

Oracle Clinical

Creating a Study



Release 5.2.2
F50701-01
February 2022

ORACLE®

Oracle Clinical Creating a Study, Release 5.2.2

F50701-01

Copyright © 1996, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	xxv
Documentation accessibility	xxv
Related resources	xxv
Diversity and Inclusion	xxv
Access to Oracle Support	xxv

1 Overview of Creating a Study

Global Library	1-1
Design	1-2
Easy Study Design	1-2
Standard Design	1-2
Definition	1-3
Definition Components	1-4
Data Entry and Report Layout Design	1-5
What the Layout Systems Have in Common	1-5
How the Layout Systems Differ	1-5
Character-Based Layouts	1-5
Graphic Layouts	1-6

Part I Designing Studies

2 Planning and Designing a Study

Maintaining Programs	2-1
Defining Projects	2-2
Assigning Compounds	2-2
Creating Organization Units	2-3
Maintaining Regions	2-3
Deleting a Region	2-3
Updating a Region	2-3

Maintaining Region Components	2-4
Maintaining Parent Regions	2-4
Maintaining Child Regions	2-4
Creating Planned Studies	2-4
Creating a Clinical Study	2-5
Maintaining Clinical Study Objectives	2-6
Maintaining Enrollment Criteria	2-6
Maintaining Study Termination Criteria	2-7
Creating Clinical Study Comments	2-7
Assigning Clinical Study Regions	2-7
Maintaining a Clinical Study's Historical Events	2-8
Reviewing Planning Details	2-8
Updating Clinical Study Statuses	2-8
Deleting a Clinical Study	2-9
Designing Response-Dependent Branching	2-9
Setting Up Partial Source Data Verification	2-10
Designing a Flexible Study	2-11
Enabling Flexible Study Functionality	2-12
Defining Intervals and CPEs for Flexible Studies	2-13
Using a Single DCI Book	2-14
Using Rules	2-14
Rule Triggers	2-14
Rule Targets	2-16
Using Oracle Clinical-Siebel Clinical Integration	2-17
Copying Clinical Studies	2-18
Replicating Clinical Studies	2-18
Creating Study Versions	2-19
Copy Logic for Clinical Study Versions	2-19
Planning Study and Site Enrollment	2-20
Creating a New Study Plan	2-20
Study Site Plans	2-20
Maintaining Clinical Study States	2-21

3 Easy Study Design

Overview of Easy Study Design	3-1
Requirements	3-1
Using Easy Study Design	3-2
Data Structure Compatibility	3-2
Querying a Study	3-2
Creating a Clinical Study in the Easy Study Design Window	3-2

Planning Study Intervals	3-4
Creating Periods	3-5
Resequencing Periods	3-5
Creating Sub-periods	3-5
Resequencing Sub-periods	3-5
Defining Clinical Planned Events	3-5
Create a Clinical Planned Event	3-6
Defining Treatment Patterns	3-7
Creating Patient Positions	3-7
Displaying Patient Positions	3-7
Deleting Patient Positions	3-8
Completing Easy Study Design	3-8

4 Sites, Investigators, and Patient Positions

Maintaining Site Records	4-1
Creating Site Records	4-1
Assigning Sites to a Study	4-1
Maintaining Investigator Records	4-2
Creating Investigator Records	4-2
Assigning Investigators to Study Sites	4-3
Maintaining Patient Positions	4-3
Creating Patient Positions	4-4
Creating Patient Positions with System-Generated Codes	4-4
Creating Patient Positions with User-Defined Codes	4-5
Marking Patients Eligible for Source Data Verification	4-6
Manual Set Up	4-6
Validation Procedure	4-7
Batch Load Job	4-8
Introducing a Patient SDV Plan in an Ongoing Study	4-9
Assigning Patient Positions to Sites	4-10
Assigning Patient Positions from the Study Sites Window	4-10
Assigning Patient Positions from the Patient Positions Window	4-10
Assigning Patient Positions by Range	4-10
Assigning New Blocks of Patient Positions	4-11
Changing Owning Locations	4-11
Retrieving Patient Positions	4-11
Deleting Patient Positions	4-12
Deleting System-Generated Patient Positions	4-12
Deleting Alpha Code Patient Positions	4-13
Enrolling Patients in a Study	4-13

Adding Patient Position Details	4-14
Batch Load Patient Enrollment Dates	4-15
Linking Patient Positions in Different Studies	4-16
Assigning Patient Positions to DCI Books	4-16
Assigning Patient Positions by Range	4-17
Assigning Patient Positions by Site	4-17
Assigning Patient Positions by Study	4-17
Treatment Assignments	4-18
Assigning Selected Treatment Assignments to a Site	4-18
Linking Treatment Assignments	4-18
Linking Treatment Assignments by Range	4-19
Linking Treatment Assignment Ranges to a Site	4-19
Blind Breaks	4-20
Creating, Reviewing, or Canceling Blind Breaks	4-20

5 Treatments and Schedules

Treatments	5-1
Maintaining Active Substances	5-2
Maintaining Drugs	5-2
Maintaining Single Treatment Regimens	5-3
Creating or Modifying a Single Treatment Regimen	5-3
Specifying Dosage by Range	5-4
Maintaining Non-linear Titration Steps for a Treatment Regimen	5-4
Maintaining Daily Dosage Details for a Treatment Regimen	5-5
Changing the Drug Assigned to a Treatment Regimen	5-5
Maintaining Combined Treatment Regimens	5-5
Creating a New Combined Treatment Regimen	5-6
Modifying a Combined Treatment Regimen	5-6
Sequencing Regimens in a Combination Treatment Regimen	5-7
Maintaining Patterns	5-7
Maintaining Treatment Patterns	5-8
Assigning Treatment Pattern Regimens	5-8
Displaying Drugs Assigned to Treatment Patterns	5-9
Copying Treatment Patterns	5-9
Maintaining Treatment Pattern Codes	5-9
Defining a Study Schedule	5-10
Maintaining Intervals	5-10
Maintaining Phases	5-11
Maintaining Periods	5-12
Maintaining Sub-periods	5-12

Resequencing Phases, Periods and Sub-periods	5-12
Assigning Planned Processes and Procedures	5-13
Maintaining Processes	5-13
Maintaining Clinical Procedures	5-14
Creating Clinical Planned Events	5-14
Using CPE Offsets	5-16
Examples of Offsets	5-16

6 Stratification and Randomization

Stratification	6-1
Maintaining Factors	6-2
Glossary of Terms for Factors	6-2
Maintaining Single Strata	6-2
Selecting a Factor	6-3
Selecting a Stratum	6-3
Maintaining Combination Strata and Combined Combination Strata	6-3
Assigning Strata to Combination or Combined Combination Strata	6-4
Displaying Strata Assigned to Combination or Combined Combination Strata	6-4
Maintaining a Study Stratification Factor	6-4
Selecting a Clinical Study Version	6-4
Maintaining Strata	6-4
Randomization	6-5
Some Concepts	6-5
Access	6-5
Interim Analysis	6-6
Randomization Privileges	6-6
Maintaining Blinding in a Study	6-6
Choosing a Randomization Type	6-6
Version-level Randomization	6-7
Strata-Level Randomization	6-7
Version Block-Level Randomization	6-8
Strata Block-Level Randomization	6-8
Deleting, Locking/Unlocking, and Verifying Randomizations	6-9
Batch Loading a Randomization	6-9
Downloading a Randomization	6-10
Validating and Loading a Randomization	6-12
Handling a Legacy Randomization	6-13
Randomization Maintenance	6-16
Replicating a Randomization	6-16
Mirroring Randomization	6-16

Maintaining Randomization Statuses	6-17
Creating or Revoking an Access View	6-17
Creating an Access View	6-18
Revoking an Access View	6-18
Cohort Views	6-18
Creating a Cohort View	6-18
Revoking Cohort View Grants	6-18
Managing Patient Treatment Assignments	6-19
Disclosing a Patient Position's Treatment Pattern	6-19
Changing the Latest Treatment Assignment to a Different Treatment Pattern	6-19
Replacing a Patient Position with Another Patient at the Same Site	6-20
Replacing a Patient Position with Another Patient in the Same Study	6-20
Linking a Patient Position to an Existing Treatment Assignment	6-20
Creating a New Treatment Assignment	6-21
Removing a Treatment Assignment	6-21
Linking Patient Treatment Assignments	6-21
Disclosing Patient Treatment Assignments	6-21

Part II Defining Data Collection Objects

7 Questions and Question Groups

Creating and Maintaining Questions	7-1
Creating a New Question	7-2
Assigning Question Categories	7-6
Extended Text Questions	7-7
Extended Text Question Usage Restrictions	7-7
Extended Text Question Definition Restrictions	7-7
Attachment Prefix	7-8
Database Storage	7-8
Assigning DVGs to a Question	7-9
Defining Details for Complex and Question Set Questions	7-10
Question Set Questions	7-10
Complex Questions	7-10
Defining Extended Attributes	7-10
Deleting a Provisional Question	7-11
Creating and Maintaining Question Groups	7-12
Creating a New Question Group	7-12
Assigning Questions to a Question Group	7-14
Defining Indicator Branching	7-15
Creating a Question Group by Copying	7-15

Deleting Provisional Question Groups	7-16
Managing Question Groups	7-16
Setting Question Group Status	7-16
Creating Lists of Values of Question Group Types	7-17
Creating and Using DVGs	7-17
Qualifying Questions	7-17
DVG Subsets	7-18
Kinds of DVGs	7-18
Alpha DVGs	7-18
Notes on Using Alpha DVGs	7-19
Thesaurus DVGs	7-20
Creating a Thesaurus DVG	7-22
Using Thesaurus DVGs with TMS	7-23
Creating a DVG	7-23
Adding Values to a DVG	7-27
Creating a DVG Subset	7-28
DVG Subset Example	7-29
Alpha DVG Subset Examples	7-30
Maintaining a DVG	7-30
Modifying a DVG	7-30
Changing the Status of a DVG	7-31
Deleting a DVG	7-31
Adding DVGs to Active Questions, Question Groups, and DCMs	7-31
Question Group Questions	7-31
DCM Question Group Questions	7-31
Applying New DVGs to Existing Data	7-32
Migrating DVG Changes into Layouts	7-32

