jar,xmlcomp.jar,xmlparserv2.jar,jle2-0-3.jar</td>
<td>The names of the OPA JAR files</td>
</tr>
<tr>
<td>OPA_XMLTEMP_HTTP</td>
<td><a href="http://server.domain/opa46/rdc/temp">http://server.domain/opa46/rdc/temp</a></td>
<td>The URL to access the xmltemp directory</td>
</tr>
<tr>
<td>OPA_XMLTEMP_UNC</td>
<td>Either: \appserver\rdc\temp or c:\opapps46\html\rdc\temp</td>
<td>The value that is passed to the report server that informs it how to access xmltemp. If the forms and reports servers are on the same computer, the value is a directory. If the forms and reports servers are on different computers, the value will be a UNC</td>
</tr>
</tbody>
</table>

### RDC Front End

These registry variables are located on the SOFTWARE\ORACLE branch of the registry on the middle-tier computer where RDC is installed.

**Table A–5  RDC Front End Registry Variables and Example Values**

<table>
<thead>
<tr>
<th>Registry variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMS_PATH</td>
<td>c:\opapps46\rdc</td>
<td>The path that is searched to find forms</td>
</tr>
<tr>
<td>RDC_HOME_DIR</td>
<td>c:\opapps46\rdc</td>
<td>The location of the RDC directory</td>
</tr>
<tr>
<td>RDC_DCIF_IMAGES</td>
<td>\appserver\rdc\dcif_images or c:\opapps46\html\rdc\dcif_images</td>
<td>The location of the RDC directory images</td>
</tr>
<tr>
<td>OPA_LOCALHOST</td>
<td>oclw2k16.us.oracle.com</td>
<td>If the RDC installation is used by client computers other than the application server, use the fully qualified application server name. If you plan to run the client locally on the application server computer, you can use the machine name e.g., oclw2k16.</td>
</tr>
</tbody>
</table>
**Table A–5 (Cont.)**  
**RDC Front End Registry Variables and Example Values**

<table>
<thead>
<tr>
<th>Registry variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPA.XML.LOC</td>
<td>OPA_HOME/temp</td>
<td>This is a folder where temporary files are created at runtime and deleted at the end. During installation, this is set to its default value: OPA_HOME/temp.</td>
</tr>
<tr>
<td></td>
<td>e.g.: c:\opapps46\temp</td>
<td></td>
</tr>
<tr>
<td>OPA.PJC_LISTENER_DEBUG_LEVEL</td>
<td>1</td>
<td>Sets the debug level for the Pluggable Java Component (PJC). There are three options for the value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. 0: no debug</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. 1: Low debug level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. 2: High debug level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>By default, it is set to &quot;1&quot; during installation.</td>
</tr>
<tr>
<td>OPA.PJC_PORT_STARTNUMBER</td>
<td>5567</td>
<td>The port number at which the PJC communicates with the PDF Data Entry Form. This is typically set to a random value of &quot;5567&quot;.</td>
</tr>
<tr>
<td>OPA.PJC_PORT_NUMATTEMPTS</td>
<td>10</td>
<td>The number of attempt the PJC will make to open a listener port. With each attempt, the PJC increases the port number by one and continues until it has attempted to open the port a number equal to this value.</td>
</tr>
<tr>
<td>OPA.PDF_FILE_SOURCE</td>
<td>SERVER</td>
<td>The setting that specifies whether the PDF files will be served from the application server computer or from the cache location on the client computer. The valid values are 'CLIENT' or 'SERVER'. The default value will be 'SERVER'. You change this value to allow users to cache PDF files locally.</td>
</tr>
</tbody>
</table>

**OCN Front End**

These registry variables are located on the SOFTWARE\ORACLE branch of the registry. Each value is set by the Installer.

**Table A–6**  
**OCN Front End Registry Variables and Example Values**

<table>
<thead>
<tr>
<th>Registry variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMS_PATH</td>
<td>c:\opapps46\oc</td>
<td>The path that is searched to find forms</td>
</tr>
<tr>
<td>OCN_HOME_DIR</td>
<td>c:\opapps46\oc</td>
<td>The OCN Home directory</td>
</tr>
</tbody>
</table>
Report Server

These registry variables are located on the SOFTWARE\ORACLE\<HOME> branch of the registry for the Reports Server installation. Each value is set by the Installer.

<table>
<thead>
<tr>
<th>Registry variable</th>
<th>Example Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORTS_PATH</td>
<td>c:\opapps46\opa, c:\opapps46\oc, c:\winnt\fonts</td>
<td>The path the system searches to find reports</td>
</tr>
<tr>
<td>REPORTS_CLASSPATH</td>
<td>c:\opapps46\lib\pdrgenerator.jar,</td>
<td>The path Reports uses to find classes</td>
</tr>
<tr>
<td></td>
<td>c:\opapps46\lib\pdappend.jar,</td>
<td></td>
</tr>
<tr>
<td>NLS_DATE_FORMAT</td>
<td>DD-MON-RRRR</td>
<td>The date format when running reports for Oracle Clinical/RDC with the NLS option</td>
</tr>
<tr>
<td>OC_RSERVER_DIR</td>
<td>c:\opapps46\oc</td>
<td>The Reports Server directory</td>
</tr>
<tr>
<td>NLS_LANG</td>
<td>american_america.utf8</td>
<td>The NLS language that is used when running Reports; supported values are: UTF8, US7ASCII, WE8IS08859P1, or any single byte character set</td>
</tr>
<tr>
<td>OC_PDF_REPORTS_TEMP_OC</td>
<td>c:\opapps46\temp</td>
<td>The temporary directory that the system uses when you run reports.</td>
</tr>
<tr>
<td>OC_RPT_GRIDWIDTH</td>
<td>8</td>
<td>The default gridwidth for the report</td>
</tr>
<tr>
<td>OPA_HOME</td>
<td>c:\opapps46\</td>
<td>The top level Oracle Health Sciences products directory. This is written to both the default and the specific branches of the registry.</td>
</tr>
<tr>
<td>OC_DE_FIELDFONT</td>
<td>Arial.8</td>
<td>The default font for response fields in data entry</td>
</tr>
<tr>
<td>OC_DE_TEXTFONT</td>
<td>Arial.8</td>
<td>The default font for data entry field prompts and boilerplate text</td>
</tr>
<tr>
<td>OC_RPT_FIELDFONT</td>
<td></td>
<td>The font for response field data in reports. If not specified, the system uses the value for OC_DE_FIELDFONT.</td>
</tr>
<tr>
<td>OC_RPT_TEXTFONT</td>
<td></td>
<td>The font for CRF header field prompts, question prompts, and boilerplate text in reports. If not specified, the system uses the value for OC_DE_TEXTFONT.</td>
</tr>
<tr>
<td>RDC_DCIF_IMAGES</td>
<td>c:\opapps46\rdc\dcif_images</td>
<td>The location of images that are used for DCI Forms and PDR generation.</td>
</tr>
<tr>
<td>RDC_PDF_PRINT_TOOL</td>
<td>&quot;C:\Program Files\Adobe\Acrobat 5.0\Acrobat\Acrobat.exe&quot; /t</td>
<td>The path to the Adobe Acrobat or Reader application on the Report Server. Path must be in double-quotiation marks and the &quot;/t&quot; must be included in the value.</td>
</tr>
</tbody>
</table>

Registry Keys

This section provides details about individual registry keys. Use this information if it becomes necessary to modify the value of a registry key due to configuration or hardware changes.

In general, the values assigned to the keys are set by the Oracle Universal Installer (OUI), during the installation of various Oracle Health Sciences (formerly known as Oracle Pharmaceutical Application—OPA) components, based on answers you provide during the information-collection phase of the installation.
The following registry keys are described in the following sections:

- **FORMS_PATH**
- **OPA_JARS**
- **OPA_XMLTEMP_UNC**
- **OPA_XMLTEMP_HTTP**
- **RDC_DCIF_IMAGES**
- **OC_DE_FIELDFONT**
- **OC_DE_TEXTFONT**

**FORMS_PATH**

The value that is assigned to this key is based on the Oracle Health Sciences products that are installed on the computer. As each component is installed, the OUI appends product-specific values to the existing value. For example, if the Thesaurus Management System (TMS) is installed on a system where Oracle Clinical and RDC are installed, the FORMS_PATH value would be:

```
c:\opapps46\opa;c:\opapps46\oc;c:\opapps46\rdc;c:\opapps46\tms
```

The path string that each component contributes to the FORMS_PATH key value is listed in Table A–8, "Product-Specific Registry Values for the FORMS_PATH Key".

<table>
<thead>
<tr>
<th>Product</th>
<th>Addition to the FORMS_PATH value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPA Front End</td>
<td>c:\opapps46\opa</td>
</tr>
<tr>
<td>Oracle Clinical Front End</td>
<td>c:\opapps46\oc</td>
</tr>
<tr>
<td>RDC Front End</td>
<td>c:\opapps46\rdc</td>
</tr>
<tr>
<td>TMS Front End</td>
<td>c:\opapps46\tms</td>
</tr>
<tr>
<td>AERS Front End</td>
<td>c:\opapps46\aers</td>
</tr>
</tbody>
</table>

**OPA_JARS**

The value that is assigned to this key is based on the OPA products that are installed on the computer. As each component is installed, the OUI appends product-specific values to the existing value.

**OPA_XMLTEMP_UNC**

The value that is assigned to this key is the directory on the application server that Oracle Clinical uses to write temporary files during the DCI form generation process. When the Oracle Clinical client is installed, the OUI sets this value to:

```
<drive>:\opapps46\html\rdc
```

The Oracle Reports server also writes files to this directory during DCI form generation. If any reports server used for this task is located on a different computer than the application server, the directory must be shared, with read/write privileges, to the <domain>/account on the report server; see the Oracle Clinical Installation Guide. Also, the path specified in the value must use the Microsoft Universal Naming Convention (UNC) format.
For example, if, during initial installation, you reply to the OUI that there is a standalone report server, the Installer sets this the value of this key to:

```
\<appserver\rdc\temp
```

and requires that you share `<drive>:\opapps46\html\rdc` with the share name "rdc".

If the only report server that you use to generate DCI forms co-exists on the same computer with the application server, there is no need to share the xmltemp directory and the path specification can be a simple local directory name, such as, `c:\opapps46\html\rdc\temp`.

**OPA.XMLTEMP.HTTP**

The value that is assigned to this key must be a valid URL that points to the forms server directory to which Oracle Clinical writes temporary files during DCI form generation. When the Oracle Clinical client is installed, the OUI sets this value to:

```
<drive>:\opapps46\html\rdc\temp
```

This is the same directory specified by the **OPA.XMLTEMP.UNC** key.

In order for the system to use this directory, it must be supported by an http virtual directory that can serve files from it.

For example, if the temp directory is `c:\opapps46\html\rdc\temp` on myOCSever, then a virtual directory must be associated with `c:\opapps46\html`:

```
alias /opa46/ "c:\opappas46\html"
```

This allows a URL of `http://myOCServer/opa46/rdc/temp/MYFILE.pdf` to resolve and serve the file, `c:\opapps46\html\rdc\temp\MYFILE.pdf`.

**RDC.DCIF.IMAGES**

The value that is assigned to this key must be the path specification of the directory in which image files that are referenced in DCI forms are found; see "Setting Up Image Viewing for DCI and DCM Form Layouts HTML Preview Feature" on page 11-3. However, the Patient Data Report generation subsystem uses its own registry variable to locate the path to the directory.

When the Oracle Clinical client is installed, the Installer sets this value to:

```
<drive>:\opapps46\html\rdc\dcif_images
```

In a manner similar to **OPA.XMLTEMP.UNC**, the `dcif_images` directory specification must be one that can be resolved by any reports server that generates DCI forms.

If the only report server that you use to generate DCI forms co-exists on the same computer with the forms server, there is no need to share the images directory and the path specification can be a simple local directory name, such as, `c:\opapps46\html\rdc\dcif_images`.

If any reports server used for DCI forms generation is located on a different computer than the forms server, then:

- The path specification used for the value of **RDC.DCIF.IMAGES** must use the UNC format.
- The forms server images directory must be shared, so it can be accessed from other computers.
- The `<domain>/<account used to set up the Reports Server>` must have read/write privileges on the shared forms server directory.

For example, if, during initial installation, you reply to the Installer that there is a standalone report server, the Installer sets this the value of this key to:

```
\\<appserver>\rdc\dcif_images
```

and requires that you share `<drive>:\opapps46\html\rdc` with the share name "rdc".

---

**Note:** If the Patient Data Report generation process cannot locate the path to the `dcif_images` directory, each CRF that contains an image in its layout is not printed in the report.