8 Global Library Activities

Storing Objects in the Global Library	8-1
Maintaining Global Library DCMs	8-1
Copying Study-Level DCMs to the Global Library	8-1
Maintaining Global Library DCIs	8-2
Copying DCIs into the Global Library	8-2
Maintaining Global Library Procedures	8-2
Copying Procedures to the Global Library	8-3
Maintaining Global Library View Definitions	8-3
Copying View Definitions to the Global Library	8-3
Maintaining Domains	8-3
Limiting Access to Definitions Through Domains	8-4

Domains of Parent and Child Definitions	8-4
Modifying the Domains of Library Definitions	8-5
Creating Domains	8-5
Domains and Prioritization	8-6
Domains and Object status	8-6
Maintaining Domains	8-6
Creating Default Domain Lists	8-6
Overriding Default Domain Lists	8-7
Maintaining User-Level Domain Lists	8-7
Maintaining Study-Level Domain Lists	8-7
Using Domains within the Global Library	8-8
Querying a Library Object	8-8
Creating a Library Object	8-8
Modifying the Domain of a Library Object	8-9
Copying an Object in the Global Library	8-9
Adding and Locating Objects	8-9
Domain Selection for Data Objects in Studies	8-9
Domain Rules When Creating or Copying Objects	8-9
Adding and Locating Objects	8-10
Examples of Domain Use	8-10
Support Conversion and Current Standards	8-10
Support Project-Specific Standards Exceptions and Development	8-10
Remove Obsolete Objects to Allow Reuse of Names	8-10
Create a Special Domain for Testing	8-11
Maintaining Standards Affiliations	8-11
Using Affiliations	8-11
Locating Library Objects	8-12
Mapping Library Objects to Affiliation Types	8-12
Maintaining Affiliations	8-13
Maintaining Global Library Copy Groups	8-13
Copy Group Restrictions	8-13
Validation Rules	8-14
DCMs	8-14
DCIs	8-14
Procedures	8-15
View Definitions	8-15
Copy Rules	8-15
DCM Copy Rules	8-15
DCI Copy Rules	8-15
Procedures Copy Rules	8-15
View Definitions Copy Rules	8-15

Creating a Global Library Copy Group	8-15
Performing Copy Group Operations as Batch Jobs	8-16
Copying from the Global Library to a Study	8-16
Copying from Study to Study	8-16
Copying from Your Current Study to the Global Library	8-17
Performing Ad Hoc Copies	8-17

9 DCMs and DCIs

Defining DCMs	9-1
Generating Layouts from DCMs	9-2
Using Variations of the Same DCM	9-2
Using Provisional and Active DCMs	9-2
Creating a New DCM	9-2
Copying a DCM	9-3
Adding Question Groups to a DCM	9-4
Configuring Question Groups in a DCM	9-4
Adding or Modifying DCM Question	9-7
Configuring DCM Subsets	9-7
To Create a New DCM Subset	9-8
Configuring Qualifying Questions	9-8
To Associate a Qualifying Question with a DCM	9-9
Using Repeating Question Groups	9-9
Using Default Repeat Values	9-9
Defining Conditional Branching	9-11
Conditional Branching and Character-Based Layouts	9-13
Conditional Branching and Graphic Layouts	9-13
Using Normalized Lab Data	9-16
Defining Date Formats	9-17
Scheduling DCMs	9-18
Associating DCMs to CPEs	9-18
Schedule DCMs Form	9-19
Copying the CPE Schedule	9-19
Copying the DCM Schedule	9-19
Resequencing DCMs	9-19
Creating Character-Based Layouts	9-20
Creating Default DCM Layouts	9-21
Creating Additional Layouts	9-21
Editing DCM Layouts with the DCM Layout Editor	9-21
Parts of the DCM Layout Editor	9-22
Positioning Objects	9-23

Selecting Objects	9-23
Editing or Adding Text	9-23
Adding Boxes	9-24
Moving Objects	9-24
Arranging Objects	9-24
Deleting Objects	9-25
Resizing Objects	9-25
Controlling Spacing and Sizing of Objects	9-25
Changing Page Numbers of Objects	9-26
Sequencing Data Entry	9-28
Ordering by Row and Column	9-28
Ordering by Clicking	9-29
Ordering by Entering Sequence Numbers	9-29
Validating a Layout	9-30
Saving a Layout	9-30
Closing the Editor	9-30
Laid Out Versus Not Laid Out Objects	9-30
Using Undo or Revert	9-31
Using Data Entry Forms in Test and Production Modes	9-31
Character-Based Layouts in Test and Production Modes	9-31
Graphic Layouts in Test and Production Modes	9-32
Defining DCIs	9-32
Creating a DCI	9-32
Adding DCMs to DCIs	9-33
Defining DCM Pages	9-35
Copying a DCI	9-35
Marking a DCI as Critical for Source Data Verification	9-35
Deleting a DCI	9-36
Defining DCI Access at the Study Level	9-36
Defining Custom Reviews Required at the DCI Level	9-37

10 Defining DCI Books

Traditional and Enhanced DCI Books	10-1
DCI Books	10-1
Enhanced DCI Books	10-1
Support for Flexible Studies	10-2
Additional Features	10-2
Defining an Enhanced DCI Book	10-3
Defining Book Properties	10-4
Navigator: Building an Enhanced DCI Book	10-5

DCI Book Pages: Assigning DCIs to CPEs	10-7
DCI Book Page Information	10-8
Rule Trigger Information	10-9
Copying Pages for a CPE	10-9
Deleting Pages from a CPE	10-10
Numbering and Renumbering Pages	10-10
How Oracle Clinical Page Numbering Works	10-11
Renumbering Start Pages	10-12
Sample Numbering Schemes	10-12
Defining Interval Rules	10-14
Defining an Interval Rule Trigger	10-15
Defining an Interval Rule Target	10-16
Defining DCI Rules	10-18
Defining a DCI Rule Trigger	10-18
Defining a DCI Rule Target	10-20
Validating an Enhanced DCI Book	10-21
Validation Checks	10-22
Viewing Validation Warnings and Errors	10-24
Defining DCI Book Constraints	10-24
Calculating DCI and Interval Expectedness for Patients	10-26
Automatic Expectedness Calculation for Patients	10-26
Automatic Incremental Expectedness Calculation	10-26
Automatic Full Expectedness Calculation	10-27
When and How to Run Expectedness Calculation Manually	10-27
When to Run Expectedness Calculation Manually	10-27
How to Run Expectedness Calculation Manually	10-28
Data Entry and Expectedness	10-29
Data Entry in Remote Data Capture Onsite	10-29
Data Entry in Oracle Clinical	10-29
Assigning Patients to a DCI Book	10-30
Defining a Traditional DCI Book	10-30
Creating a DCI Book	10-31
Defining DCI Book Constraints	10-31
Using CRF Page Tracking	10-31
Enabling CRF Page Tracking for a Study	10-32
Defining DCIs to Track CRF Pages	10-32
Defining DCMs to Track CRF Pages	10-33
Defining DCM Pages	10-33
Defining DCI Books to Track CRF Pages	10-34
Creating the DCI Book	10-34
Defining DCI Book Pages	10-35

11 Annotating CRFs

Discussion	11-1
Definition	11-2
Requirements	11-2
Annotation Placement Styles	11-2
Procedures	11-2
Annotating a DCI Form	11-3
Procedure	11-3
Notes	11-3
Annotating all DCI Forms in a Single Study	11-3
Working with an Annotated CRF in Adobe Acrobat	11-4
Displaying Acrobat Toolbars	11-4
Saving an Annotated CRF	11-5
Opening a Saved Annotated Form	11-5
Methods of Organizing and Viewing Annotations	11-5
Using the CRF Annotation Tool Window	11-6
Access	11-8
Components	11-9
DCM Attributes Section	11-9
Components of the Field Annotations Tab	11-10
Components of the DCM Block Annotations Tab	11-11
Components of the Display Characteristics Section	11-12

12 Creating Graphic Layouts

Creating a DCI Form	12-2
DCI Form Layout Areas	12-3
Creating DCI Form Page Definitions	12-4
Defining Page Definitions	12-5
About Points	12-5
About the Starter Page Definitions	12-5
Sizing Margins	12-6
Setting Binding Offsets	12-6
Enabling DCI Forms	12-7
Configuring Study-Level DCI Form Settings	12-7
Configuring Local Studies	12-8
DCI Form Runtime	12-8
General	12-9

Graphic Layout Editor/Updater	12-10
Graphic Layout Generator - General	12-10
Graphic Layout Generator - DCMs	12-11
DCI Form Generation Defaults	12-11
Default Settings for Showing DCM Header Fields	12-12
Default DCM Header Field Prompts	12-13
Version Migration	12-13
Patient Data Report	12-15
Validation	12-16
Options in Group Verify dialog	12-16
Options in Group Approve dialog	12-16
Source Data Verification Plan Defaults	12-17
Configuring DVGs and DCI Modules	12-17
Configuring DVGs	12-18
Configuring DCI Modules' Graphic Layout Settings	12-18
Maintaining DCM Graphic Layouts	12-18
Generating Default DCM Graphic Layouts	12-18
DCM Graphic Layout Statuses	12-20
Maintaining DCI Graphic Layouts	12-21
Viewing DCI Graphic Layouts' Status	12-21
Viewing Status Values on the Graphic Tab	12-21
Generating Provisional DCI Layouts	12-22
Generating DCI Forms	12-23
Maintaining Form Layout Versions	12-23
Form Layout Version Rules	12-24
About the DCI Form Versions Window	12-24
Changing DCI Form Version Status	12-25
Changing a PDR's Page Definition	12-25
Migrating Data to New Oracle Clinical DCI Form Versions	12-25
Assigning Patients for Migration	12-25
Form Migration Examples	12-26
Maintaining Approvals and Verifications During Form Version Migration	12-27
Configuring Default Settings	12-28
Running Form Version Migration	12-29
Maintaining Form Layout Templates	12-30
About the Starter Form Layout Templates	12-31
Copying a Form Layout Template	12-31
Validating Form Layout Templates	12-31
Editing Layouts	12-32
Introducing the Editor	12-32
Choosing Unit of Measure	12-34

Setting Grid Options	12-34
Working with Layout Objects	12-35
About Input Objects	12-35
About Static Objects	12-36
Inserting Flex Fields	12-36
Selecting Objects	12-36
Spell Checking	12-37
Positioning Objects	12-38
Grouping Selected Objects	12-40
Freezing Selected Objects	12-40
Resizing Objects	12-40
Viewing and Modifying Object Properties	12-41
Formatting Masks	12-42
Changing Mask Formats	12-42
Resizing Masks	12-42
Formatting Comb and Box Set Masks	12-43
Formatting Label and Prompt Text	12-44
About Undo, Redo, and Revert	12-45
Setting Cursor Visitation Order	12-45
Ordering by Row and Column	12-46
Ordering by Clicking	12-46
Ordering by Entering Sequence Numbers	12-46
Validating Layouts	12-47
Editing DCM Graphic Layouts	12-47
Editing Blocks	12-48
Editing Repeating Question Groups	12-49
Handling Changes to DCM Definitions	12-50
Previewing DCM Graphic Layouts	12-51
Editing DCI Graphic Layouts	12-51
Copying DCI Layouts	12-51
Maintaining Pages	12-51
Adding Pages	12-52
Deleting Pages	12-52
Incorporating Changes to DCM Graphic Layouts	12-52
Sequencing DCM Header Visitation Order	12-52
Validating DCI Layouts	12-52
Editing Form Layout Templates	12-52
About the Template Editor's Workspaces	12-53
Sizing Form Layout Templates	12-54
Previewing Form Layout Templates	12-54
Sequencing Visitation Order in Form Layout Templates	12-55

Creating and Maintaining Conditional Blocks	12-55
Creating Conditional Blocks	12-56
Conditional Block Display	12-57
Nested Conditional Blocks	12-57
Invalid Conditional Block Conditions	12-58
Maintaining Conditional Blocks	12-58
About Conditional Blocks in DCI Form Versions	12-60
Troubleshooting Graphic Layout Problems	12-60
FRM 41213 Followed by a 771000 Error Message	12-60
FRM-41214 Followed by a 777100 Error Message	12-60
Nothing Happens When You Try to Generate a Provisional DCI Form	12-62

13 Copy Groups

Creating Copy Groups	13-2
Grouping Objects	13-2
Adding DCMs	13-3
Adding DCIs	13-3
Adding Procedures	13-3
Adding View Definitions	13-3

14 Data Extract

The View Builder's Structure	14-1
Global Library	14-2
About View Templates	14-3
About Key Templates	14-4
About Extract Macros	14-4
About Complex and Simple Questions	14-5
About Extended Question Attributes	14-5
Example of Associating Extract Macros to Questions	14-6
Definition	14-7
A Sample Workflow	14-7
Customizing Data Extract	14-8
Copying	14-9
Create Extract Macros	14-9
Simplified Extract Macro Workflow	14-9
Attributes of a SQL Macro	14-10
Example	14-10
Macro Data Type	14-11
Some SQL*Plus Functions	14-11

Some Examples of SQL and SAS Macros	14-12
Some Hints for SAS Users	14-13
Defining Extract Macros Using the Maintain Extract Macros Window	14-13
Changing a Macro Definition	14-15
Listing Available Functions in the Database	14-15
Creating Key Templates	14-15
Maintain/Query Key Templates	14-15
Customize/Query Key Template Columns	14-15
Order Specification for Data Viewing	14-15
Aggregation	14-16
Create Questions and View Templates	14-17
Associating Questions with Views	14-17
Maintaining View Definitions	14-17
Linking a View Definition to a DCM	14-18
Viewing the Data for a View Definition	14-19
View Model	14-19
Creating View Templates	14-19
Mapping DCM Questions to View Template Questions	14-19
Having Clauses	14-20
Copying a View Definition	14-20
Defining Study Sets	14-20
Creating Union Views	14-21
Creating a Cross-study Union	14-21
Troubleshooting Cross-study Union Views	14-22
Creating In-Study Union Views	14-23
Example of Union Views	14-23
Creating Where Clauses	14-24
Copying Where Clauses	14-24
Adding DISTINCT terms	14-24
About Joining Fast Extract Views	14-25
General Example	14-25

15 Lab Ranges

Establishing Units of Measure	15-1
Defining Lab Units	15-2
Defining lab test Question Units	15-2
Defining Unit Conversions	15-3
Testing a Lab Conversion Calculation	15-4
Defining Preferred Conversion Groups	15-4
Defining Standard Lab Ranges	15-4

Defining Textbook Ranges	15-5
Defining Preferred Copy Groups	15-5
Defining Lab Panels	15-6
Defining Labs	15-7
Defining Specific Lab Ranges	15-8
Age	15-9
Effective Date	15-9
Lab Range Records without Effective Start or Effective End Dates	15-9
Defining Lab Ranges Using the Maintain Lab Ranges Window	15-10
Defining Lab Subsets	15-10
Defining the Subset Name	15-11
Defining the Subset Range Values	15-11
Using Textbook Ranges	15-11
Copying Lab Ranges	15-12
Lab Assignment Criteria	15-13
Example of Lab Assignment Criteria Evolution	15-14
Applying Lab Assignment Criteria	15-14
Granting Privileged Update to a User	15-15
Resolving Problems	15-15
Missing Lab References in the RDCMs	15-15
Missing Data in Patient Positions	15-16
Batch Loading Lab Ranges	15-16
Loading Data	15-17
File Format	15-17
Example	15-17
Validating Data	15-17
Transferring Data	15-18
Example	15-18
Executing the Process	15-19
Running the Lab Procedures Re-execution Pre-Process Batch Job	15-19

16 Validation and Derivation Procedures

About Validation Procedures	16-2
Examples of Validation Procedures	16-3
Simple Validation Procedure	16-3
Validation Procedure with an Arithmetic Function	16-3
More Complex Validation Procedure	16-3
About Derivation Procedures	16-4
Restrictions on Derivation Procedures	16-5
Examples of Derivation Procedures	16-5

Overview of Defining a Procedure	16-5
How Procedures Work Internally	16-6
Cursors	16-7
Patient Information Cursors	16-7
Procedure Question Group Cursors	16-7
Structure of Procedure PL/SQL Programs	16-7
Example: Procedure with More Than One Procedure Question Group	16-8
Example: Procedure Containing One Procedure Question Group	16-9
Example: Procedure with Multiple DCMs	16-9
Sort Order	16-10
Sample Procedure Code, with Custom Code Locations	16-11
Viewing Generated PL/SQL Code	16-12
Special Processing	16-13
Correlation	16-13
Correlation on Event	16-14
Correlation by Qualifying Value	16-15
Correlation by Question Value	16-17
Restrictions	16-17
Setting Up Correlation	16-17
Aggregate Functions	16-17
Variable Names	16-18
Restrictions	16-18
Setting up Aggregate Functions	16-19
Aggregate Lag Functions	16-19
Variable Names	16-20
Restrictions on Aggregate Lag Functions	16-20
Using Aggregate Lag Functions	16-20
Lag Checks	16-21
Variable Names	16-21
Setting Up Lag Checks	16-22
Lag Check Examples	16-22
Where Clause Extension	16-24
Qualifying Expression	16-25
Custom Code	16-25
Procedure Variables	16-26
Question Variables	16-27
Aggregate and Lab Attribute Variables	16-27
Aggregate Variables	16-27
Lab Attribute Variables	16-27
System Variables	16-28
Standard Variables	16-28

Question Group Variables	16-29
DVG Variables	16-30
User Variables	16-31
Using Lab Ranges	16-31
Defining Validation and Derivation Procedures	16-33
Where to Start	16-33
Creating an Entirely New Procedure	16-33
Creating a New Version of a Procedure	16-33
Modifying an Existing Procedure	16-34
Procedure Definition Sequence	16-34
Naming and Describing a Procedure	16-34
Access Privileges	16-35
Defining a Procedure	16-35
Defining Procedure Question Groups	16-37
Performance Issues	16-37
About the Question Groups Window	16-38
Defining Procedure Questions	16-40
Defining Correlating Questions for Procedure Question Groups	16-42
Defining User Variables	16-42
Defining Procedure Details	16-43
Order Number	16-43
Continue If Discrepancy	16-43
Test Not Null Only	16-43
Detail Types	16-43
Externally Derived	16-44
Failure Type	16-44
Description	16-44
Expression	16-44
Message	16-45
Discrepancy Code	16-45
Using System Variables in Pre-v3.1-Style Procedures	16-45
Using Variables in V3.1-Style Procedures	16-45
Variable	16-45
Perform Detail Only If...	16-45
Discrepancy Test	16-47
Report	16-47
Report Order	16-47
Adding Custom Code	16-47
Testing and Activating Procedures	16-48
Generating and Compiling a Procedure	16-48
Editing Generated PL/SQL Code	16-49

Testing a Procedure	16-49
Testing a Procedure from Definition	16-50
Testing a Procedure from Test a Study	16-50
Assigning Procedure Status	16-50
Assigning Standards Affiliation	16-51
Executing Procedures	16-51
Executing a Single Procedure	16-51
Execute a Single Procedure from the PSUB Window	16-52
Execute a Single Procedure from the Procedures Window	16-52
Executing Batch Validation	16-52
Facts About Procedure Execution During Batch Validation	16-53
Batch Validating TMS Derivation Procedures	16-53
Procedures and Frozen-status Patients	16-54
V3.1-Style and Pre-V3.1-Style Procedures	16-54
V3.1-Style Procedure Enhancements	16-54
Converting to the V3.1 Style	16-55
Conversion Example	16-56
Creating a New Pre-V3.1-Style Procedure	16-57
Maintaining Patient Statuses	16-57
Populating the Patient Statuses Table	16-58
Example	16-59
Patient Synchronization	16-60
Procedure Definition	16-61
Example	16-62
Pre-Details Custom Code	16-63
Alternative Approaches to Error Reporting	16-64
Single Failure	16-64
Multiple Failure	16-64
Type Coercion and Comparison of Derived Values	16-65