---

**OC_DE_FIELDFONT**

The value assigned to this key regulates the font size of response value that are typed in response fields and displayed in the data entry windows and the Patient Data Reports (PDRs). This registry key is present in:

- the `SOFTWARE\ORACLE\` branch, where it is set as part of the Oracle Clinical Front End installation
- the `SOFTWARE\ORACLE\<9iDS>` branch, where it is set as part of the Oracle Clinical Report Server installation

The default value in both locations is "Arial.8".

If you want Patient Data Reports (PDRs) to look the same as the CRFs that are displayed in data entry windows, the value for both keys must be identical.

**OC_DE_TEXTFONT**

The value assigned to this key regulates the font size of the question prompts and boilerplate text in the data entry windows and the Patient Data Reports (PDR). This registry key is present in:

- the `SOFTWARE\ORACLE\` branch, where it is set as part of the Oracle Clinical Front End installation
- the `SOFTWARE\ORACLE\<9iDS>` branch, where it is set as part of the Oracle Clinical Report Server installation

The default value in both locations is "Arial.8".

If you want PDRs to look the same as the CRFs that are displayed in data entry windows, the value for both keys must be identical.

**RDC_PDF_PRINT_TOOL**

The value assigned to this key determines the location of the Adobe Acrobat or Reader executable, which allows users to run PDF patient data reports with "PRINTER" specified as the output type. This value must be in the form:

```
"<acrobat-reader_path>" /t
```

Note that the double-quotiation marks around the path and the "/t" switch are required. A typical example of a value is:

```
"C:\Program Files\Adobe\Acrobat 6.0\Acrobat\Acrobat.exe" /t
```
In addition to setting this key correctly, ensure that the Adobe Acrobat or Reader application is running on the Reports Server prior to users initiating this type of report job, that is, a job that specifies the output type as "PRINTER".
All Oracle Clinical data not stored in the database, such as SAS Views created with the Data Extract facility, is stored in the SAS_VIEW directory on the Back end server computer. The access restrictions enforced in the database must also be enforced at the file-system level. The structure of the SAS_VIEW directory tree is shown below, together with the associated permissions. Owner is not relevant; in fact, the owner is the user who created the file or directory through the use of the application.

The top-level directory, opapps, is set by the installer. Entries are indented to show the relative subdirectory nesting level.

<table>
<thead>
<tr>
<th>Directory</th>
<th>UNIX</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>opapps</td>
<td>(0775)</td>
<td></td>
</tr>
<tr>
<td>sas_view</td>
<td>(2775, oclsascr)</td>
<td>(oclsascr, FULL Control)</td>
</tr>
<tr>
<td>db_name</td>
<td>(2775, oclsascr)</td>
<td>(oclsascr, FULL Control)</td>
</tr>
<tr>
<td>study</td>
<td>(0775, oclsascr)</td>
<td>(oclsascr, FULL Control)</td>
</tr>
<tr>
<td>account_type</td>
<td>(0775, oclsascr)</td>
<td>(oclsascr, FULL Control)</td>
</tr>
<tr>
<td>*.sas</td>
<td>(0664, oclsascr)</td>
<td></td>
</tr>
<tr>
<td>*.log</td>
<td>(0664, oclsascr)</td>
<td></td>
</tr>
<tr>
<td>*.com</td>
<td>(0775, oclsascr)</td>
<td></td>
</tr>
</tbody>
</table>

Where db_name is the name of the database instance, study is the study code in Oracle Clinical, and account_type is the data extract view account type (TEST, CURRENT, STABLE or SNAPSHOT). For example:

**UNIX:** /pharm/home/opapps/sas_view

**Windows:** \opa-db1\sas_view

To change the location of the SAS_VIEW storage, change the appropriate environment settings for your platform, as shown below:

**UNIX:** db_env_setting:_DEFAULT_:RXC_USER:/u01/home/opapps
db_env_setting:_DEFAULT_:RXC_SAS_ROOT:/u01/home/opapps/sas_view

**Windows:** set RXC_SAS_ROOT=%oui_sas_root%\%database%
To set the protections on your directory structure, run the appropriate script:

**UNIX:**  
$RXC_TOOLS/set_rxc_user.sh

**Windows:**  
%RXC_TOOLS%\set_rxc_user.bat

(These scripts are run automatically by the Installer.)

**On UNIX Systems Only:**
The path to the SAS_VIEW directory must not contain uppercase characters (PSUB changes all path specifications to lowercase for UNIX). If your standard naming conventions require uppercase characters in the path, you can provide lowercase and uppercase versions of paths with symbolic links as needed. For example, if the standard path to the SAS_VIEW directory is

/usr1/home/Clinical/opapps/sas_view

you could create this link:

% cd /usr1/home  
% ln -s Clinical clinical

**References to**

/usr1/home/clinical/opapps/sas_view

would also work.
Managing High Sequence Numbers

If you upgraded to Oracle Clinical 4.6 from either release 4.5.1, 4.5.2, or 4.5.3, it is important to ensure that the internal identifier for each of the following does not exceed 2,147,483,647 (that is, \(2^{31}-1\)):

- Received DCMs (RDCMs)
- Received DCIs (RDCIs)
- Discrepancies

When the internal identifier for these tables exceeds 2,147,483,647, the system incorrectly processes the identifiers in other tables, such as the Responses table. You can still view and update the responses, however batch validation, data extract, replication, and procedure execution operations fail or run incorrectly.

In Oracle Clinical 4.5.1 and after, there is code to prevent sequence numbers from exceeding 2,147,483,647. The `OCL_DE_CONFIG` reference codelist includes a short value entry, "SEQUENCEBUFFER", which is assigned an initial long value of 1,000,000. At this setting, when a sequence number is within 1,000,000 of 2,147,483,647, the system displays a warning message when a user attempts to use a relevant subsystem and the system exits the current screen.

Assessing Sequence Sizes

Perform this test to determine if your database is nearing the point where this situation may occur. If the results of this test for the number of Received DCMs, Received DCIs, and discrepancies is well below 2,147,483,647, you do not have to perform the other tasks described in this section.
Assess the Number of RDCMs
To determine the sequence number for RDCMs in your system:
1. Connect to your database through SQL*Plus as RXC.
2. Issue this command to assess the number of Received DCMs:
   
   ```sql
   SELECT received_dcm_seq.nextval FROM dual;
   ```
   
   The system returns a number.
3. Compare the number returned to 2,147,483,647:
   - If the internal identifier number is approaching 2,147,483,647, you should reseed the received_dcm_seq sequence after you complete this task.
   - If the number exceeds 2147483647, reseed received_dcm_seq, identify data errors, and repair data errors according to the instructions in the following sections.
   - If neither of the above conditions apply, run this command from SQL connected as RXC:
     ```sql
     alter sequence received_dcm_seq maxvalue 2147483646;
     ```

Assess the Number of RDCIs
To determine the sequence number for RDCIs in your system:
1. Connect to your database through SQL*Plus as RXC.
2. Issue this command to assess the number of Received DCMs:
   
   ```sql
   SELECT received_dci_seq.nextval FROM dual;
   ```
   
   The system returns a number.
3. Compare the number returned to 2,147,483,647:
   - If the internal identifier number is approaching 2,147,483,647, you should reseed the received_dci_seq sequence after you complete this task.
   - If the number exceeds 2147483647, reseed received_dci_seq, using the instructions in the following sections. If you use RDC or DCAPI, contact Oracle Support.
   - If neither of the above conditions apply, run this command from SQL connected as RXC:
     ```sql
     alter sequence received_dci_seq maxvalue 2147483646;
     ```

Assess the Number of Discrepancies
To determine the sequence number for discrepancies in your system:
1. Connect to your database through SQL*Plus as RXC.
2. Issue this command to assess the number of Received DCMs:
   
   ```sql
   SELECT discrepancy_entry_seq.nextval FROM dual;
   ```
   
   The system returns a number.
3. Compare the number returned to 2,147,483,647:
   - If the internal identifier number is approaching 2147483647, you should reseed the discrepancy_entry_seq sequence after you complete this task.
   - If the number exceeds 2147483647, reseed the discrepancy_entry_seq sequence after you complete this task.
   - If neither of the above conditions apply, run this command from SQL connected as RXC:

```
alter sequence discrepancy_entry_seq maxvalue 2147483646;
```

### Reseeding Sequences

If any of your response numbers are approaching or exceed 2,147,483,647, you should reseed them.

#### Reseed RDCM Sequence Numbers

To reseed the Received DCM sequence:

1. Stop all Oracle Clinical activity on the database until this procedure completes.
2. Connect to the database through SQL*Plus as RXC.
3. To determine which seed numbers are in use, issue this command.

```
SELECT distinct mod(received_dcm_id,100)
FROM received_dcms;
```

4. Choose a new starting seed number value between 0 and 99 (inclusive) that is not in the list returned by the above step.
5. Issue this command to drop the received_dcm_seq sequence:

```
DROP sequence received_dcm_seq;
```

6. Issue this command to recreate the Received DCM sequence with this value as the new starting value (for example, SEQ_START_NO):

```
CREATE SEQUENCE received_dcm_seq
INCREMENT BY 100
START WITH &SEQ_START_NO
MAXVALUE 2147483646
MINVALUE 1
NOCYCLE
CACHE 20
NOORDER;
```

7. Issue this command to grant access:

```
GRANT SELECT on received_dcm_seq to RXCLIN_MOD;
```

---

**Note:** If this is a replicated environment, run this command in all replicated instances.
Reseed RDCI Sequence Numbers
To reseed the Received DCI sequence:

1. Stop all Oracle Clinical activity on the database until this procedure completes.
2. Connect to the database through SQL*Plus as RXC.
3. To determine which seed numbers are in use, issue this command.
   
   ```sql
   SELECT distinct mod(received_dci_id,100)
   FROM received_dcis;
   ```

   **Note:** If this is a replicated environment, run this command in all replicated instances.

4. Choose a new starting seed number value between 0 and 99 (inclusive) that is not in the list returned by the above step.
5. Issue this command to drop the received_dci_seq sequence:
   
   ```sql
   DROP sequence received_dci_seq ;
   ```

6. Issue this command to recreate the Received DCI sequence with this value as the new starting value (for example, SEQ_START_NO):
   
   ```sql
   CREATE SEQUENCE received_dci_seq
   INCREMENT BY 100
   START WITH &SEQ_START_NO
   MAXVALUE 2147483646
   MINVALUE 1
   NOCYCLE
   CACHE 20
   NOORDER;
   ```

7. Issue this command to grant access:
   
   ```sql
   GRANT SELECT on received_dci_seq to RXCLIN_MOD;
   ```

Reseed Discrepancies Sequence Numbers
To reseed the discrepancies sequence:

1. Stop all Oracle Clinical activity on the database until this procedure completes.
2. Connect to the database through SQL*Plus as RXC.
3. To determine which seed numbers are in use, issue this command.
   
   ```sql
   SELECT distinct mod(discrepancy_entry_id,100)
   FROM discrepancy_entries;
   ```

   **Note:** If this is a replicated environment, run this command in all replicated instances.

4. Choose a new starting seed number value between 0 and 99 (inclusive) that is not in the list returned by the above step.
5. Issue this command to drop the discrepancy_entry_seq sequence:

   DROP sequence discrepancy_entry_seq ;

6. Issue this command to recreate the Discrepancy sequence with this value as the new starting value (for example, SEQ_START_NO):

   CREATE SEQUENCE discrepancy_entry_seq
   INCREMENT BY 100
   START WITH &SEQ_START_NO
   MAXVALUE 2147483646
   MINVALUE 1
   NOCYCLE
   CACHE 20
   NOORDER;

7. Issue this command to grant access:

   GRANT SELECT on discrepancy_entry_seq to RXCLIN_MOD;

---

**Error Messages**

This section offers fixes or workarounds for certain error messages that users may encounter. It includes the following topics:

- **Message: ORA-12223** on page C-5
- **Message: ORA-04020** on page C-5
- **Message: Unable to Change Mode** on page C-6

**Message: ORA-12223**

Full message text:

ORA-12223 TNS: internal limit restriction exceeded  
*This message may appear in the .log file.*

**Cause:** This error can occur when you submit a job to the server while running the process invoked by selecting **Conduct**, then **Data Extract**, and **Maintain Views**.

**Action:** Increase the swap space on the PSUB server.

**Message: ORA-04020**

Full message text:

ORA-04020: Deadlock detected while trying to lock.  
*This message may appear when batch validation is running and the user who submitted it switches between production and test modes.*

**Cause:** This message may appear when batch validation is running and the user who submitted it switches between production and test modes.

**Action:** Create a separate test account for each user who needs to switch modes frequently. See **Chapter 2, "Oracle Clinical Menu-Based Security"**, for instructions on modifying menu roles.
**Message: Unable to Change Mode**

323600 Unable to change to test mode, another session may be connected.
323700 Unable to change to production mode, another session may be connected.
325700 Unable to change to test mode (\0), synonyms not created.