17 Test Mode

Overview	17-1
A Typical Test Mode Workflow	17-2
How Test Data and Production Data Differ	17-2
Test Mode Limitations	17-3
Launching Test Mode Sessions	17-3
Setting Up Test Study Design	17-3
About Test Mode Synonyms	17-4
Creating and Loading Test Data	17-5
Creating Test Data	17-6

Batch Loading Test Data	17-6
Using Test Mode Batch Data Delete	17-7
Testing Definitions	17-7
Testing Validation Procedures	17-8
Testing Derivation Procedures	17-8
Batch Validating Test Data	17-9
Testing Layouts in Oracle Clinical	17-9
Testing Layouts in Remote Data Capture (RDC) OnSite	17-9
Testing in Debug Mode	17-9
Checking Validation Results	17-10
Testing Mass Changes	17-10
Testing Data Extract Views	17-11

18 Integrating with a Clinical Trial Management System

Enabling Patient and Subject Activity Integration	18-1
Enabling Integration at the Database Level	18-1
Overriding Default Integration Values for Individual Studies	18-3
Setting Up and Using Activity Completion Integration	18-3
Creating Activity Completion Criteria	18-4
Adding Activity Completion Criteria Details	18-6
Running the Track Activities Batch Job	18-8
Completion Criteria Processing	18-8
Completeness Logic	18-8
Completion Date Updates	18-8

A Flexible Study Design Examples

Basic Examples	A-1
Enabling a DCI within a Clinical Planned Event	A-1
Enabling DCIs Across Clinical Planned Events	A-2
Enabling Intervals Based on the Collection of Any Data	A-3
Enabling Intervals Based on a Data Value	A-4
Skipping to an Interval	A-5
Enabling the Next Interval	A-7
Complex Examples	A-8
Defining Multiple Paths Through a Trial	A-8
Enabling DCIs in a Conditional Interval	A-10
Defining Multiple Conditional Pretreatment Intervals and DCIs	A-11

Index

Preface

This guide describes how to translate clinical study design into Oracle Clinical and how to define data collection elements and systems for checking, deriving, and viewing data values.

This preface contains the following topics:

- [Audience](#)
- [Documentation accessibility](#)
- [Related resources](#)
- [Diversity and Inclusion](#)
- [Access to Oracle Support](#)

Audience

This manual is intended for the scientists, statisticians, and others responsible for designing clinical trials and implementing the design in Oracle Clinical.

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>.

Related resources

All documentation and other supporting materials are available on the [Oracle Help Center](#).

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through Support Cloud.

Contact our Oracle Customer Support Services team by logging requests in one of the following locations:

- English interface of Oracle Health Sciences Customer Support Portal (<https://hsgbu.custhelp.com/>)
- Japanese interface of Oracle Health Sciences Customer Support Portal (<https://hsgbu-jp.custhelp.com/>)

You can also call our 24x7 help desk. For information, visit <http://www.oracle.com/us/support/contact/health-sciences-cloud-support/index.html> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

1

Overview of Creating a Study

This manual covers the study design process from the initial organization planning stage to the active study stage.

Oracle Clinical organizes the study creation process into several subsystems. The subsystems correspond to a progressively maturing protocol. They parallel the stages of clinical study creation, from the general features common to all studies to features specific to modeling one protocol for one study.

However, the tasks in creating a study are iterative, nonsequential, and often performed by more than one individual or team, so Oracle Clinical's modular subsystems enable your company to develop different parts of a study concurrently. The general stages for creating any study in Oracle Clinical, (with the corresponding subsystems in parentheses) are:

- Program and project definition (the Plan subsystem)
- Study definition (Design)
- Defining data collection objects (Glib and Definition)
- Testing (Design)

The following sections describe the tasks required in each of the major and minor subsystems of Oracle Clinical to create a study:

- [Global Library](#)
- [Design](#)
- [Definition](#)

Global Library

The Global Library is the repository for all the required objects for creating and maintaining your organization's data collection, validation, and extract operations. You can create some of these objects in the Global Library. Others, you can create, test, and validate at the study level, then elevate to the Global Library. Once in the Global Library, they are available for use in other studies and at other locations.

Only people with certain database roles can access the Global Library screens, thereby retaining the integrity of the objects. For instance, by default, only RXC_SUPER and RXC_GL_FULL users have access to the Maintain Questions form (from the **Glib** menu, select **Questions**). (Oracle Clinical provides many roles, and your company can modify them or add new ones.)

Few of the tasks in the Global Library fit into simple linear sequences where you start with one task and progress straight to the end. Most of the tasks are iterative; many require cooperation with other users carrying out such system tasks as designing data collection; and all have a management component, such as determining Question status or managing reference codelists.

Design

The Design subsystem provides the facilities to model a clinical study after a study protocol. It provides a set of planning and recording facilities to handle diverse studies. No single study would use all its functions and options. Oracle Clinical provides two approaches to modeling a protocol: applying the Easy Study Design module, or applying the full complement of standard design modules.

During the early phases of study design, some details are often either unclear or disputed. For this reason, the Design subsystem enables you to design a study as a series of refinements. You can design the study's basic outlines early in its lifecycle, and refine it, exploring alternatives, until it matures into a fully operational study.

The following sections describe the major levels and components of this subsystem:

- [Easy Study Design](#)
- [Standard Design](#)

Easy Study Design

You can employ the Easy Study Design module—an integrated subset of the standard Design modules—to develop the fundamental details of a study to a testable stage of development. Oracle Clinical recommends that you begin creating a study with the Easy Study Design module, then clarify the details, as necessary, with the standard modules. If you have study planning privileges, you can create a planned study and a live study simultaneously. You also have the option to create or assign Intervals, planned events, treatment patterns, and patient positions.

Standard Design

Use the standard Design subsystem modules to refine the properties of a study originating in the Easy Study Design module, or to create new properties. The Design subsystem includes the following facilities:

Planning - At the planning level, you establish the organizational and geographic relationships for a new study and the program and project it belongs to.

Clinical Study - As the design of your study matures, you move it to the clinical study level. The clinical study record holds the basic details about the study.

Clinical Study Versions - A version is a variation on a proposed study design. Often basic study objectives and strategy are clear, but doubt remains about the effects that different stratifications, randomizations, or schedules of events might have on the results. Because a study has many variables, it is often difficult to predict which variation of a study would produce the clearest results. Clinical study versions contribute to solving this problem in two ways: (1) they facilitate building and comparing variations on one basic study design; (2) they permit altering one component, such as the stratification, without redesigning the entire study.

Planned Study Intervals - The planned study Intervals in a study make up the study timeline. A timeline is composed of *phases*, divided into *periods*, further divided into *sub-periods*. Each of these represents a planned study Interval.

Clinical Planned Events and Clinical Processes - A clinical process is a name for a type of visit in a study, such as a normal dosing visit. The clinical process is then used

to describe the clinical planned event (CPE). A clinical process can be scheduled to occur several times within a study.

Treatment Patterns - A study's treatment patterns—one or more treatment regimens in a particular sequence—specify the drugs and medical procedures. You can assign a patient to a treatment pattern and then follow the course of treatment specified by the treatment regimens during parts of the study.

Patient Positions - With patient positions you specify the number of screening patients, "normal" patients, and replacement patients in the study.

Blinding - Details of pattern assignment are usually kept secret until after completion of a study's inclusion/exclusion process. If an Investigator discovers details of the pattern for a patient, creating a blind break, Oracle Clinical provides facilities to record and report it.

Stratification - Strata are groups of people with a common characteristic. Stratifying your patient population ensures the soundness of the analysis of the study's results.

Randomization - Oracle Clinical provides four randomization styles you can apply to the distribution of treatments among the study's patients. The randomization modules include facilities for bringing randomization data into or out of the Oracle Clinical database.

Definition

Defining data collection for a clinical study within Oracle Clinical consists of building a database model of the study in the Oracle Clinical relational database. From this model, Oracle Clinical creates the logic necessary for data capture, validation, extract, and management. The main tasks you perform in the Definition subsystem are:

1. Determine how many and what kind of DCMs you need for the new clinical study. The initial work you do in the Design subsystem provides the structure of the study and a clinical study name.
2. Create the necessary new DCMs, or copy them from the Global Library or other clinical studies.
3. Populate the DCMs with Questions in Question Groups. Most of these objects are available for copy from the Global Library. A study designer with Global Library privileges can create new Questions.
4. Set the DCM schedule. Since each DCM corresponds to medical data collected at visits, you must associate each DCM intended for collection at a particular visit to the appropriate clinical planned event.
5. Customize each DCM's data entry form. Oracle Clinical creates forms automatically, but you can modify them to suit individual study needs or to simulate a Case Report Form.
6. Construct the DCIs, each of which establishes a correspondence between a Case Report Form (CRF), or batch data, and its DCM(s). Since a DCI may contain one or more DCMs and may run over more than one clinical visit, you must complete the construction of your DCMs before building DCIs. In the common case of a single DCM corresponding to a single DCI, Oracle Clinical automatically creates the DCI with the same name. You can optionally create one or more DCI books to structure the data entry process and to associate particular instances of a DCI with CRF page numbers. These DCI books consist of an ordered list of DCIs with their page numbers and visit identification. Later, during data entry, you can choose to step through the data entry screens in the order defined in your DCI book.

7. Define Validation and Derivation Procedures to standardize data entry. Validation Procedures test that data values meet your criteria, and Derivation Procedures create new data values.
8. Test the DCM data entry screens and Validation and Derivation Procedures with test data. You can then refine your objects if necessary, before using them in production.
9. Build or refine the study's data extract views. Condition your data to conform to the formatting requirements of other applications, such as SAS.

For more information, see:

- [Definition Components](#)
- [Data Entry and Report Layout Design](#)

Definition Components

The Definition subsystem includes the following data collection objects you create or modify:

Data Collection Modules - A data collection module (DCM) associates one or more related groups of Questions to a single clinical study visit. It is the equivalent of the section of a CRF that must be completed during a single clinical visit. A DCM can also define data such as laboratory results for electronic loading. Sometimes it is useful to define a DCM to collect data that continues across visits, such as the data in an adverse event log.

Data Collection Instruments - A DCI groups DCMs according to the study's CRFs.

DCI Books - You can optionally organize the DCI s (and the DCMs they reference) into DCI books, which associate the DCI s with CRF pages and organize the workflow of data entry.

CRF Page Tracking - You can set up new studies you create with Oracle Clinical to have an exact correspondence between a DCM, DCI, or DCI book, and the page number of a paper CRF.

Procedures - Write Validation Procedures to specify allowable entry values and Derivation Procedures to create additional values.

Validation Procedures - Validation Procedures help to ensure the accuracy of data collected in a study by checking for inconsistencies and discrepancies among Question responses. They also help find and analyze medically important relationships in collected data. Validation Procedures compare data collected for more than one response, and include simple comparisons of data from a single DCM as well as complex comparisons of data from multiple DCMs and visits. You can use arithmetic functions and manipulations of the data in tests. Validation Procedures can compare previously derived responses with user-entered data. When Validation Procedures uncover discrepancies, or values you have defined as inconsistent or out of range, Oracle Clinical records them in a discrepancy database.

Each Validation Procedure consists of one or more logically related test, or detail. You can specify the data objects (DCM Question Groups or Questions) and visit to test and write an expression using these items that, if true, should result in an entry in the discrepancy database.

Derivation Procedures - Derivation Procedures perform calculations on one or more user-entered responses to derive responses to derived Questions. For example, you

can calculate the response to the derived Question **Age** from the patient's date of birth and the date of the visit. The system generates a derived Question's response from a Derivation Procedure.

Labs - The Lab subsystem establishes and maintains the required information to track labs, lab ranges, lab assignment criteria, and units of measure and conversion for clinical laboratory test results.

Data Extract - Set up Oracle Clinical's extract modules to create data extract views, SAS datasets, and SAS Proc Prints.

Data Entry and Report Layout Design

Oracle Clinical provides character-based and graphic-based layout systems for creating data entry and report output. Features that are specific to either system include the terms *Character* or *Graphic* in their interface names.

For more information, see:

- [What the Layout Systems Have in Common](#)
- [How the Layout Systems Differ](#)
- [Character-Based Layouts](#)
- [Graphic Layouts](#)

What the Layout Systems Have in Common

The systems share the same data collection definitions. Both systems have a set of tools for generating and editing layouts. The systems' layout tools are similar.

It is possible to use both the character-based and the graphic systems simultaneously. However, if you enable a study for DCI forms, but continue to allow character-based data entry against the CRFs in a study, you create the possibility that data can be entered against the incorrect character mode layout, and have a different meaning when viewed through the DCI forms layout. You must guard against this possibility to ensure the integrity of your data. One way you might prevent this possibility is to block character mode Data Entry in RDC for such studies.

Even if you do not implement RDC Onsite, you can utilize the graphic system to create rich-format Patient Data Reports.

How the Layout Systems Differ

The graphic system allows you to create much richer layouts with more exact placement of objects. The character system lays out DCMs, but the graphic system lays out all the parts of a CRF, including DCI headers, footers, and DCMs.

Character-Based Layouts

For creating layouts for the Oracle Clinical Data Entry subsystem, configure your definitions to use a character-based layout editor. It is most suitable for trained data entry operators who can perform rapid data entry with no dependence on mouse actions or physical resemblance to CRFs. The instructions for maintaining the character-based system are part of [DCMs and DCIs](#)

Graphic Layouts

Oracle Clinical includes a Graphic layout system to create CRF renderings in PDF format. The renderings can look like—or even replace—your CRFs. They can be a single source for blank CRFs, CRFs with patient data filled in (Patient Data Reports, or PDRs), and for data entry through RDC Onsite. See [Creating Graphic Layouts](#)

Part I

Designing Studies

Part 1 describes how to model Oracle Clinical to match your organization's infrastructure and programs. It includes:

- [Planning and Designing a Study](#)
- [Easy Study Design](#)
- [Sites, Investigators, and Patient Positions](#)
- [Treatments and Schedules](#)
- [Stratification and Randomization](#)

2

Planning and Designing a Study

The Plan subsystem provides an organizational framework for research groups that conduct diverse studies. You can use it to arrange the Oracle Clinical database to model your group's associations: programs, projects, organizations, or regions.



Note:

Oracle recommends that you use the Easy Study Design facility to create a new study. See [Easy Study Design](#)

This chapter describes these Plan tasks:

- [Maintaining Programs](#)
- [Defining Projects](#)
- [Assigning Compounds](#)
- [Creating Organization Units](#)
- [Maintaining Regions](#)
- [Creating Planned Studies](#)
- [Creating a Clinical Study](#)
- [Designing Response-Dependent Branching](#)
- [Setting Up Partial Source Data Verification](#)
- [Designing a Flexible Study](#)
- [Using Oracle Clinical-Siebel Clinical Integration](#)
- [Copying Clinical Studies](#)
- [Planning Study and Site Enrollment](#)
- [Creating a New Study Plan](#)
- [Study Site Plans](#)
- [Maintaining Clinical Study States](#)

Maintaining Programs

A program represents the top level in the grouping hierarchy of studies. A program contains projects, which contain studies. In general, you create a program for each intervention. For pharmaceutical research, you map a program to each research compound. However, you can also use programs to control which users can access studies.

The program hierarchy organizes research around a drug. You can also use programs to control user access to studies. Typically, pharmaceutical companies assign a program for

each drug under investigation. Assign programs to planning-level studies in the Maintain Planned Studies window.

You can define and maintain programs from the Maintain Programs and Projects window: from the Plan menu, select Programs. To add a new program (or to modify an existing program):

1. Place the insertion point in an empty row. If there are no empty rows, select Insert Record (F6).
2. Enter a new, unique code for the program. This code identifies the program throughout Oracle Clinical.
3. Describe the program.
4. Save. By default, the system selects the Active box upon saving this record. You can make it inactive at any time by clearing the Active box.

The system stores the information you enter through these windows in the PROGRAMS database table.

The Maintain Programs and Projects window has two buttons:

- **Projects** – Displays the Projects window, listing all available projects.
- **Compounds Assigned to Program** – Displays the window where the compounds assigned to the selected program appear.

Defining Projects

You can group programs into projects that specify the dimension of the intervention under investigation. For pharmaceutical research, a project maps to an indication, or a formulation, of the drug. Each project is a collection of the individual studies for the drug.

To create, update, or delete projects associated with the current program, from the Plan menu, select Programs. From Maintain Programs and Projects, choose a program, click the **Projects** button, and then select the project record you want.

Assigning Compounds

To assign compounds to the selected program, click the **Compounds Assigned to Program** button. The Compounds for Program window opens, displaying the details of the compounds for the selected program.

To enter a new compound, click the **Insert** button. You can assign a compound to a particular program only once. Use the list function to choose from existing compounds. Choose the **Primary** field if this is the primary compound under investigation in this program. You can only designate one compound assigned to this program as the primary compound. This field defaults to selected when the first compound is assigned to the program. You can change this designation at any time.

The system populates the **Compound Code** and **Description** fields, according to the compound values set in the Maintain Active Substances window. There are two columns under the field title **Compound Code**. The first is the Compound Code, the second is the Compound Subcode. The subcode column has values only if your company uses subcodes to classify compounds.

As you add compounds, assign one of them as the primary compound. This affects Oracle Clinical reports and window displays, but you can change this designation at any time.

Creating Organization Units

Organization units designate the sponsors of clinical studies. To create and maintain organization units, access the Maintain Organization Units window. From the **Plan** menu, select **Organization Units**. The window automatically displays all available organization units. If you want to select an existing unit to update or delete, select the record from the list. If you want to create an organization unit, click the **Insert** button, and enter the name. The system enters the current date in the **Start Date** field. You can indicate the unit's expected end date, or leave it blank.

Maintaining Regions

In this module, you can create and maintain records of geographical regions and the structure of regions within regions. To reach the window, from the **Plan** menu, select **Regions**. Information you enter is stored in the Maintain Regions module in the Regions and Regions Components database tables.

For more information, see:

- [Deleting a Region](#)
- [Updating a Region](#)
- [Maintaining Region Components](#)

Deleting a Region

You can delete a region from either of the Maintain Regions windows. When you delete a region, the system deletes any region components linked to it (Made From or Part Of). If you try to delete an active region—that is, one assigned to a clinical study, you get an error message, "Deletion not allowed, assigned to Clinical Studies." You receive a similar message if the region is used elsewhere in the system.

Updating a Region

Occasionally you may need to update or change a region or one of its attributes. You cannot update a region in the Part of Region window or the Region Contains window. You can only insert and delete "part of" regions or "contained" regions in these windows.

To update a region:

1. From the **Plan** menu, select **Regions**.
2. Query for criteria about the region you want to update.
 - Click the **Clear Field** button to remove the field's contents, or just type over the existing text.
 - Enter new information in the field.
3. Save your work by clicking the **Commit** button.

Maintaining Region Components

This Maintain Regions window has two buttons:

- **Regions This One is a Part Of** – Lists the regions, including the current region.
- **Regions Within This Region** – Lists the regions composing the selected region.

You can use region components to further refine the organization of your regions. The Region Components window records the regions of which a region is a part. To open the Maintain Region Components window, from the **Plan** menu, select **Regions**, then select **Regions This One is a Part Of** button.

For more information, see:

- [Maintaining Parent Regions](#)
- [Maintaining Child Regions](#)

Maintaining Parent Regions

You can only assign or remove "parent" regions in the Maintain Regions This One is a Part Of window; to create new regions or modify them, click the **Exit** button to return to the Maintain Regions window. As it appears, this window lists the existing regions in order of region code. Note that you can insert and delete in this window, but not update.

Maintaining Child Regions

To maintain regions within this region, from the **Plan** menu, select **Regions**, then select the **Regions Within This Region** button. The Region <region name> window displays the regions this region currently includes. Use the list function to choose available values. You can only assign or remove "child" regions in this window; to create new regions or modify them, click the **Exit** button to return to the Maintain Regions window.

Creating Planned Studies

The Planned Studies utility allows you to assign general attributes that the clinical studies inherit. These attributes characterize studies at the design level. The Easy Study Design facility writes to a subset of these same database tables; you can modify the planning details of a study that originates in the Easy Study Design facility here.

Before you create a planned study, create affiliations: program, project, organization, and region. (See above.)