**325800 Unable to change to production mode (\0), synonyms could not be dropped.**

For each of the above error messages, Oracle Clinical users should check with the administrator. This problem could be due to an RXCSYN package error, missing grants to RXC, or synonym conflicts with your schema objects.

These messages may appear if you switch between production and test modes while having another session open under the same userid. It may also appear if a user submitted a reports job before switching modes, and the reports engine is still associated with that user. The system considers this to be another session by the same user.

To check if you are logged on to more than one session, from SQL, you can enter the command:

```sql
select username from v$session;
```

If you do not have access to v$session, consult with a DBA.

In the case of reports jobs, you can prevent future problems by changing the MAXIDLE time of the Reports Server, which controls the length of time a user/engine session is kept open.

Follow these steps:

1. Open up the report queue manager and select the report queue of concern.
2. Select **Options**, then **Privileges**, and **Administrator** and log on as administrator.
3. Choose **Queue**, then **Properties** and change the maximum idle time to one minute or some reasonable smaller number (depending on the number of reports, users, and so forth on that queue).

In the case of a user switching modes, you can create a separate test account as described under ORA-04020, above.

---

**System Malfunction: GPF Occurs During Data Entry**

When a general protection fault (GPF) occurs during data entry, the system creates file rxcdecde.dbg, which contains a description of the cause of the GPF. The file resides in the RXC_ROOT directory.

**Situation: $ulimit unlimited**

On HP-UX and Compaq Tru64 UNIX, if the operating system parameter ulimit is set to unlimited, you get the following error when you run opa_setup or oraenv:

```
sh: unlimited: The specified number is not valid for this command
```

The workaround is to edit the oraenv shell script, adding a test for the word "unlimited." To modify this script, follow these steps:

1. Change your location to the bin directory:

   ```bash
   % cd $ORACLE_HOME/bin
   ```
2. Make a backup copy of oraenv named oraenv.O

3. Use a text editor to add the two lines shown below in **bold** to the file oraenv.

```bash
if [ $? = 0 -a "$ULIMIT" != "unlimited" ]; then # added line
    if [ $? = 0 -a "$ULIMIT" -lt 2113674 ]; then
        if [ -f $ORACLE_HOME/bin/osh ]; then
            exec $ORACLE_HOME/bin/osh
        else
            for D in 'echo $PATH | tr : " "'
            do
                if [ -f $D/osh ]; then
                    exec $D/osh
                fi
            done
        fi
    fi
fi
```

4. Save and exit the file oraenv

**PSUB Jobs**

This section describes steps you should take, in order, when you troubleshoot PSUB problems. To troubleshoot a PSUB job:

1. "Check the Failure Text in the Submitted Batch Jobs Window" on page C-7
2. "Check the PSUB Log Files" on page C-8
3. "If Batch Jobs Hang and the Batch Queue Is Full" on page C-9
4. "Determining if PSUB is Running for a Database" on page C-10
5. "Handling PSUB Failures that Return "Fatal two-task communication protocol" Error" on page C-14
6. "Tracking Previous PSUB Process Connections" on page C-15

**Check the Failure Text in the Submitted Batch Jobs Window**

If a problem arises while you are running PSUB, you should first review the Failure Text field of the Submitted Batch Jobs window.

To check this field for your batch job:

1. Open the Submitted Batch Jobs window: select Admin, then PSUB/Reports Jobs, and then Batch Jobs.
2. Locate the relevant Batch Job ID number.
3. Check the Execution Status of the job. If there is an entry in the Failure Text field, make a note of its contents
If the failure text does not help you to resolve the problem, refer to the "Check the PSUB Log Files" section on page C-8. If your batch job is hanging because the batch queue is full, refer to the "If Batch Jobs Hang and the Batch Queue Is Full" section on page C-9.

Check the PSUB Log Files

The PSUB process log files are cumulative, text-based descriptions of PSUB activity. These files are very helpful when you are troubleshooting problems with PSUB. Process log files can include time stamped entries for:

- error messages returned by the PSUB process
- all jobs submitted by the user; the entry may include each job's:
  - message id
  - batch_job_id
  - user name.

Naming Convention

On both UNIX and Windows systems, PSUB process log file names are in the form:

```
rxcpsd_product_instance_code_environment_1.log
```

On UNIX systems, there is a second process log file. Whenever you examine the "*_1.log" process log file on UNIX systems, you should also check this second file to see if it contains relevant entries. Its name is identical to the first log file, except that it has an ".2" suffix, rather than ".1". So the second UNIX process log file name is in the form:

```
rxcpsd_product_instance_code_environment_2.log
```

The "*_2.log" process log files contain error and warning messages that are generated by certain UNIX commands that the PSUB daemon executes (e.g., non-background commands). These commands are not present in the PSUB service on Windows. Therefore, Windows systems only generate "*_1.log" process files.

Verbose vs. Nonverbose Mode

The [verbose | nonverbose] argument must be included when the PSUB startup command, `rxcpsdps`, is executed. We recommend that you start PSUB in verbose mode because the process logs that are generated:

- contribute to efficient troubleshooting, and
- do not pose significant disk space concerns.

On UNIX systems, rxcpsdps is 'wrapped' in the OPA script start_psub. By default, the start_psub script executes rxcpsdps in verbose mode.

On Windows systems, the PSUB service requires that you explicitly provide the [verbose | nonverbose] argument. Refer to the "Managing the PSUB Process" section on page 8-3.

If you cannot check the Failure text or the .out and .log files because the batch queue is hung, refer to the "If Batch Jobs Hang and the Batch Queue Is Full" section on page C-9.

Verify that the process log files for the relevant Batch Job ID exist.

Review the .out and .log files. The following table summarizes these files. Make a note of any error messages.
Check the job-specific log and output files first, then the central log file. In the central log file, search for the batch job ID number to find the relevant entry. See if the database and code environment settings are correct.

### If Batch Jobs Hang and the Batch Queue Is Full

If all PSUB jobs hang (that is, they do not reach a completed execution status), and the batch queue is full, attempt to clear the queue and submit a single job to PSUB. If a problem then occurs with a single job, it may be clearer which area is causing the problem. The method for clearing the queue is either: stop all of the hung batch jobs (on UNIX systems) or stop the PSUB service (on Windows systems).

The queue may become full and PSUB jobs may hang under the following circumstances:

- PSUB is waiting, either for an operating system resource, or a database resource
- the operating system is overloaded, for example, a built-in limitation, such as maximum number of processes, is exceeded.

### Stopping Batch Jobs on UNIX Systems

This section describes how to stop batch jobs on UNIX systems. See also "Starting and Stopping PSUB Manually in UNIX" on page 8-4.

**Stopping an Individual Job** Follow these steps to stop an individual batch job:

1. Navigate to Admin, PSUB/Reports Jobs, and Batch Jobs).
2. Locate and select the row associated with the relevant Batch Job ID.
3. Click the Stop button.

**Stopping all Jobs** On UNIX servers, a series of hanging jobs can cause the batch queue to become full. When the queue fills and is backlogged with hanging jobs, all jobs are eventually given an execution status of SUBMIT_FAILED or SUBMITTED. If this type of problem occurs frequently, it may be advantageous to fine-tune the existing queues or add more queues.

**Administrator-level Troubleshooting** If, after trying all relevant solutions, you are unable to stop the jobs on a UNIX server, contact your Administrator so that she may use the solutions described here.

Only Administrator-level personnel should attempt to stop PSUB jobs using these solutions. Use these strategies, in the order they are listed, to stop PSUB jobs.

1. Use the stop_psub utility. Refer to the Stopping PSUB on a UNIX system question.
2. Identify and then stop the processes that are hanging:
   a. To identify the process that is hanging, use either:
      
      ```
      ps -ef|grep rxcprod
      
      or
      
      ps -ef|grep userid
      ```
   
   b. To stop all of the hanging processes that you identified in Step 2a, use this command:
      
      ```
      kill -9 pid
      ```

3. Log in as rxcprod and, at a command prompt, enter:
   
   ```
   at -l [-q]
   ```

   This command lists all of the jobs that are currently in all of the queues. Each job has a unique ID number. (Refer to the at man pages for additional information.)

   If there are jobs pending in the queue, the following command, which uses the unique ID number to remove specific jobs from the queue, may be of use:
   
   ```
   at -r id
   ```

1. If you are able to stop all PSUB jobs, stop and then restart the PSUB daemon and submit one job. If it hangs, try to isolate whether one particular module is the cause or if any PSUB job hangs, regardless of module.

2. If you are unable to determine a module that is causing the problem and jobs are still hanging, the only recourse is to reboot the computer.

**Stopping Batch Jobs on Windows**

On Windows systems, we recommend that you:

1. Stop the PSUB service.

2. Shut down any databases, if any, that are on the computer.

3. Reboot the computer.

4. Start the PSUB service.

   See “Starting and Stopping PSUB Manually in Windows” on page 8-6.

**Determining if PSUB is Running for a Database**

To find out if a PSUB process is listening to a particular database, and if it is, what code environment it is running in, enter this query:

```sql
SQL> select host, code_environment, stop_ts
2  from psub_process_log
3  where start_ts = ( 
4  select max (start_ts) from psub_process_log;
```

This query returns the:

- computer on which PSUB was last started against the database
- code environment
- state of the process:
  - if stop_ts is null, the PSUB process is currently active
  - if stop_ts is not null, the PSUB process is stopped.

---

**What PSUB Processes are Running on a Given UNIX Server?**

Use this command to find out if PSUB is running on particular UNIX server:

```
% ps -ef | grep -i rxcpsdps
```

The process search command, ps, returns descriptions of the PSUB daemons that are currently running. Each row that is returned represents one PSUB process running on the server. Each process has a unique product_instance and code_environment pair. The format of the response to the process search command listed above is:

```
rxcpsdps [verbose|nonverbose] product_instance code_environment
```

**Example C–1 Using the ps Command**

Two examples of ps command usage:

```
rxcprod 15685 1 0 Apr 04 ? 0:00 rxcpsdps verbose sun3x8 ssuneja_oc40_sun
rxcprod 4143 1 0 Apr 02 ? 0:00 rxcpsdps verbose sun1x40 40102_8163
```

---

**Is PSUB Running on a Given Windows Server?**

Use this procedure to find out if the PSUB service is running on a given Windows server.

1. Open the Control Panel.
2. Double-click the Services icon.
3. In the Services window, note the status of the PSUB service with the relevant database name. The status will be “Started” if the service is running.

---

**Troubleshooting PSUB Based on the Batch Job’s Execution Status**

Execution status as reported in the Submitted Batch Jobs form is shown below. You can take various actions depending on execution status.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTERED</td>
<td>The user has requested a job submission.</td>
</tr>
<tr>
<td>SUBMITTED</td>
<td>The process submitted the job to the batch queue; it may be pending.</td>
</tr>
<tr>
<td>SUBMIT_FAILED</td>
<td>The process attempted to submit the job to the batch queue but failed.</td>
</tr>
<tr>
<td>STARTED</td>
<td>The job is executing on the batch queue.</td>
</tr>
</tbody>
</table>
Examine the Submitted Batch Jobs window, look for the Execution Status and Failure Text for your Batch Job ID, and take one of the following actions, depending on the circumstances.

**ENTERED**
If the Execution Status of your batch job remains at ENTERED, perhaps:

- The PSUB process is not running on the server, or it is not receiving the request from the client.
- The corresponding Oracle user’s operating system account does not exist.

**SUBMITTED**
If the Execution Status of your batch job remains at SUBMITTED, perhaps:

- The user is not connected through an OPS$USER Oracle account.
- The job is pending in the batch queue, or the batch queue is stopped.
- The PSUB process does not have Write permission for the user’s RXC_LOG directory on the PSUB server.
- A log file exists with the same number as the one for the submitted job. This is a rare situation. Delete the old log file and resubmit the job.
- In the case of a PSUB job that stays in SUBMITTED status even though the PSUB daemon is up and running, if your .log file says:
  
  ERROR: Error while connecting:
  
  ORA-01017: invalid username/password; logon denied

  Exiting...
  
  Edit the file sqlnet.ora by commenting out the following line, and save it.

  SQLNET.AUTHENTICATION_SERVICES = (NTS)

  If the PSUB daemon is still running you are all set to resubmit the PSUB jobs. Otherwise, stop, and then start the PSUB daemon.