To create a new planned study, follow these steps:

1. From the **Plan** menu, select **Planned Studies**.
2. Enter a unique study identifier in the **Study** field of a blank line. If there are no empty rows in the window, use the Insert function. (From the **Data** menu, select **Insert Record**, or press F6.) You cannot change this field once you save your work.
3. Complete the **Name** and **Title** fields.

4. Assign an organization unit.
5. Assign one region to be the master country: the region that owns the study.
6. Enter the expected number of Investigators.
7. Use the list function to choose a design type. Design type is set in the EXP DESIGN TYPE CODE installation reference codelist.
8. Use the list function to choose a clinical phase. Clinical phase values are contained on the CLINICAL PHASE installation reference codelist.
9. Assign a program.
10. Assign a project. To modify the planning details of an existing study, select the appropriate field.

Creating a Clinical Study

The Maintain Clinical Studies window is the starting point for creating a new clinical study. This window names and describes a clinical study, consolidating many of its details. You can create the new clinical study provided projects, organization units, and regions already exist in Oracle Clinical's Planning subsystem. You can also augment the details of a clinical study originating in the Easy Study Design facility.

Follow these instructions to create a new clinical study:

1. Navigate to **Design, Studies**, then **Clinical Studies**.
2. Click the **Insert Record** icon in the toolbar.
3. In the **Study** field, select a study code from the list of available codes. (Designers set these values in the Planned Studies window. See [Creating Planned Studies](#).)
4. Enter, modify, or accept the short and long study titles.

The system populates the **Access** field according to its designated randomization access. Designers with randomization privileges can set the list's values in the Amend Randomization (from the **Design** menu, select **Randomization**, then select **Randomization Maintenance**, finally choose **Randomization Statures**).
5. Select the **Pivotal Study?** box if this is a pivotal study. **Pivotal Study** is an industry term indicating that failure terminates development of the drug.
6. Select the **FDA?** box if you expect to include the study's data in a submission to the FDA.
7. Save. When you exit, the system prompts you to confirm the creation. If you click **Yes** the system saves the record to the database and closes the window.

For more information, see:

- [Maintaining Clinical Study Objectives](#)
- [Maintaining Enrollment Criteria](#)
- [Maintaining Study Termination Criteria](#)
- [Creating Clinical Study Comments](#)
- [Assigning Clinical Study Regions](#)
- [Maintaining a Clinical Study's Historical Events](#)
- [Reviewing Planning Details](#)
- [Updating Clinical Study Statures](#)

- [Deleting a Clinical Study](#)

Maintaining Clinical Study Objectives

In the Maintain Clinical Study Objectives window, you can create, update, or delete descriptions of clinical study objectives. A study can have any number of objectives. The study can have more than one objective for an objective type. The system automatically queries the objectives for the selected clinical study when you open the window.

Select an objective to work with or use the Insert Record function to add an objective.

Choose the objective's type. You must choose the type of objective from values set in the OBJECTIVE TYPE CODE codelist.

If you have more than one objective for a type, you can rank their importance by assigning them numbers in the **Order** field.

Select the **Final Analysis?** field if the objective should be taken into account during the study's final analysis. Leave it deselected if the objective is for interim analysis.

Describe the objective.

Maintaining Enrollment Criteria

Use the Maintain Enrollment Criteria window to create, update, or delete the enrollment criteria of participants in a study.

A clinical study can have any number of enrollment criteria. Use the multi-record window to view all the enrollment criteria already assigned to a study. The single-record window contains more complete names for the fields. In multi-record mode, the system sorts the list by the criteria flagged as inclusion criteria first.

To create, update, or delete the enrollment criteria:

1. From the **Design** menu, select **Studies**, then choose **Clinical Studies**. The Maintain Clinical Studies window opens.
2. Query for and select a study, then click the **Enrollment** button. The Enrollment Criteria for Study window opens.
3. Choose an enrollment criterion, or use the Insert Record data function to add a new criterion. Copy or write a description of the criterion in the **Criteria Description** field.
4. Select the **Inclusion?** field (labeled **Is this one of the Inclusion Criteria?** in the single-record window), if you use the criterion as a requirement a patient must meet to be included in the study. Leave the field deselected if you use the criterion to exclude patients from a study. Currently, the system does not use this information.
5. Select the **Phone?** field (labeled **Used by Phone Randomization System?** in the single-record window), if the criterion helps define a phone randomization system.
6. Select the **Screening?** field (labeled **Used for Initial Screening?** in the single-record window), if you use the criterion as an initial screening of patients. A blank field indicates the criterion applies to final enrollment into the study.
7. Enter a grouping code in the **Group?** field (Grouping code in the single-record window), if a criterion is an element of a set of criteria.

The field is optional and accepts any value. The list of values shows the existing values for the study. Grouping criteria creates an OR logic condition. For example, to define a criterion for blood group A or B, you would group a criterion for blood group A, and a second for blood group B. You would join them with a code of A_OR_B.

8. Choose any factors to measure in the enrollment criterion. (A factor is a trait people share that may affect the outcome of a study.)

To clarify a trait's influence, and to reduce the likelihood of an improperly distributed factor, regulate its distribution to randomization groups with the system's strata forms. Designers with randomization privileges can set the code values for factors in the Maintain Factors window (from the **Design** menu, select **Strata**, then choose **Factors**). Set minimum and maximum values to factors that are part of a range (type R). Set a single value for a single-value factor (type V).

Maintaining Study Termination Criteria

1. From the **Design** menu, select **Studies**, then select **Clinical Studies**, and click the **Termination** button.
2. Select a termination criterion to work with, or use the Insert Record function to add one.
3. Describe the conditions that constitute an early termination.
4. Select the **Early Term?** box (**Is Criteria for Early Termination?**, in the single-record window), if the termination criterion disqualifies a terminated patient's data from being part of the study's final analysis. Leave the field deselected if an early termination does not remove any of the patient's collected data from final analysis.
5. Select the **Incl In Safety?** (**Include in Safety Analysis?**, in the single-record window), box if the termination criterion disqualifies an early-termination patient's data from being included in the study's safety analysis. Leave this field deselected if an early termination does not remove any of the patient's data from the safety analysis.
6. Select the **Incl In Efficacy?** (**Include in Efficacy Analysis?**, in the single-record window), box if the termination criterion disqualifies an early-termination patient's data from being included in the study's efficacy analysis. Leave this field unchecked if an early termination does not remove any of the patient's data from the efficacy analysis.
7. Choose any factors to measure in the termination criterion. Designers with randomization privileges can set the code values for factors in the Maintain Factors window (from the **Design** menu, select **Strata**, then choose **Factors**). Set minimum and maximum values to factors that are part of a range (type R). Set a single value for a single-value factor (type V).

Creating Clinical Study Comments

A study can have any number of comments, and the system displays all comments for a study after you select it. You cannot update or delete a study comment. From the **Design** menu, select **Studies**, then choose **Clinical Studies**, and click the **Comment** button. Use the Insert Record function to add a comment. Create comments to record information that have no specific attributes. The system does not use the comments.

Assigning Clinical Study Regions

You can assign the regions to a study in this window. Study planners can create the values for regions in the Maintain Regions window (from the **Plan** menu, select **Regions**), or, from the **Design** menu, select **Studies**, choose **Clinical Studies**, and then click the **Region**

button. The regions display by region code sequence. You can assign a region to a particular clinical study only once; however, a study can be conducted in more than one region and be expanded to include other regions than originally designated.

To maintain the regions for a clinical study, follow these steps:

1. Choose an assigned region, or use the Insert Record data function to add a new region.
2. Select the **Regulatory?** box if you include a region for regulatory reasons.
3. Select the **Marketing?** box if you include a region for marketing reasons.
4. Select the **Reporting?** box if this region is the region that appears in reports. Only one region can be the reporting region for a study.

Maintaining a Clinical Study's Historical Events

Historical events may be one of two kinds: event recording triggered by the system, or a user-defined event. Historical events triggered by the system include:

- Creation of this study
 - Creation of a study version
- Setting or changing the randomization access codes for a study
- Setting or changing the randomization access codes for a study phase
- Setting or changing the study status
- Setting the live study flag

Examples of possible user-defined historical events are producing a first draft of a study, or sending a first draft to another location for review. You can create or display clinical study history records, but you cannot delete or change a history entry. You can record dates for events you expect to occur in the future. The system populates the **Type** field, assigning user-generated records the value USERDEF.

To create or maintain historical event information, from the **Design** menu, select **Studies**, then select **Clinical Studies**, and click the **History** button.

Reviewing Planning Details

Open this window to display planning data about a clinical study. From the Design menu, select **Studies**, then select **Clinical Studies**, and click the **Planning** button. These details are all set in the Planning subsystem. The system sorts and displays phases in chronological sequence.

The buttons along the bottom of this window open several Design utilities. The following section describes these utilities and some of the main fields.

Updating Clinical Study Statuses

Set the study status to classify the study according to its developmental maturity. The timestamp for each change in status appears in the study's history. Use the list function to set a study's status. Your company maintains study status values on the STUDY STATUS TYPE CODE codelist (from the **Admin** menu, select **Reference Codelists**, then choose **Design Installation Codelists**, and query **STUDY STATUS CODE LIST**). You can also set randomization access in this window.

The information you enter provides a basis for creating clinical study history records.

To update clinical study statuses:

1. From the **Design** menu, select **Studies**, then select **Study Statuses**.
2. Select the study of interest.
Use the list and query functions, if necessary, to locate the study.
3. Enter the study status code.

Deleting a Clinical Study

Deleting a clinical study also deletes these elements associated with the study:

- enrollment criteria
- termination criteria
- objectives
- history
- regions
- comments
- versions and related records

To delete a clinical study, from the **Design** menu, select **Studies**, then select **Delete Study**. The Delete Clinical Studies window appears with a Clinical Studies window. The system displays all the current studies. Select the study you want to delete and click the **Delete** button. The clinical study disappears from the listing.



Note:

You receive an error message if you attempt to delete a clinical study with existing received DCIs; also, your company might place local safeguards.

Designing Response-Dependent Branching

You can design a CRF so that the response to a particular Question for a particular patient changes which additional Questions should be entered for that patient.

There are three different types of branching available:

- **Indicator branching:** If a patient's response to a Question should determine whether the remaining Questions in the same group should be collected for that patient, use indicator branching, which you define in the Global Library at the Question Group level. You define the first Question in the Question Group as an indicator Question and specify the response value that makes the remaining Questions in the Question Group collectible. If a value other than the indicator value is entered for a patient, the cursor moves next to the first Question of the next Question Group.

For example, define the first Question, SEX, to collect the patient's gender, with only two allowable values: MALE and FEMALE. If a patient's response to SEX is FEMALE, then the remainder of the Questions in the Question Group, which cover pregnancy and

menopause, become collectible for the patient. If the patient is male, these Questions are not collectible for the patient. The next Questions available in data entry are in the next Question Group in the DCM, if any.

- **Conditional branching:** If the initial, or *source*, Question should make Questions in the same or a different Question Group in the same DCM collectible, use conditional branching, which you can define in either the Global Library or in a study at the DCM Question Group Question level. You must specify a target Question for each allowable response. The logic can include a range of responses, defined as greater than, less than, or equal to a particular value.

For example, define a Question "Are you pregnant?" with a DVG containing three values: Yes, No, and I Don't Know. For the response Yes, define a target of the first Question in a Question Group about the pregnancy. For I Don't Know, define a Question "Is lab test scheduled?" as the target, with the Question "Date scheduled" nested. For No, either define a target of "Verification that lab test confirms" or do not define a target at all, in which case the cursor goes to the next sequential Question during data entry.

- **Branching to make Questions, visits, or Intervals expected or not expected for a patient:** If the initial, or *source*, Question should make Questions collectible that are included in a different DCM, or even a different CPE (Clinical Planned Event, or Visit) or Interval, use Enhanced DCI Books and their rules to define these relationships; see [Designing a Flexible Study](#) and [Flexible Study Design Examples](#).

 **Note:**

This type of branching is different from indicator and conditional branching in that the Questions made collectible by the response value may or may not be the next Questions answered in data entry—they may be in another CRF or visit entirely.

For example, define a Question Age. If the study has an age limit of 60 and the patient is 60 or under, the normal series of visits becomes expected for the patient (up to the next source Question, if any). If the patient is over 60, only the End of Study visit is enabled.

Setting Up Partial Source Data Verification

Oracle Clinical and RDC Onsite now support creating a risk-based source data verification (SDV) plan that identifies only a subset of data requiring verification in RDC.

An SDV plan can have one or both of the following components:

- A Critical Forms Plan identifies CRFs that need to be verified against the original data for all patients.
- A Patients Plan identifies patients in the study that must have 100% of their data verified. The percentage of patients requiring 100% verification can be different for different study sites. Patients can be individually selected for SDV or automatically selected based on either or both of these options:
 - A count of initial patients to be 100% source data verified.

- A default rate of automatic selection of patients for source data verification.

You set system-wide default SDV plan values in Oracle Clinical and *publish* default plans based on those values for all study sites. You can then make changes to the plan for any study site in RDC Onsite.

When a patient SDV plan is published, automated patient selection begins. With the appropriate privileges you can see in the RDC Onsite user interface which patients and CRFs have been selected for SDV.

You can introduce partial source data verification in an ongoing study.

For more information, see the following:

- [Marking Patients Eligible for Source Data Verification](#) .
- [Marking a DCI as Critical for Source Data Verification](#).
- [Source Data Verification Plan Defaults](#).
- **Publish Source Data Verification Plans for All Sites** in *Oracle Clinical Conducting a Study*. After you have selected default values, use this batch job to publish a plan for every study site. You can then modify it for different sites in RDC.
- **Schedule a Job to Manage Pending Patient Updates** in *Oracle Clinical Administrator's Guide*. Set up this batch job to run at regular intervals to update patients requiring source data verification.
- **Execute Pending Patient Updates** in *Oracle Clinical Conducting a Study*. The same update job can be run at any time. It resolves locking conflicts that may prevent work in some areas of Oracle Clinical.
- **Privileges for Partial Source Data Verification** in *Oracle Clinical Remote Data Capture Onsite Administrator's Guide*.
- **Source Data Verification Plan Page** in *Oracle Clinical Remote Data Capture Onsite User's Guide*.
- **Creating and Modifying a Study Site SDV Plan** in *Oracle Clinical Remote Data Capture Onsite User's Guide*.
- **Verifying CRFs** in *Oracle Clinical Remote Data Capture Onsite User's Guide*.
- **Patient Auto-Selection in Partial Source Data Verification** in *Oracle Clinical Remote Data Capture User's Guide*
- **Web Services for Runtime Modification of SDV Plans** in *Oracle Clinical Application Programming Interface Guide*

Designing a Flexible Study

If your trial is flexible—meaning that the protocol calls for decision points that determine which Intervals (groups of visits) and DCIs (equivalent to CRFs) are expected for each particular patient—use the Enhanced DCI Book feature to define the decision points and rules that determine Interval and DCI expectedness for each patient as he or she progresses through the trial.

Implementing a flexible study design in Oracle Clinical includes the following:

Schedule - Define the study schedule, including Intervals (Phases, Periods, and Subperiods) and visits, or Clinical Planned Events (CPEs), in the same way as for nonflexible studies in

Oracle Clinical, associating CPEs with Intervals and defining both CPEs and Intervals in chronological order. In the DCI Book, associate DCIs with CPEs.

Rules - Create a conditional schedule by defining *rules*. Identify the particular Questions and responses to them (the rule *trigger*) that determine which DCIs or Intervals (the rule *target*) are expected for a particular patient.

An Intervals or DCI that is the target of a rule is *conditional*; it is expected for a particular patient only if the data condition specified in the rule trigger is met for that patient. An Interval or DCI that is not the target of a rule—or dependent on another Interval or DCI that is the target of a rule—is *unconditional*. It is expected for all patients assigned to the DCI Book.

It is also possible to define a rule with an Interval as a target and an action called Bypass To; when the trigger condition is met for this type of rule the target Interval becomes expected for the patient and all intervening Intervals are skipped; they are no longer expected.

Calculating Data Expectedness for Patients - As sites enter data for a particular patient, Oracle Clinical recognizes the decision points you have defined as triggers in the DCI Book's rules, determines whether the value entered satisfies the defined condition and if so, makes the DCI(s) or Interval(s) defined as the target of that rule *expected* for that patient.

CPE and CRF Display in RDC Onsite - RDC Onsite displays only CRFs and CPEs that are either collected or expected for each particular patient. They may be expected because the corresponding DCIs and Intervals are defined as unconditional or because their condition has been met.

Whenever a user updates data that triggers a rule, the system recalculates expectedness for the patient and RDC Onsite adjusts the display of expected CPEs and CRFs as necessary. If a CRF that already has data collected becomes not expected due to a data update, RDC Onsite still displays the CRF icon but an "N" for **not expected** appears next to the CRF icon.

**Note:**

See also [Flexible Study Design Examples](#)

For more information, see:

- [Enabling Flexible Study Functionality](#)
- [Defining Intervals and CPEs for Flexible Studies](#)
- [Using a Single DCI Book](#)
- [Using Rules](#)

Enabling Flexible Study Functionality

To take advantage of flexible study functionality in the Enhanced DCI Book window and in RDC Onsite, you must first define the study as flexible. When you create a new study using Easy Study Design, the system prompts you for the Flexible setting.

You can also make an existing study flexible in the Clinical Study States window under Conduct, Security if the following conditions are met:

- No test or production data has been entered.

 **Note:**

If the study has an active DCI Book, you must set it to Provisional and back to Active before you can use it for data entry. If you have patients assigned to the book and you plan to use the book with RDC Onsite, you must also run the Calculate Expectedness job.

- The study's Page Tracking setting is not checked.

You cannot reverse the Flexible setting from checked to unchecked if the study has production data.

Defining Intervals and CPEs for Flexible Studies

As you plan your study schedule and how to define it, consider the ramifications for flexible studies:

- Interval and CPE order determine the default order of the DCI Book. You can define rules that allow a particular patient to skip certain Intervals—for example, to follow Treatment Arm A instead of Treatment Arm B—but you cannot use the DCI Book to change Interval or CPE order. A patient can only move from one Interval or CPE to another defined later in the study schedule. For example, if a rest period is required after Treatment A and after Treatment B, you must define it after both Treatment A and Treatment B in the study schedule; see [Figure A-8](#) for an example. See [Defining a Study Schedule](#) and [Creating Clinical Planned Events](#) for information on defining Intervals and CPEs.
- Using rules, you can make a whole Oracle Clinical Interval (Phase, Period, or Subperiod) that directly contains CPEs, expected (or skipped) for a particular patient if that patient's data meets conditions you define. Therefore define groups of CPEs that you want to enable or skip all at once as a single Interval.
- You can make a particular DCI expected for a particular patient if that patient's data meets conditions you define. Even if the Interval as a whole is expected for a patient, you can define rules in such a way that every DCI at every CPE is not necessarily required.
- You may want to give Intervals and CPEs nonnumeric names so that you can incorporate these names, or abbreviations of them, in DCI Book page numbers. In a DCI Book, each DCI, or Book page, must have a unique number. Normally no patient will have data collected for every DCI, so if you use purely numeric page numbers there will be gaps; see [Sample Numbering Schemes](#) for more information.

 **Note:**

When you create a study using Easy Study Design, the system automatically creates a Phase Interval called DEFAULT STUDY PHASE. If you are creating a flexible study you cannot use this default Interval and must explicitly create Intervals.

Using a Single DCI Book

Design flexible studies with only one DCI Book. You can define multiple treatment arms and many decision points and rules within a single DCI Book. Use the same DCI Book for all patients in study.

The only time you may want to have more than one DCI Book for a study is when the protocol is amended during the trial. In that case, you may want to copy the DCI Book, make the necessary changes, and reassign patients to the revised Book.

Using multiple DCI Books that include different portions of the study schedule is not recommended because there is no validation that the definitions of the two Books are consistent with each other. Migration between such unrelated books will be successful but the new expectedness calculation will reflect only the structure and rules of the new Book.

Using Rules

Each rule consists of a trigger—the data condition that must be met to trigger the rule, and a target—the DCI or Interval that is enabled when the condition is met.