**PSUB does not start after installing or upgrading**

1. Locate the file, sqlnet.ora.
2. Locate and comment out the line (i.e., place '#' at the beginning of the line):

   sqlnet.authentication_service=(NTS)

3. Start PSUB.

If PSUB fails to start:

1. Locate and uncomment the line in sqlnet.ora:

   sqlnet.authentication_service=(NTS)
2. Locate and ensure that these lines in init.ora are **not** commented out:

   remote_os_authent=true
   os_authent_prefix="OPS$

---

**Note:** On UNIX, if SAS (and SQL*Plus and PSUB) reside on the same computer as the Oracle Clinical database, and **RXC_SAS_CONNECT** is not defined, then you can set remote_os_authent to false. On Windows, remote_os_authent must be set to false.

---

4. Start the databases.
5. Locate and comment out the line in sqlnet.ora.

   sqlnet.authentication_service=(NTS)

**SUBMIT_FAILED**

If the Execution Status of your batch job is **SUBMIT_FAILED**, examine the Failure Text. If this action gives no possible cause, perhaps:

- The command rsh (remsh on HP-UX) cannot be executed by the RXCPROD user. Check that the host name in the `/etc/hosts.equiv` file is the official name of the host as specified in `/etc/hosts`.
- The user’s password is not correct.
- The batch queue does not exist. Check the Long Value of the BATCH QUEUE NAME local reference codelist.
- The batch queue is in a stopped state.

**STARTED**

If the Execution Status of your batch job remains at **STARTED**, perhaps:

- The job is executing and waiting for some resource.
- The job is hung.

**FAILED**

If the Execution Status is **FAILED**, examine the Failure Text. If this action gives no possible cause, perhaps:

- The report or command exited with error status.
- The report or executable file does not exist.
- The print command exited with failure status, because, for example, the specified print queue does not exist.

**Other Items to Check**

Make sure the Long Value of the entry **SERVER_OS** in the local reference codelist **OCL_STATE** is correct for your operating system—that is, set to Windows or UNIX. **SERVER_NAME** in **OCL_STATE** should be set to the database/PSUB server.
Handling PSUB Failures that Return "Fatal two-task communication protocol" Error

If you submit a PSUB job that fails and returns a "Fatal two-task communication protocol" error (this failure is sometimes followed by the "End-of-communication-channel" in the core dump information on your console), you might have the environment variable NLS_LANG set inconsistently with the settings in the database.

To verify that the environment variable NLS_LANG matches the actual database settings:

1. Execute the following query:

   ```sql
   SQL> select parameter, value from V$NLS_PARAMETERS
   where parameter in ('NLS_LANGUAGE','NLS_TERRITORY','NLS_CHARACTERSET');
   ```

2. Open `opa_settings` and search for the following string:

   ```
   db_env_setting:database_name:NLS_LANG
   ```

   a. If you do not find this string, add a line with the following syntax:

      ```
      db_env_setting:database_name:NLS_LANG:NLS_LANGUAGE:NLS_TERRITORY:NLS_CHARACTERSET
      ```

      Where `NLS_LANGUAGE:NLS_TERRITORY:NLS_CHARACTERSET` are the values returned in step 1.

   b. If you find the string, correct the values to match the values in step one in the following syntax:

      ```
      db_env_setting:database_name:NLS_LANG:NLS_LANGUAGE:NLS_TERRITORY:NLS_CHARACTERSET
      ```

      Where `NLS_LANGUAGE:NLS_TERRITORY:NLS_CHARACTERSET` are the values returned in step 1.

Handling PSUB Failure that Returns "Illegal use of PSLAUNCH..." Error

In a UNIX environment, you may see the following error when you submit a PSUB job (3GL or PLSQL):

```
Illegal use of PSLAUNCH by user. Job Id=batch_job_id. Exiting...
```

This error generally does not occur each time you run a job.
Workaround

If above problem happens frequently in your environment the workaround is to modify $RXC_PSUB/launchps.sh as follows:

1. Logon to the UNIX computer on which the PSUB daemon is running, as owner of the file launchps.sh (this is usually OPAPPS).
2. Run opa_setup as appropriate for your shell environment.
3. Change to the $RXC_PSUB directory.
4. Edit launchps.sh by adding the following line just before the 'pslaunch' command:

   ```
   sleep 2
   pslaunch $4 $5 $6 $7 $3 $8
   ```

This introduces a 2-second delay before the system calls pslaunch. You may increase the delay if you still have a problem.

Tracking Previous PSUB Process Connections

To find out specific information about PSUB connections to a given database, query the table RXC.PSUB_PROCESS_LOG. This will return the:

- instance
- environment
- time a PSUB process started
- time a PSUB process stopped.

Example C–2  Host and Code Environment

This query will return the host and code environment for the last time PSUB was started against the database.

```
SQL> SELECT start_ts, host, code_environment, server_os
        FROM psub_process_log
        WHERE start_ts = (SELECT MAX(start_ts) FROM psub_process_log);
```

Example C–3  Start and Stop Timestamps

This example lists, in chronological order, all start and stop timestamps of PSUB processes.

```
SQL> SELECT start_ts, stop_ts, host, code_environment
        FROM psub_process_log ORDER BY 1;
```

Database Trace

You can trace a session connected to the Oracle Clinical Database and generate a log file. The following example, explains how to run a trace while in the Maintain DCM form.

1. Start a SQL*Plus session as SYS, or another user with the DBMS_SYSTEM role.
2. Find the session id and serial number of the Clinical user working in the Maintain DCM form:

   ```
   SQL> SELECT sid, serial# FROM v$session where username = 'OPS$userid';
   ```
3. Assume that 8 and 12 are returned for sid and serial#, enable SQL trace for the user as follows:
   SQL> exec dbms_system.set_sql_trace_in_session(8,12,TRUE)

4. Have user perform the operation that causes the error. After the error is returned disable SQL trace:
   SQL> exec dbms_system.set_sql_trace_in_session(8,12,FALSE)

5. Find the trace file out in your USER_DUMP_DEST directory. For example,
   SQL> select value from v$parameter where name = 'user_dump_dest';
   
   where value is the path, something like, /ind/oraclelogs/maria/db/udump. The trace file is placed in this directory.
This appendix lists the tablespaces in an Oracle Clinical database. Names preceded by "d-" are created when the database is created. The remainder are created when the Installer is run to install Oracle Clinical in the database.

<table>
<thead>
<tr>
<th>Name</th>
<th>Oracle Clinical Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISC_REP_DATA</td>
<td>Disconnected Replication tables</td>
</tr>
<tr>
<td>RBS</td>
<td>To hold rollback segments</td>
</tr>
<tr>
<td>RXA_DES_DATA</td>
<td>Design tables</td>
</tr>
<tr>
<td>RXA_DES_IDX</td>
<td>Indexes for Design tables</td>
</tr>
<tr>
<td>RXA_LR_DATA</td>
<td>Lab Ranges tables</td>
</tr>
<tr>
<td>RXA_LR_IDX</td>
<td>Indexes for Lab Ranges tables</td>
</tr>
<tr>
<td>RXC_APP_IDX_TSPA</td>
<td>Indexes for Application Definition tables</td>
</tr>
<tr>
<td>RXC_APP_TSPA</td>
<td>Application Definition tables</td>
</tr>
<tr>
<td>RXC_DCD_IDX_TSPA</td>
<td>Indexes for Data Collection Definition tables</td>
</tr>
<tr>
<td>RXC_DCD_TSPA</td>
<td>Data Collection Definition tables</td>
</tr>
<tr>
<td>RXC_DCMQ_IDX_TSPA</td>
<td>Indexes for Data Management tables</td>
</tr>
<tr>
<td>RXC_DCMQ_TSPA</td>
<td>Data Management tables</td>
</tr>
<tr>
<td>RXC_DEF_IDX_TSPA</td>
<td>Indexes for Definition tables</td>
</tr>
<tr>
<td>RXC_DEF_TSPA</td>
<td>Definition tables</td>
</tr>
<tr>
<td>RXC_DISC_IDX_TSPA</td>
<td>Indexes for Discrepancy Management tables</td>
</tr>
<tr>
<td>RXC_DISC_TSPA</td>
<td>Discrepancy Management tables</td>
</tr>
<tr>
<td>RXC_GLIB_IDX_TSPA</td>
<td>Indexes for Global Library tables</td>
</tr>
<tr>
<td>RXC_GLIB_TSPA</td>
<td>Global Library tables</td>
</tr>
<tr>
<td>RXC_LI_IDX_TSPA</td>
<td>Indexes for Log-In tables</td>
</tr>
<tr>
<td>RXC_LI_TSPA</td>
<td>Log-In tables</td>
</tr>
<tr>
<td>RXC_RESP_IDX_TSPA</td>
<td>Indexes for Responses table</td>
</tr>
<tr>
<td>RXC_RESP_TSPA</td>
<td>Responses table</td>
</tr>
<tr>
<td>RXC_VRV_IDX_TSPA</td>
<td>Indexes for Validation Reported Values table</td>
</tr>
<tr>
<td>RXC_VRV_TSPA</td>
<td>Validation Reported Values table</td>
</tr>
<tr>
<td>d-SYSTEM</td>
<td>Standard Oracle tables</td>
</tr>
</tbody>
</table>
In addition, the Oracle Thesaurus Management System creates these tablespaces:

<table>
<thead>
<tr>
<th>Name</th>
<th>Oracle Clinical Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>d-TEMP</td>
<td>Standard Oracle tables</td>
</tr>
<tr>
<td>TEMP1</td>
<td>Temporary Data Extract tables</td>
</tr>
<tr>
<td>TEST_DATA</td>
<td>Test tables</td>
</tr>
<tr>
<td>TEST_INDEX</td>
<td>Indexes for Test tables</td>
</tr>
<tr>
<td>d-TOOLS</td>
<td>Standard Oracle tables</td>
</tr>
<tr>
<td>d-USERS</td>
<td>Standard Oracle tables</td>
</tr>
</tbody>
</table>

In addition, the Oracle Thesaurus Management System creates these tablespaces:

<table>
<thead>
<tr>
<th>Name</th>
<th>Oracle Thesaurus Management System Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMS_DATA</td>
<td>Data tables</td>
</tr>
<tr>
<td>TMS_IDX</td>
<td>Index tables</td>
</tr>
</tbody>
</table>
This appendix covers system administration tasks that arise from changes to the system hardware or software.

Recreating Symbolic Links—UNIX Only

The symbolic links to the Oracle Clinical executables are lost when you, for example, copy the installation directory to a new drive (see also "Relocating Oracle Clinical" on page E-2).

To recreate the symbolic links, use the relink_rxc.sh script as follows:

```
% ksh
$ $RXC_TOOLS/relink_rxc.sh symbolic_links > $RXC_ROOT/relink_rxc.log 2>&1
exit%
```

The parameter `link` tells the script to re-establish symbolic links to the current Oracle Clinical executables, not to be confused with relinking the server code (see the next section).

Relinking Server Code—UNIX Only

With UNIX servers, you need to relink Oracle Clinical code files after either of the following events:

- a patch or upgrade to your operating system
- a patch to your Oracle RDBMS

Use the `opapps` userid to link all the server code from the supplied object files and set the file protections.

1. Enter the following commands:

```
% ksh
$ $RXC_TOOLS/relink_rxc.sh > $RXC_ROOT/relink_rxc.log 2>&1
$ exit
```

2. Check for errors in the log file, `relink_rxc.log`, using the following command:

```
% grep -i error $RXC_ROOT/relink_rxc.log
```

Or, to find out whether a relink has been successful by using a utility, enter:

```
% ksh $RXC_TOOLS/rxcchkobj.sh progs relink.mk
```
If all executables are created successfully, the output shows a set of empty
directory paths. Any executable that is not created is listed in the rxcchkobj output.
For example:

```
Expected progs in =====> /u01/home/oppaps/bin/46/build/tools
gen_views: No such file or directory
cnvstatus: No such file or directory
```

This indicates that two executables, gen_views and cnvstatus, were expected but
not created.
You should investigate the cause of the listed executables not being created.

**Relocating Oracle Clinical**

Should you decide to move the Oracle Clinical installation, you will have to edit the
location references in one or more files, according to your server platform.

**UNIX**

In the file `oc/46/psub/launchps.sh`, modify the directory reference in this section:

```bash
p1=$1; export p1;
p2=$2; export p2;
./pharm/home/opapps/oc/46/bin/opa_setup
USERNAME=$5; export USERNAME;
```

In the file `bin/opa_setup`, modify the directory reference in this section:

```bash
if [ ${OPA_BIN:-0} = 0 ]
then
  OPA_BIN=/pharm/home/opapps/oc/46/inst/bin
fi
```

**Windows**

Edit the file `oc/46/psub/launchps.bat`.

**Updating Oracle Clinical Seed Data**

If you run the Installer to create or upgrade an Oracle Clinical database, the Installer
places the correct seed data in that database. If, however, you upgrade an existing
Oracle Clinical database manually, you must also upgrade the seed data manually.

To upgrade Oracle Clinical seed data manually:

1. Set the environment variables as follows (for details, see Appendix A):
   ```bash
   opa_setup database code_environment
   ```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE_SID</td>
<td>set to the name of the database</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>set to your desired Oracle home directory</td>
</tr>
<tr>
<td>PATH</td>
<td>make sure <code>oracle_home/bin</code> is in your PATH variable</td>
</tr>
</tbody>
</table>

2. Collect the passwords for OPA and RXC.

3. From the install directory, run the script `loadseed.sql`:
   ```bash
   SQLPLUS RXC/password@database
   loadseed.sql
   ```
Collecting Statistics for Optimization

Oracle Clinical provides SQL scripts that you can use to collect computed statistics on the data distribution and storage characteristics of certain tables, indexes, and partitions. These statistics are used by the Cost-based optimizer.

Table E–1 lists the schema for which each script gathers statistics. If you want computed statistics for customer-created tables, such as Thesaurus Management System tables, you must collect them yourself. The Oracle Clinical scripts are located in the RXC_INSTALL directory.

You must gather statistics on a regular basis to provide the optimizer with information about schema objects. How often you run these scripts depends on how quickly the data volumes in your Oracle Clinical tables increase.

During the initial use of Oracle Clinical you should run these scripts frequently since the data distribution changes rapidly. Once the database is populated, you should run statistics whenever the data volume changes by more than 20% or when some event, such as using a new feature, causes the data distribution to change measurably.

Table E–2, "Suggested Frequency for Analysis Scripts" gives a rough estimate on the frequency of running the analysis scripts.

---

### Table E–1 Scripts for Optimization Statistics

<table>
<thead>
<tr>
<th>Schema</th>
<th>Collection script</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXA_DES</td>
<td>anadestab.sql</td>
</tr>
<tr>
<td>RXA_LR</td>
<td>analrtab.sql</td>
</tr>
<tr>
<td>RXC</td>
<td>anarxctab.sql</td>
</tr>
<tr>
<td>OPA</td>
<td>anaopatab.sql</td>
</tr>
<tr>
<td>RXA_DES, RXA_LR</td>
<td>ananalstab.sql</td>
</tr>
<tr>
<td>RXC, OPA</td>
<td></td>
</tr>
</tbody>
</table>

---

### Table E–2 Suggested Frequency for Analysis Scripts

<table>
<thead>
<tr>
<th>Number of Studies</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 10</td>
<td>For each new study</td>
</tr>
<tr>
<td>10 to 20</td>
<td>For every other study</td>
</tr>
<tr>
<td>20 to 100</td>
<td>For every 5 studies</td>
</tr>
<tr>
<td>More than 100</td>
<td>For every 20 studies</td>
</tr>
</tbody>
</table>
access
A property of a user name which denotes that certain privileges have been granted to that user. The term is used in RDC documentation as a means to convey that a set of privileges have or have not been granted to a particular user. An example of typical usage is:

The Change Study button is displayed if you have access to more than one study.

access key
The keyboard key corresponds to the letter that is underlined in an item's on-screen title. The access key may be activated either singly or in combination with the ALT key.

active
A discrepancy status that indicates the relevant discrepancy is actionable by members of your user group.

active mode
In PDF data entry mode, a status assigned to a CRF that allows the user (with appropriate privilege) to modify the CRF. Although multiple CRFs may be open in Only one CRF may is an instance that can be updated by users with appropriate privileges.

approval history
A record, or set of records, associated with a CRF that lists each change in approval status starting with the initial approval action.

approval status
A designation that describes the current state of approval for a CRF. In RDC, there are four approval statuses:

1. not approved
2. approved
3. approval undone
4. awaiting re-approval

See also: discrepancy status, entry status, verification status
approval undone
An approval status that indicates a user with the approve privilege has removed the approved status of a CRF via the Undo Approval action.

approve
A privilege assigned to a user name that allows the user to alter the approval status of a CRF or a group of CRFs. The privilege is assigned at the site level only.

approved
An approval status assigned to a CRF that indicates a user with the approve privilege certifies that the CRF is an accurate representation of the source data. In RDC, an approval is equivalent to an electronic signature.

audit history
The set of all audit records for a given data point.
See also: audit record, data point

audit record
A set of information that describes an instance of data update. Each audit record includes the following information:

- the current value of the data point
- the previous value of the data point
- the user name that changed the data point
- the timestamp the data update occurred
- the change reason
- an optional comment

automatic progression
A feature of RDC that enhances user data entry, monitor verification, and investigator approval sessions. When automatic progression is enabled, as you complete work on the current CRF focus moves to the next CRF in a sequence specified by the settings in the Preferences window. The next CRF may be: within the current patient record, across Spreadsheet tabs; within the current patient, within the current event tab; or within the current CRF column.

Note that, based on system administrator settings, the Preferences window may not be available in your session.

awaiting re-approval
This is a system-generated approval status that indicates the CRF is approved, but that one or more response values were updated. A CRF in this approval status can be re-approved or the approval can be undone.

The CRF changes that cause the change in approval status include:

- response value update
- update to an investigator comment
- initiation of a new investigator comment
- update to a discrepancy
- initiation of a new discrepancy
**awaiting re-verification**

This is a system-generated verification status that indicates the CRF is verified, but that one or more response values were updated. A CRF in this verification status can be re-verified or the verification can be undone.

The CRF changes that cause the change in verification status include:

- response value update
- update to an investigator comment
- initiation of a new investigator comment
- update to a discrepancy
- initiation of a new discrepancy

**batch loaded data**

A designation that specifies response values were entered into a CRF by electronic means, as opposed to manually entered via a user performing data entry.

**blank**

An RDC entry status designation that is assigned to a CRF a user has defined as blank. A Blank CRF does not contain data, nor can data be collected while it is marked as blank.

**blank flag**

A check box GUI component used in the RDC Classic Data Entry window. The Blank Flag, or check box, is used to designate a CRF as blank. It is a standard item in the CRF Header area of a DE window and may also be present in the CRF Section, especially in multi-section CRFs.

**book**

A collection of phases, patients, visits, and CRFs within a study.

**Boolean expression**

In RDC, a statement that uses a Boolean operator to construct a search string by combining two or more search strings. For example, the Data Subsearch window, which is subcomponent of the RDC Search window, uses a Boolean operator to join two search phrases.

**Boolean operator**

A key word that is used to logically join two search strings in a Boolean expression. RDC accepts the following Boolean operators:

- AND
- OR

**browse**

1) A user action that entails reviewing existing data without adding new data or changing existing data.
2) A privilege, specific to manually entered CRFs, that provides the user with the ability to view existing data, but not to add new data or update existing data in CRFs. All RDC users must be assigned this privilege or a higher-level privilege that incorporates it.

**browse batch**

A privilege, specific to batch loaded CRFs, that provides the user with the ability to view existing data, but not to add new or update existing data in CRFs. This is the minimum privilege required for a user to view batch loaded CRFs.

**browse mode**

In PDF data entry mode, a status that allows the user to work with the CRF as read-only; that is, data entry or update is not permitted. More than one browse mode CRF can open simultaneously.

Two data entry window components allow you to quickly identify a CRF that is browse mode:

- the **watermark** that is visible along the left-hand margin of the CRF
- the CRF button bar, which differs significantly from the active mode version; in browse mode it contains three buttons:
  1. Close button
  2. Make Active
  3. Help.

See also: **active mode**

---

**cancel**

A user action that halts the current process or action and reverts to the state of the application that immediately preceded the process or action, without changing the status of the system or data.

**Cancel button**

In a dialogue or pop-up window, a Cancel button causes the application to dismiss the window, disregard changes made in the window, and revert to conditions that existed before the window opened.

**case book**

See **book**.

**case report form**

A paper or electronic record associated with a patient in a clinical study. Its purpose is to facilitate accurate collection of clinical data. RDC, CRFs are depicted electronically, either through the classic data entry window or the PDF data entry window.

See also: **CRF**
change history
A listing of the values that have been assigned to data or information. Each list item includes information that uniquely identifies it. Specifically, a timestamp and the user name of the person who made the change are recorded. In RDC, a change history can be associated with the following:

- response value
- investigator comment
- discrepancy

change reason
A constituent of an audit record. A standardized entry that explains why a data point changed. The change reason can be supplied either automatically (system-provided) or manually (user-provided).

Classic mode
In RDC, one of two data entry modes that are available which enable users with appropriate privileges to create and update patient data. In classic mode, the CRF is presented as an Oracle Form in a window that overlays the Main Application window, with the header displayed at the top of the form and each section presented as its own tab within the form window.

see also: data entry mode, PDF mode

clinical planned event
An occurrence, usually a visit, that is scheduled as part of a protocol to collect clinical data for a patient. In RDC, visits appear in the context of a case book.

closed
A discrepancy status that indicates the relevant discrepancy is not actionable by any user group. The discrepancy has been resolved, either by a user or the system.

See also: discrepancy status, user group, active, other, obsolete, discrepancy state

CPE
An acronym for clinical planned event.

created
An RDC entry status designation that is assigned to a CRF when all required CRF header data has been entered and saved, and no other data, i.e., response data, has been entered.

This is used when your internal process involves an administration person who logs the paper as received in-house, but the data have not yet been entered by the DE staff.

CRF
See case report form

criterion
In reference to the RDC Search window, one of the components that can be altered so that a certain set of study data is retrieved to the Spreadsheet. Each criterion is comprised of a set of parameters that allow you to make choices about the data you wish to retrieve.
The Search window is comprised of seven criteria:

1. Book criterion
2. Site criterion
3. Patients Subsearch window
4. Visit/Page Range Subsearch window
5. CRF Status Subsearch window
6. CRF/Visit Dates Subsearch window
7. Data Subsearch window

Each criterion is comprised of a certain number of parameters. With the exception of the book criterion, all Search window criteria have a default setting that retrieves the maximum amount of data. (The default for the Site criterion, for example, is “All Sites”).

You modify a criterion from its default by changing one or more of the parameters that comprise it.

- The Book and Site criteria each have a single parameter, which you access and modify from the Search window.
- All other criteria have multiple parameters, which you access and modify in the Subsearch window that is specific to each criterion.

**CRF header**
A component of a CRF as it is displayed in the Classic data entry window. It consists of one or more header fields, in which you collect information that uniquely describes and defines the current CRF.

RDC will not save a CRF to the study database until all required CRF header and CRF section header fields are collected.

**CRF information**
A field in a PDF mode CRF in which you collect information that uniquely describes and defines the current CRF. Based on the type of information it collects, the field may be required or optional. It correlates directly with a CRF header field.

RDC will not save a CRF to the study database until all required CRF information and CRF section information fields are collected.

**CRF section**
In RDC, a constituent of a CRF that is comprised of a set of related questions. Each CRF contains at least one section and may contain more than one. In Oracle Clinical terms, a section equates to a data collection module (DCM).

**current**
A discrepancy state that indicates action can be taken on the discrepancy – either by a user or by the system. It has not been made obsolete by the system.

See also: discrepancy state, obsolete, open, active, other, closed

**current study**
The data set that is active in RDC. The name of the current study is displayed in the title bar of the Main Application Window.
data collection instrument (DCI)
(General) The Oracle Clinical term for an RDC CRF. A DCI is composed of one or more DCMs.
See also: case report form, data collection module (DCM), CRF

data collection module (DCM)
(General) The Oracle Clinical term for a CRF section.
(Oracle Clinical) A set of one or more related groups of questions that pertain to a single clinical study visit.
See also: section

data entry mode
In RDC, a designation that describes the method that is used to enter data into a CRF and save it to the study database. There are two modes that are available:
1. Classic mode
2. PDF mode

data entry status
A designation that describes the current state of data entry for a CRF. In RDC, there are four data entry statuses:
1. created
2. blank
3. entry started
4. entry complete

data field
A location in the Question area of the Data Entry window in which you type a value that is the response to a CRF question.

data point
A location in a form where a data value may be entered. In most cases, a data point corresponds to a field in the data entry window.

data update
In RDC, the process of changing a CRF that has been created and saved to the database by altering a data point and saving the new version of the CRF to the database.
see also: created, data point, CRF, initial data entry

DCAPI
An acronym for the Data Capture API system.

DCM
An acronym for data collection module (DCM).
**default study**

The preferred study associated with your user name. RDC automatically selects the default study when you initiate a session.

- If you have access to one study, that is your default study.
- If you have access to more than one study, the study that was active when you closed the previous session is the default study.
- If you have access to more than study, but have not initiated a previous session in the current database, the system administrator can specify a default study.
- If the a default study is not specified, the system presents the Change Study window when you logon, which allows you to choose a study from among those to which you have access.

**discrepancy**

Data that falls outside of an expected range of values or is otherwise ‘flagged’ during the edit check process.

see also: discrepancy management, manual discrepancy, multivariate discrepancy, univariate discrepancy

**discrepancy action**

A process that changes the status of a discrepancy. There are two types of actions:

1. routing
2. resolution

**discrepancy change history**

The listing associated with a discrepancy that provides details of each update that was made it. The specific components of the discrepancy change history that are available to RDC users are:

1. timestamp of update
2. error text
3. discrepancy comment
4. current status
5. resolution
6. resolution comment
7. change by

**discrepancy management**

A process that systematically addresses discrepancies generated within a study. Discrepancy management attempts to identify the cause and assess the implications of each discrepancy and determine an appropriate action for the discrepancy. Its goal is to satisfactorily resolve all discrepancies associated with each CRF.

See also: discrepancy, multivariate discrepancy, univariate discrepancy, manual discrepancy

**discrepancy record**

An entry which is part of the study database that defines the pertinent aspects of a discrepancy, from its initial occurrence and through each action that is taken on it.
discrepancy state
The highest level designation of a discrepancy. A discrepancy can be in one of two states:

1. current
2. obsolete

discrepancy status
1) A designation that describes a current discrepancy
A designation that describes the current state of a CRF with regard to discrepancies. In RDC there are four discrepancy statuses:

1. none
2. closed
3. active
4. other

discrete value group
(Oracle Clinical) A set of responses that are acceptable for a given question. A DVG constrains the responses to a question to a distinct set of values. It may also allow the same question to be used in multiple instances.

It is also possible for one DVG to be a subset of another DVG. In this case, the child DVG is made up of responses that are part of the parent.

document
In RDC, the equivalent of a CRF.

document number
A system-assigned unique identifier for a particular collected CRF.

DVG
(Oracle Clinical) An acronym for discrete value group.

E

element

entry complete
An entry status that is assigned to a CRF in which all required fields have been entered, including CRF header fields and Question area response data points.

element status

entry started
An entry status that is assigned to a CRF in which data entry has been initiated but is not complete. CRFs that are assigned this entry status, some required data fields are complete, while some are not; the document has been saved in an incomplete status.

entry status
Formal stages of data entry, delineated in Oracle Clinical and RDC, that track the progression of a CRF from no data entered (“Created”) through entry complete, to approved.
See also: **blank, created, entry started, entry complete**

**event page**
The Spreadsheet view that is associated with a given *spreadsheet view tab*. An event page may represent a study, phase, or visit event and includes all of the patient rows and visit-page columns that are in the currently retrieved data set.

**F**

**focus**
In RDC, where the cursor is currently active. Focus may change from window to window, as when the cursor moves from the Main Application window to the first data field in the Data Entry window when you click a CRF cell. How focus changes is a consideration when you are modifying the settings on the Preferences window to enhance the efficiency of your data entry or verification/approval sessions.

**frozen**
A designation that is applied to a patient which indicates that all data has been received, entered, reviewed, and cleaned for the patient, CRF, or visit.

**G**

**Graphical User Interface**
The screen representation of a software application that uses graphical components, such as windows, icons, and menus, to effect user interaction, rather than typing command line entries.

**GUI**
An acronym for *Graphical User Interface*.

**H**

**header field**
A location in the CRF Header or the CRF Section Header in which you collect values that provide information about the CRF. A header field is either required or optional. All required header fields must be collected before the system permits a CRF to be saved to the study database.

In the CRF Header, the following header fields are available:

- visit date
- visit time
- comment
- blank flag
In CRF Section Headers, the following fields are available:

- date
- time
- blank flag
- clinical planned event
- lab
- qualifying value

Note: The preceding lists are specific to RDC only.

indicator question
A question used with certain question groups that allows "branching" during data entry based on the response.

For example, in a Drug Allergy question group, an indicator question could be, "Allergic to any drug?"

- If the response is "Yes", the remaining questions in the question group, such as "Drug Name" and "Type of Reaction", require responses.
- If the response is "No", the rest of the question group is not collected.

initial data entry
The step in the RDC workflow during which the CRF is initially opened and created. During this process all required CRF and CRF section header information is collected. Response data may or may not be collected.

installation reference codelist
(Oracle Clinical) A reference codelist that is defined and populated upon initial installation of the application.

See also: reference codelist

internal
A discrepancy status that can be assigned to a section discrepancy through a routing action. This type of discrepancy can be configured so that it is "hidden" from one or more user groups.

investigator comment
A textual explanation that is written by the investigator. It provides the investigator with the opportunity to include additional information with a response value. Each investigator comment is saved as part of the response with which it is associated.

RDC provides visual cues to alert the user to the presence of an investigator comment associated with a data point:

- the response field is displayed with a yellow background color
- the data value is displayed in a green font
when focus is in the relevant response field, the Data Entry window header includes an entry: `<Inv>`, to indicate the presence of the investigator comment.

**list of values**
A set of possible values for a data field. The list of values can generally be displayed by either clicking the button that is associated with list of value fields, pressing the List button or by pressing the F9 key.

Values that are defined for a discrete value group are displayed in a list of values.

See also: list of values, discrete value group

**lock**
a) a privilege that may be assigned that enables a user to lock a CRF or a set of CRFs see also: privilege,
b) A process that prevents subsequent update of a CRF. Under most circumstances, a locked CRF cannot be 'unlocked,' although administrators may permit, on a limited basis, a user to unlock a single CRF so that data may be updated.

**locked**
A status assigned to a CRF that indicates all data has been collected, approved, and verified. A locked CRF may be viewed in browse mode and may be included in PDRs, however, its data may not be updated under normal circumstances.

see also: PDR, status, browse mode, update mode, lock, unlock

**lock status**
A designation that describes the current state of a CRF, with regard to whether or not it may be updated. In RDC, there are two lock statuses:

- locked
- unlocked

**list of values**
An acronym for list of values.

**mandatory response field**
A response field in the question area of CRF section that should be completed before the CRF is saved in the Entry Complete status. Failure to do so results in the generation of a discrepancy, which is associated with the relevant response field.

Note that when RDC is in PDF-enabled mode, you may leave a mandatory field uncollected and save the CRF in the "Save Incomplete" status to avoid the initiation of a discrepancy.
**mandatory field discrepancy**

A discrepancy associated with a mandatory response field that is generated by the system when a CRF is saved. The discrepancy triggers when data for the field is not collected.

**manual discrepancy**

A discrepancy that is generated by a user, rather than a data point value. In RDC, a manual discrepancy may be associated with an entire CRF, a CRF section header or a specific response in the question area of a CRF.

See also: discrepancy, discrepancy management, section discrepancy

**menu bar**

The section of the Main Application window that provides access to RDC commands. Refer to the "Main Application window” and the "Menu bar" topics in the RDC User’s Guide for further information.

**menu command**

Each menu command is displayed in either black or grey font. If the command is displayed in black font, it is available and may be invoked under the current application conditions. If the command is displayed in grey font, it is not available under the current application conditions. If you attempt to utilize a menu command that is displayed in grey font, the system does not respond.

**menu item**

Each label (e.g., "File" or "Insert") that is used to categorize commands that are available from the menu bar. When a menu is accessed, either by clicking the label or using its access key, it drops down to display the list of menu commands that are associated with it.

**multivariate discrepancy**

A discrepancy that is dependent on two or more data point values, which can be within a single CRF or across multiple CRFs and/or visits. A multivariate discrepancy is generated when a CRF is saved, which causes the system to run the validation procedures that locate this type of discrepancy.

See also: discrepancy, discrepancy management, manual discrepancy

**N**

**Navigation pane**

In the Activity List window, the section, or frame, that comprises the left-hand portion of the window and presents a hierarchical listing of the current study and, the sites to which you have access. When you click a Site node, all patients assigned to the site that you have access to are displayed under the site name.

See also: node, scope, Task pane

**news item**

A message that is communicated by the study sponsor to some portion of its RDC users. News items are displayed in the News window.
node
An item in the hierarchical tree in the Navigation pane of the Activity List window. When you select a node, the tasks that are associated with it are displayed in the Task pane.

Each node represents one of three scopes: Study, Site, or Patient. Within the Navigation pane, there is only one Study node displayed. However, depending on the security settings associated with your user name, there may be more than one Site node displayed under the Study node, and generally many Patient nodes listed under each Site node.

non-repeating question group
A set of questions that are related, but for which there is not a single set of possible answers.

See also: question group, repeating question

not approved
An approval status assigned to a CRF that indicates the CRF has never been approved.

See also: approval status, approved, not approved, awaiting re-approval

not verified
A verification status that indicates the CRF has not yet been verified.

See also: verification status, verified, not verified, awaiting re-verification

O

obsolete
A system-generated discrepancy state assigned to a discrepancy that is associated with a response that is a constituent of a:

- repeating question row that was deleted
- a question that was deleted
- a CRF section that was deleted
- a CRF that was deleted.

A section discrepancy is made obsolete when its parent CRF is deleted or made blank. A data discrepancy is also made obsolete if the validation procedure upon which it is based is retired.

open
1) A designation for a discrepancy that indicates it is either in the active or other discrepancy status; that is, it is actionable by a user group.

2) A designation for a CRF that indicates it contains at least one active or other discrepancy.

optional CRF
A CRF that is planned in a visit, but which the protocol does not require to be collected. Optional CRFs are not included when the system determines whether there are missing pages. The information in the CRF Column Header of optional CRFs is displayed in italic font to distinguish each from required CRFs.
other

A **discrepancy status** that indicates the discrepancy is actionable by a user group other than yours.

See also:

P

**parameter**

In reference to the Search window, a component of a **criterion** that you use to define a specific property of the data you wish to comprise the **workset**. A parameter may be comprised of settings that represent a value or range of values that, when combined with any other parameters in the same criterion, define specific data.

See also: criterion, setting, search phrase, workset

**pass 2 complete**

A **data entry status** that assigned to CRFs that originate in the Oracle Clinical data entry system. It indicates that two-pass data entry was required for the CRF and that the second pass is complete.

**pass 2 started**

A **data entry status** that assigned to CRFs that originate in the Oracle Clinical data entry system. It indicates that two-pass data entry was required for the CRF and that at least one response field has been recorded in the second pass.

**patient**

The data that represents a participant in a clinical study. This includes demographic information and clinical results.

**patient data report**

In Oracle Clinical or RDC, a patient data report (PDR) is a generated compilation of data that is presented in a **PDF** document.

**patient list**

The set of patients that contain at least one CRF which satisfies the search criteria.

**patient number**

A designation for a a set of patient data that is unique across a given study. Patient numbers are assigned to a study as part of the Oracle Clinical Design process. Alternative terms include: enrollment number, allocation number, and randomization number.

The following rules apply to all patient numbers:

1. Each patient number must always be assigned to a site.
2. Each patient number may not be assigned to more than one site at a time.
3. The first character in the patient number string may be a non-zero numeric or an alphabetic character.
4. If the first character in the patient number string is alphabetic, the second character must be a non-zero numeric character.
5. Only the first character may be alphabetic.

**PDF**
An acronym for *Portable Document Format*.

**PDF mode**
In RDC, one of two *data entry modes*, PDF data entry enable users to create, view, and update a CRF that is presented as a PDF electronic document.

See also: *Classic mode, data entry mode*

**pending changes**
Changes that are made to a CRF that have not yet been committed to the study database. The changes that may be pending are response value, investigator comments, or discrepancies. The *save* action commits pending changes to the database.

**phase**
An attibute of a book that denotes a stage of a study. Phases are used to divide the study into logical groupings of visits. Examples of phases include: Screening, Dosing, and Follow-up.

You can use the RDC Spreadsheet to view CRFs by phases. To do this, select the Phase Spreadsheet view from the Spreadsheet View drop-down list box.

**Portable Document Format**
A type of file format.

Portable Document Format is a universal file format published by Adobe Systems, Inc., that preserves all of the fonts, formatting, graphics, and color of a source document that is generated on with any application on any operating system.

see also: *PDF*

**privilege**
The ability for an RDC user to perform a certain task. Privileges are granted to users in the RDC Administration study and site maintenance windows by administrators. In general, users within a user group, that is, those that are given the same role, are assigned the same set of privileges.

The following privileges may be assigned to RDC users:

1. **browse**
2. Browse batch loaded data
3. Update data
4. Update batch data
5. Update discrepancies
6. Verify CRFs
7. Approve CRFs (on a site basis only)
8. Lock
9. Unlock (on a site basis only)
**progression sequence**
The order that RDC uses to navigate to and open CRFs. There are three different modes available in RDC:

1. By patient
2. By patient, within Spreadsheet view (Classic only)
3. By CRF column

The specific sequence that is employed at any time is defined by the Progression to next CRF setting in the Preferences window. In Classic mode, the system uses the progression sequence when other progression settings are selected. In PDF mode, the browse sequence is invoked when you use the Previous and Next buttons.

**qualifying value question**
A question that differentiates between sets of identical questions. In a multi-section CRF, where the same section, containing the same set of questions, is collected more than once, a qualifying question is used in each such section. The purpose of the qualifying question is to elicit a unique response, called a **qualifying value**, which allows differentiation of the responses in the sections.

When you respond to a qualifying value question, you select from a discrete set of values that are specified in the question definition.

An example of a qualifying question is a multi-section CRF that collects vital sign data multiple times in a single visit. Each set of vital sign data comprises a section. Each section is differentiated by "time post dose" question. The result is a set of vital signs collected at specific times.

**qualifying value**
The value assigned to a **qualifying value question** that is associated with a CRF section. For multi-section CRFs, where each section includes a qualifying question, the qualifying value is used to differentiate between the sections.

**query**
1) A procedure that is run against a database with the goal of returning a subset of a data that satisfy the query criteria.
2) An industry term that is a synonym for the Oracle Clinical term, **discrepancy**.

**question definition**
The set of information that delineates what data a question collects. Among the information is:

- question name
- data type
- length
- lower bound
- upper bound
question group
A set of questions in a CRF that are related due to similarity or study protocol considerations.
An example of a question group is Demographics, which collects such data as: sex, race, and date of birth.
See also: non-repeating question group, repeating question

question name
The label that describes a question. It may be in the form of a question or it may simply be a word or phrase that serves as the prompt for a response.

repeating question
A question that usually consists of more than one response. The responses are generally situated in a single row and are referred to as a "repeating question row". For each response, there may be a default value, which is a system-provided value that entered automatically when you open the CRF. A repeating question is usually one of a set, each of which are distinguished by the initial response or by a question label.
An example is a question group titled, "Body Systems". Each repeating question row collects data about a different body system with three response values. The first response in each row identifies the part of the body, for instance, chest, or head. The next response in each row requires one of three values: "Normal", "Abnormal", or "Not Done". If the response to this question is "Abnormal", the third response, "Explanation", is required.
See also: question group, non-repeating question group

required CRF
A CRF that the protocol specifies as a planned CRF in a visit, for which data must be collected. Planned CRFs are analyzed when the system determines whether there are missing pages. The information in the CRF Column Header of optional CRFs is displayed in regular font to distinguish each from optional CRFs.
See also: CRF, optional CRF.

resolution
A type of discrepancy action that causes the status of the discrepancy to change from active to closed.
See also: discrepancy, discrepancy status, discrepancy action, routing, user role

resolution reason
A parameter associated with a discrepancy action that provides a sponsor-defined reason when a user closes a discrepancy.

response value
The value that is assigned to a data point. This term usually refers to fields in the Question area of a CRF.
See also: CRF, data point
role
See user role

routing
A type of discrepancy action that causes the status of the discrepancy to change from active to other for your user group and from other to active for a different user group.
See also: discrepancy, discrepancy status, discrepancy action, resolution, user role

S

save
An action that commits pending data changes to the study database.
In Classic mode RDC, there are two modes of save action: explicit and implicit. The former is a result of user action, for example, clicking the Save button. The latter is the result of a system action.
In PDF mode RDC, all save actions are explicit, initiated by the user clicking the Save button or choosing to save pending changes through the Save Edits? window.

save complete
In PDF mode, a save action that causes RDC to assign the CRF to the entry complete data entry status.

save incomplete
In PDF mode, a save action that causes RDC to assign the CRF to the entry started data entry status.

scope
In the Activity List window, a category or classification of a set of tasks; there are three scope levels that are available in the Activity List window:
1. study
2. site
3. patient.
These are listed and identified in the Navigation pane. When you select an item in the Navigation pane, the tasks that are associated with its scope are listed in the Task pane. Because only one study can be active in RDC at any given time, the study scope is listed once. The site scope is listed once for each site to which you have access for the current study. The patient scope is listed once for each patient to which you have access for the site.

search criteria
The settings that determine the CRFs that comprise the workset. These settings can be defined directly, through the Search window, or indirectly, through the Activity List window.

search phrase
The group of settings that, when taken together, define a searchable property. An example is, "search all CRFs named 'Vitals' for systolic BP values that are greater than or equal to 120,'." You use several drop-down lists in the Subsearch window to construct the search phrase.
Search window

A GUI component in RDC that allows a user to define directly the search criteria that is displayed in the RDC Spreadsheet. The Search window provides access to seven criteria, which are can be modified from default values, that the system combines to retrieves CRFs from the study database.

section

1) In RDC, a constituent of a CRF that is comprised of a set of related questions. Each CRF contains at least one section and may contain more than one. In Oracle Clinical terms, a section equates to a data collection module (DCM) (DCM).
2) In the Graphical User Interface - a part of a window or other informational feature that contains related information and/or data fields. It is often delineated by a descriptive label and a border that surrounds its related components.

For example, the different views that are available in the Summary Task tab are differentiated by the sections that comprise each view. In some cases, the sections are utilized in more than one view. Refer to the "Summary task tab" topic in the RDC User’s Guide for examples of sections.

section discrepancy

A user-generated discrepancy that is associated with a CRF section. There can be multiple discrepancies associated with a CRF section. This is the only type of discrepancy can be routed as an internal discrepancy.

In PDF mode, the system displays a section discrepancy bar along the right-hand side of the CRF. The bar is colored according the highest discrepancy that is associated with the section and it is present along the extent of the CRF section.

In Classic mode, the system displays "<Oper>" as an indication of the presence of a section discrepancy in the title bar of the classic data entry window when focus is in the discrepant CRF section.

See also: discrepancy, CRF section

session

The period that starts when you successfully log in to RDC and ends when you exit RDC. This is also referred to as an RDC session. It is constricted by limitations that are imposed by the following conditions:

- only one user name is granted access to a session
- the role and privileges assigned to the user name determine the patient data and functionality that is available within a session
- only one database can be accessed during a given session – if you want to access a different database, you must initiate a new RDC session
- only one study can be open at a time during a session, however, unlike the restriction on the active database, you can change to another study within an RDC session – if you want to access a different study you must close the current study and select another.
- only one book can be active at a given time, however, you can change to another book within an RDC session
- one or more sites can be active during a given session and the privileges assigned to the user name may vary from site to site.
setting
The value of a parameter in the Search window, which is a contributor to the value of a criterion. Usually, the default setting for a parameter is "ALL", which means that the parameter does not limit the data that is retrieved. When the value of a parameter is set to something other than "ALL", it generally limits the number of CRFs that are retrieved.
See also: criterion, parameter

sequence number
In the RDC Discrepancy task tab, the ordering number that is assigned to each discrepancy associated with the current CRF or CRF section (if the current CRF is multi-section). Discrepancies are listed in the List of discrepancies tab in numerical order, according to the sequence number. The number assigned to each discrepancy is not static. It is based on the following parameters: the current status, the timestamp, and the location of the response field within the CRF or section.

shortcut key
A key or key combination that allows you to implement a function in the application by using the keyboard.
See also: access key

site
A criterion that contributes to the generation of search criteria based on the sites to which the user has access. The criterion can have a value of either:
- <ALL> – includes all sites to which the user has access
- single site – limits search criteria to one site to which the user has access.

spreadsheet view tab
In the RDC Spreadsheet, these determine how the system displays the workset data. There are three different tabs that each provide a different view of the workset:
- Study
- Phase
- Visit
You select the type with the Spreadsheet view drop-down list that is located above the patient listings in the Spreadsheet. (Note that the presence of this component is dependent on a sponsor-specified configuration.)

subsearch window
A window that is accessed from the Search window that allows a user to change the values of parameters. Changes in the subsearch window affect a Search window criterion.

T

Task pane
In the Activity List window, the section, or frame, that comprises the right-hand portion of the window and presents a listing of activities, or tasks, associated with the currently selected node in the Navigation pane. When you click a task, RDC retrieves
the data necessary to complete the task from the study database and displays it in the RDC Spreadsheet.

See also: node, scope, Navigation pane

test mode
A method of using RDC during study design, prior to the initiation of the protocol. Under normal circumstances, RDC runs with Production mode active. This mode mirrors the look and feel of production mode but uses a separate set of tables to store the data.

timepoint
A significant event in the history of a CRF. Used as criterion when viewing the Audit Trail tab. Examples of timepoints include:
- creation date
- verification dates
- approval dates.

timestamp
A value assigned to a data point that provides a chronology for significant events during a study. Such events include: the date/time when a value was created, the date/time when a value was updated, etc.

Uniform Resource Locator

An Internet address that points to a specific resource on the World Wide Web by its location. The address is described in combinations of syntax and special characters that, when combined, represent a unique string. In general, URLs make-up a subset of URIs. Common URLs are those that point to Web pages or to FTP sites. The former are identified by an "http:" prefix string, the latter are identified by an "ftp:" prefix string.

univariate discrepancy
A discrepancy that is dependent on the value of a single data point. This type of discrepancy is usually when the value recorded for a response does not meet certain criteria that is deemed acceptable by the study sponsor.

See also: discrepancy, discrepancy management, manual discrepancy, multivariate discrepancy

unlock
a) A privilege that is assigned to a user to unlock a CRF. The privilege can only be assigned at the site level.

see also: privilege, lock

b) A process that allows a user with unlock privilege to assign to another user the capability to update a CRF that is in the locked status.

unlocked
A lock status that indicates a CRF may be updated.
**unplanned**
A designation attributed to any event or CRF that was not part of the protocol schedule or which occurs at a time other than was originally specified in the protocol schedule.

See also: **unplanned CRF**, **unplanned visit**

**unplanned CRF**
A CRF collected at a visit at which it was not planned, that is, it is not part of the case book.

**unplanned visit**
A clinical event which occurs that was not scheduled by the protocol.

**update**
1) A process or condition in which CRF data or information, which has previously been saved, is changed.
2) A **privilege** that allows a user to initiate data entry, update data, initiate discrepancies, and update discrepancies.

**update batch**
A **privilege** that allows a user to initiate or update a discrepancy in CRFs that are batch-loaded.

**update discrepancy**
A **privilege** that allows a user to initiate or update a discrepancy in CRFs that are not batch-loaded.

**URI**
An acronym for Uniform Resource Identifier.

**URL**
An acronym for Uniform Resource Locator.

**user group**
A set of users that are assigned to the same **user role**. (RDC)

**user role**
A database role that is granted to a user or **user group**.

In RDC, there are five default user roles, however, any given study database may include some or all of these, and may include sponsor-specific roles. RDC allows privileges to be assigned independently of user role assignment.

The roles that RDC ships with are:
1. Super User (SU)
2. Data Manager (DM)
3. Clinical Research Associate (CRA)
4. Site Coordinator (SITE)
5. Site Investigator (INV)

See also: **privilege**, **user group**
validation
An action that entails the initiation and processing of sponsor-defined procedures, in the case of multiple data points, or edit checks, in the case of a single data point, that analyze collected data and return an query, or discrepancy, for each data point that does not meet the criteria defined in the procedure. Such a data-generated discrepancy is also referred to as a validation error.

validation error
A condition associated with one or more data points that indicates the value does not meet the criteria defined in a question definition or validation procedure. It is equivalent to a data-generated discrepancy.

value
When used in the context of criteria and parameters, the choice that you assign to a parameter, which was chosen from a list of possible values.

verification history
A record, or set of records, associated with a CRF that lists each change in verification status starting with the initial verification action.

verification status
A designation that describes if a CRF has been verified. In RDC, there are four verification statuses:
1. not verified
2. verified
3. awaiting re-verification
4. verification undone

verification undone
A verification status that indicates the CRF was verified but subsequently the verification was undone. This status is equivalent to the not verified status, with the exception that a verification history exists for a CRF in verification undone status.

verified
A verification status that indicates the CRF has been verified by a user with the verify privilege.

verify
A privilege assigned to a user name that allows the user to alter the verification status of a CRF or a group of CRFs. The privilege can be assigned at the study level or the site level.

Note: The RDC documentation uses this default set of user roles to describe various functionality that is dependent on certain sets of privileges.
visit
A clinical event, which generally denotes the occurrence of a meeting between a patient and clinical staff at a study site. In the course of a visit, data related to the study is collected, which at some point is recorded and saved to the study database.

In RDC, a visit consists of one or more CRFs. By default, the system displays Spreadsheet Visit view in the RDC Workspace, which displays, for the current patient list, all of the CRFs collected for a single visit.

W

watermark
A visible feature that overlays a browse mode CRF in the PDF DEW. It displays the timestamp of the last modification and the words "BROWSE ONLY". The browse mode watermark is visible along the left-hand margin of each CRF page.

workset
The collection of patient data that is currently displayed in the RDC Spreadsheet. You use either the Search window or the Activity List window to select the data that you want to view in the Spreadsheet. This may include, but is not limited to, the following:

- patients
- CRFs, including entry, approval, verification, and lock statuses
- response data
- investigator comments
- discrepancies
- audit trail history

This term may also be called a dataset.
Accessible Data Only? column, 3-6
accounts
   create OPSS, 1-5
   requirements for batch jobs, 8-1
Add Review Status button, 3-8
anadestab.sql, E-3
analrtab.sql, E-3
ANALYZE command, 9-12
analrtab.sql, E-3
anaopatab.sql, E-3
anaopatab.sql, E-3
asynchronous PSUB jobs, 8-2
attributes
   extended, 7-25
auto fill, 5-6
auto skip, 5-6

batch jobs
   execution status, C-12
   troubleshooting, C-7
   user accounts for, 8-1
blocking jobs, 8-3
Browse mode
   access to, 5-3

CDS
   setting field display, 4-1, 4-2
check_opa_settings script, A-6
cnvstatus, 10-1
   running on UNIX, 10-1
   running on Windows, 10-2
codelists, setting up in a discrepancy database, 3-17
collection scripts
   anadestab.sql, E-3
   analrtab.sql, E-3
   analrtab.sql, E-3
   anaopatab.sql, E-3
   anaopatab.sql, E-3
   anarxctab.sql, E-3

comment
   operator, 5-3
Comparison Beep, 5-6
cost-based optimization, 5-3
cost-based statistics, 9-9
CRF information
   defined, Glossary-6
current study
   defined, Glossary-6
Custom button, 5-24
data collection instrument (DCI)
   defined, Glossary-7
data entry
   navigating during, 5-8
Data Entry Display Format, 5-6
data entry operator
   and privileges in Browse mode, 5-3
   for both first and second pass, 5-3
   hiding data from, 5-9
   in silent mode, 5-3
   notified of errors, 5-3
   notified of unenrolled patient, 5-3
data entry screen settings, 5-1
   and thesaurus questions, 5-3
   defaults for, 5-2
   levels for, 5-1
   list of, 5-2
   local database-level, 5-2
   modifying, 5-2 to ??
   overriding, 5-2
   precedence of, 5-4
   study-level, 5-4
   undefined, 5-2
   user-level, 5-2
data entry user preferences
   and date format, 5-6
   and non-updatable fields, 5-9
   defaults for, 5-5
layout, 5-5
list of, 5-6
data extract	enabling 3.1-style, 6-8
data extract views
generating, 6-7
Database Server
Windows
ropa_settings, A-2
database trace session, C-15
date
formats of, 5-6
DCF
placeholder graphic, 3-30
views, 3-31
DCF REPORT LABELS codelist, 3-30
DCF STATUS CODES codelist, 3-28
DCI Book
sequencing, 5-3
using, in data entry, 5-3
deadlock detected, C-5
Decrease Width button, 5-9
default study
defined, Glossary-8
DISC COLS2 codelist, 4-1
DISC_FLEX1 VALUES codelist, 3-10
DISC_FLEX2 VALUES codelist, 3-10
discrepancies, 10-1
resolution of, by data entry operator, 5-3
DISCREPANCY FUNCTIONS codelist, 3-3
DISCREPANCY REV STATUS CODE codelist, 3-8, 3-14
DISCREPANCY STATUS SITE, 3-22
discrete_val_grp_type_code, 7-23
files
protection, B-1
PSUB, 8-2
FLD RXCMCMCD RDCI DELETE codelist, 4-2
FLD RXCMCMCD RDCI KEY codelist, 4-2
FLD RXCMCMCD RDCM KEY codelist, 4-2
FLD RXCMCMCD RESPONSE codelist, 4-2
FTP file viewing protocol, 11-1
function privileges
defining, 3-3
G

gen_alter_partition script, 9-16
gen_create_part_table script, 9-8
gen_procs
syntax, 10-2
gen_views, 6-7
running on UNIX, 6-7
running on Windows, 6-8
genprocs
running on UNIX, 10-3
running on Windows, 10-3
granting
program access, 1-12
project access, 1-12
H
HTTP file viewing protocol, 11-1
I
imp_resp_param.dat file, 9-12
Import utility, 9-12
Increase Width button, 5-9
IND DISC COLS2 codelist, 4-1
indexing, 9-1 to 9-21
initialization error, C-14
installation reference codelist
DISCREPANCY STATUS SITE, 3-22
invalid objects
partitioning, 9-13
Items Not Displayed, 5-9
K
Kernel32.DLL error, C-14
key templates
domain, 6-4
L
launchps.sh file, E-2
layout
Log-In
editing function of, 5-7 to ??, 5-8, ?? to 5-8
editing instructions, 5-9
modifying RDCI window in, 5-8
modifying RDCM window in, 5-9
Index-3

modifying Smart RDCM window in, 5-9 to 5-10
layout definitions, 3-4
list_study_resp_cnt_part script, 9-15
locking
profile criteria, 3-6
Log-In
navigating during, 5-8
Log-In Layout Editor menu option, 5-8

M
Maintain Partition Mapping Tables window, 9-7
MANHD DISC COLS2 codelist, 4-2
Manual Discrepancy in Browse parameter, 5-3
master profiles, 3-5
MC CDS SORT ORDER codelist, 4-1
MC COLUMNS codelist, 4-1
menus
modifying, 2-8
More button, 5-24
MULTI DISC COLS2 codelist, 4-2

N
NLS_DATE_FORMAT, A-2
NLS_DATE_FORMAT environment variable, A-7
NLS_LANG, A-2
NLS_LANG environment variable, A-8
NOLOGGING clause, 9-9
non-blocking PSUB jobs, 8-2

O
objects
special database, 1-18
OCL MC PREFS codelist, 4-2
ocl_add_user.sql script, 1-5
OCL.DE_CONFIG, 7-10
OCL_STATE codelist, 3-17
OCL_STATE reference codelist, 9-3, C-13
OCLASCR user group
and SAS data extract views, 1-17
oclupg32to40opt5.sql script, 9-19
opa_settings
Windows back end, A-2
opa_settings script, A-4
checking for errors, A-4
opapps directory, B-1
opapps userid, E-1
operating system passwords, 1-20
optimization, E-3
Oracle accounts, 5-5
assign access, 1-12
deleting, 1-13
for studies, 1-12
revoking, 1-13
to Programs or Projects, 1-12

P
page dimension, 5-2, 5-4
partitioning, 9-1 to 9-21
compile_all_invalid.sql, 9-13
cost-based statistics, 9-9
database trigger, 9-13
defined, 9-4
disk space, 9-5
invalid objects, 9-13
merging, 9-7
percent free, 9-9
performance, 9-5
preparing for, 9-11
pruning, 9-20
redo log file, 9-9
storage clauses, 9-7
tablesspaces, 9-11, 9-17
 tuning, 9-20
passwords
auditing, 1-20
characters allowed, 1-18
operating system, 1-20
schema or role, 1-18
security, 1-20
setting up, 1-18
planned events
in the Smart RDCM window, 5-10
populate_part_map_table script, 9-10
privileges
of data entry operator, 5-3
procedures
deleting inactive, 10-4
production mode, C-6
.profile, 1-4
.profile file, 1-18
profile views, 3-5
profiles, 3-5
criteria, 3-5
locking criteria, 3-5
setting up, 3-5 to 3-9
protocols
for file viewing, 11-1
PSUB
and character sets, on Windows, A-2
and opa_settings on Windows, A-2
asynchronous jobs, 8-2
blocking jobs, 8-3
create log directory, 1-3
definition, 8-1
difficult permission, 1-4
files created, 8-2
non-blocking jobs, 8-2
checking, 8-3
troubleshooting, C-7
PSUB jobs
printing on UNIX, 8-11
PSUB process, 8-3 to 8-12
Q

Question Groups
repeating, 5-3
questions
attributes of, 7-25

R

RDC
customize workspace, 5-23
RDCI sort order, 5-6
RDCM, 5-8, 5-9, 5-10
RDCM window, 5-9
rdcp_client.sql
customize CRF Column Header, 5-23
rebuild_respt_index script, 9-19
Received DCMs
updating locked, 5-3
record type key
code_environment, A-5
db_code_pair, A-5
db_env_setting, A-5
opa_home, A-4
oratab_filespec, A-4
remote_db_home, A-4
tnsnames_filespec, A-4
recreating symbolic links, E-1
reference codelist
OCL_DE_CONFIG, 7-10
reference codelists
design, 7-2
installation, 7-2, 7-19
local, 7-2
OCL_STATE, 9-3
system, 7-2
registry settings, A-10 to A-17
relink_rxc.log file, E-1
relink_rxc.sh script, E-1
relinking server code, E-1
relocating Oracle Clinical, E-2
repeat default, 5-3
replacement parameters, and standard text, 3-35
REPORTSERVER reference codelist, 1-9, 1-10, 1-11
resolution type codes
resolution, of discrepancies, 10-1
RESPONSES table
validation status, 10-1
Responses table, 9-1 to 9-21
resptrig script, 9-13
Revert button, 5-9
review status codes
defining, 3-14
roles
creating new, 2-6
RXC_BATCH_QUEUE environment variable, A-8
RXC_BDL_DIR environment variable, A-8
RXC_DEBUG_BUFFER_SIZE environment variable, A-8
RXC_DISC_REP, 1-18
RXC_DISC_REP schema, 1-18
RXC_IMMED_QUEUE environment variable, A-8
RXC_INSTALL
and USER QUERIES table, 1-17
RXC_LOG, A-8
and PSUB log directory, 1-3
RXC_MAA, 1-18
RXC_MAA schema, 1-18
RXC_MAA_TAB_SPACE environment variable, A-9
RXC_NOW_STRING environment variable, A-9
RXC_PD schema, 1-18
RXC_PD schema, 1-18
RXC_PRINTER environment variable, A-9
RXC_REP, 1-18
RXC_REP schema, 1-18
RXC_SAS_BATCH_QUEUE environment variable, A-9
RXC_SAS_CONNECT environment variable, A-9
RXC_LANGUAGE environment variable for SAS data extract views, 1-17
RXC_USER
location for SAS data extract views, 1-17
RXC_USER environment variable, A-10
rxchkobj.sh utility, E-1
rxcdf.bmp, 3-30
rxcdelproc, 10-4
RXCLIN_MOD, 1-19
RXCLIN_MOD role, 1-19
RXCPROD
and execution permission, 1-4
using PSUB to create OS account, 1-2
RXCPROD account
PATH, 1-4
S

SAS data extract views
adding users on UNIX, 1-17
location, 1-17
SAS datasets, 1-17
SAS_VIEW directory, B-1
SASORA, A-10
schema
RXC_DISC_REP, 1-18
RXC_MAA, 1-18
RXC_PD schema, 1-18
RXC_REP, 1-18
RXCLIN_MOD, 1-19
script
login, 1-17
scripts
ocl_add_user.sql, 1-5
Second-Pass comparison failure alert, 5-3
seed data
updating manually, E-2
sequencing
DCI Book, 5-3
of RDCIs, 5-6
server code
relinking, E-1
SERVER_NAME reference codelist entry, C-14
SERVER_OS reference codelist entry, C-13

Index-4
set_pwd utility, 1-19
Smart RDCM window, 5-8, 5-10
SQL
  adding to profiles, 3-7
standard text
  adding replacement parameters, 3-35
  creating, 3-34
  editing, 3-36
  setting up, 3-34
statistics, cost-based, 9-9
Sub Type, and standard text, 3-35
sub-event
  number, in Smart RDCM window, 5-10
symbolic links, E-1
system accounts, 1-3
System malfunctions, C-6

T
tablespaces, D-1
  partitioning, 9-17
TEMP environment variable, A-10
test mode, C-6
text type, 3-35
Thesaurus Management System
  see also TMS
  tablespaces
  tuning
    partitions, 9-20
type codes, 10-1

U
UNC file viewing protocol, 11-1
Unenrolled Patient Alert parameter, 5-3
UNI DISC COLS2 codelist, 4-1
Univariate Beep, 5-6
Univariate Failure Alert, 5-6
Univariate Failure Alert parameter, 5-3
UNIX passwords, 1-21
Update mode, 5-3
update privileges
  and review status settings, 3-9
  setting for discrepancies, 3-8
user accounts
  creating on UNIX, 1-3
  creating on Windows, 1-3
genereal, 1-3
requirements, 8-1
user groups
  setup, 3-2
User menu, 2-8
USER_BV_JOB environment variable, A-10
USER_QUERIES table
  creating, 1-17

V
validation procedures
  generating, 10-2
validation status
  computing, 10-1
  variables, and standard text, 3-35
view mode icons, illustrated, 3-5
visit
  number, in Smart RDCM window, 5-10

W
wildcards
  to grant access to Programs and Projects, 1-12
Windows passwords, 1-21

X
Xhelp, 5-24
topic files, 5-26