To determine what rules your study requires, it may help to draw a diagram similar to those in [Flexible Study Design Examples](#)

[Validation Checks](#) covers information about rule definition restrictions.

For more information, see:

- [Rule Triggers](#)
- [Rule Targets](#)

Rule Triggers

In most cases, a trigger is one or more specified values that, when entered in response to a specified Question in a specified DCI at a specified CPE, constitute the condition that invokes the rule and makes the trigger DCI or Interval expected for the patient. It is also possible to specify that *any data* collected for a specified DCM constitutes the condition; see "Using the Any Data Trigger" under [Rule Triggers](#).

Questions used as a rule trigger must:

- have a Discrete Value Group (DVG) of type Internal or Thesaurus. The user must select the Question response value from the list of valid values specified in the DVG; see [Creating and Using DVGs](#).
- be of type CHAR
- not be a complex Question
- not be included in a repeating Question Group

Question response values cannot be numeric and they cannot be a range of numbers. However, if you need a numeric value or range of values to serve as a trigger, you can effectively do that using a Derivation Procedure.

Using Derivation Procedures

To convert a numeric value to a character value that can be used as a trigger, define a numeric Question to collect the numeric data, a derived Question to hold the character value, and a Derivation Procedure that takes the numeric Question response value as input, performs a calculation, generates a character value, and supplies the character value as the response to the derived character Question.

For example, if you want to require additional tests for patients whose systolic blood pressure is over 160, you can:

- Define numeric Question SBP to collect the patient's actual systolic blood pressure.
- Define a Derivation Procedure to take the value of SBP and determine whether it is greater than, equal to, or less than 160 and, if it is greater than 160, to return a value of Y and if not, to return a value of N.
- Define derived Question DSBP to hold the derived value—either Y or N. (You must define the derived Question before you define the Derivation Procedure.)
- Assign a Discrete Value Group (DVG) with the values Y and N to Question DSBP.
- Define the rule trigger to be the value of DSBP, in a specified DCI and CPE, equal to Y.
- Define one or more DCIs that collect the additional data—at one or more CPEs—as the target of the rule.

If a patient's systolic blood pressure has a value greater than 160, the Derivation Procedure generates a value of Y, the rule is triggered, and the specified DCIs become expected for that patient and visible in RDC Onsite.

The timing of the execution of the Derivation Procedure depends on the setting of its Exec Content field; if it is set to ON-LINE/DCM the Procedure is executed when the RDC data entry operator saves the value of Question SBP (if the RDCI status is high enough). If the Procedure generates a value of Y, the target DCI is immediately evaluated as expected for the patient and displayed in RDC Onsite.

See [Validation and Derivation Procedures](#) for more information on defining Derivation Procedures.

 **Note:**

The RDCI (collected data) must have a status of Batch Loaded or Pass 1 Complete or beyond. This might not be the case if the RDC user used the Save rather than the Save Complete action, or if data was entered through Oracle Clinical and Pass 2 was required for the study.

Data entered through Oracle Clinical or earlier versions of RDC Onsite triggers rules and makes Intervals and DCIs expected. However, expectedness is not interpreted by the user interface except in RDC Onsite.

Using the Any Data Trigger

The **Any Data** trigger satisfies the rule condition if the trigger DCI has any entered data at all. You do not need to define a particular Question as the rule trigger. As soon as any data is entered and saved for the DCI for a patient (if the RDCI has a status of Batch Data Loaded or Pass 1 Complete or beyond), the target Interval is enabled, or expected, for that patient.

For example, if the Rest or Followup Interval is not the target of any other rule, but must come after each treatment Interval, use **Any Data** in a DCI in the treatment Interval as the trigger to make the Rest or Followup Interval expected for a patient.

If you define a Rest or Followup Interval after each treatment Interval in the study schedule, it will be expected even without being the target of a rule. However, by defining the rule, the target Interval does not appear in RDC Onsite until they are logically expected.

Any Data is available only for Interval rules, and only with an action of **Enable**.

Rule Targets

A rule target can be one or more DCIs or Intervals. Rules that have one or more DCIs as the target are called DCI Rules. Rules that have one or more Intervals as the target are called Interval Rules.

DCI Rules

In a flexible study, all patients may not have the same assessments at every CPE. For example, a subset of pages may be required only for patients assigned to a specific Treatment Group, or detailed information may be required only if certain criteria are met, such as a detailed smoking history for smokers. Use DCI rules to make this possible.

For example, to make the detailed smoking history expected only for patients who smoke, define a DCI to contain the smoking history Questions, associate the DCI with the appropriate CPE(s) in the DCI Book Pages window, and define a DCI rule with the DCI as the target.

There are two types of DCI rules:

- **Within CPE:** The target DCI is in the same CPE as the trigger DCI, and is enabled in that CPE when the rule is triggered. (The same target DCI may be included in other CPEs, but other occurrences of the DCI do not become expected for a patient when the "Within CPE" DCI rule is triggered.)
- **Across CPEs:** The target DCI is enabled in every CPE where it exists, including the same CPE as the trigger DCI if it exists there, when the rule is triggered.

Note:

If only a few Questions need to be conditional on a particular Question response, you may prefer to use conditional branching instead of a "Within CPE" DCI rule; see [Defining Conditional Branching](#).

A DCI can be the target of only one rule. A DCI that is the target of a rule can also be the trigger for a rule, but it can act as a trigger only when it is expected and collected for a patient. See [Defining DCI Rules](#) for further information.

Interval Rules

In a flexible study there are multiple pathways through the trial for different patients. There may be entire Intervals (groups of CPEs) that are not required for some treatment groups, but are for others.

An Interval can be defined in the Oracle Clinical study schedule as a Phase, Period or Subperiod. Phases can contain Periods, and Periods can contain Subperiods. You can define Phases, Periods, and Subperiods as targets, as long as the rules do not conflict with each other (see [Validation Checks](#)).

An Interval can be the target of multiple Interval rules. It becomes expected for the patient the first time a rule targeting it is triggered.

If you do not define Interval rules, all Intervals are expected for all patients, though DCIs within Intervals may be conditional if they are the target of DCI rules. See [Defining Interval Rules](#) for further information.

There are two types of Interval rules:

- **Enable:** The target Interval(s), with all their unconditional CPEs and DCIs, become expected for the patient when the trigger condition is met. You can use an Enable Interval rule, for example, to make Phase I of a particular treatment arm expected, depending on the response to a particular Question; see [Figure A-4](#).
- **Bypass To:** The target Interval becomes expected for the patient when the trigger condition is met and all intervening Intervals—all Intervals that occur between the trigger and target Interval—are not expected. Any remaining expected DCIs in the trigger Interval are still expected.

You can use **Bypass To** Interval rules, for example, to handle patients who discontinue prematurely from the trial, using a Question as straightforward as "Will the patient continue in the trial?"; see [Figure A-5](#).

A **Bypass To** Interval rule can have only one Interval as a target, and that target cannot be **Next Interval**. Bypass rules take precedence over Enable rules of all types (DCI—Within and Across—and Interval) if there is a conflict.

In **Enable** Interval rules, you can specify **Next Interval** as the target instead of a named Interval. **Next Interval** is defined as the Interval containing the next CPE in ascending CPE number order that is in an Interval different from the one that contains the trigger DCI.

Using **Next Interval** has the following advantages:

- It reduces the number of rules required when the same trigger condition may appear repeatedly and the target is always the next Interval; for example, in oncology studies where patients continue on cycles of treatment until the disease progresses or they respond to treatment. You can define a single rule with a **Next Interval** target to move each continuing patient from one cycle to the next. See [Figure A-7](#) for an example.
- All potential Next Intervals—all Intervals whose preceding Interval contains a the trigger of a **Next Interval** rule—are treated as conditional by the system; RDC Onsite does not display for a particular patient them unless they become expected.

The target Interval may be an Interval within the trigger Interval; for example, if the trigger Interval is a Phase that contains CPEs directly and also contains Periods, a DCI in a CPE contained directly in the Phase could trigger the next Period in the same Phase.

Using Oracle Clinical-Siebel Clinical Integration

If you are using both Oracle Clinical and Siebel Clinical for a clinical study, you can integrate the two systems to automatically pass certain data and metadata from one system to the other in order to:

- Signal the completion of visits and activities.

- Simplify study setup in Oracle Clinical and Subject setup in Siebel Clinical.

You can also integrate Oracle Clinical with other clinical trial management systems. For further information, see [Integrating with a Clinical Trial Management System](#)

Copying Clinical Studies

You can copy the details of a clinical study. This is limited to the Clinical Studies, Clinical Study Enrollment Criteria, Clinical Study Termination Criteria, Study Comments, Clinical Study History, and Study Regions tables. The copying function does not copy the Study Design. Use the Copy a Clinical Study Version function to copy the actual Design.

This utility is useful for importing the attributes of an existing study into a new study. Use this utility to reduce the effort required to design a new study when a similarly structured study already exists. This differs from copying a study version, where the system copies all attributes of another version exactly as is, including randomization. Before you can copy the attributes of one study to another, the target study must—at least—be at the planned level of development.

Select or query the record you want. You cannot insert, delete, or update in this window. Click the **Copy Study** button to view the Copy Study window, where you specify information about the new study.

1. From the **Design** menu, select **Studies**, then choose **Copy Clinical Studies**. The Clinical Studies window opens, listing only the studies available for copying attributes.
2. Enter—or query—the study code for the study to copy.
3. Click the **Copy Study** button.
4. Use the list function to choose a target study—the study to acquire the attributes of the study you selected in the previous window. Note: There is no recourse from the following step. Cancel if you are not certain you are selecting the correct target study.

For more information, see:

- [Replicating Clinical Studies](#)
- [Creating Study Versions](#)

Replicating Clinical Studies

A clinical study can be replicated from one location to another. Only one location can maintain a study design but other locations can have read-only copies. Use this function to replicate a study design from its owning location to yourself. See the "Distributed study conduct" chapter in the *Oracle Clinical Administrator's Guide* for more information about replication.

To replicate clinical studies, from the **Design** menu, select **Studies**, and choose **Replicate Clinical Study** for the Replicate a Clinical Study window. The system stores information from this form in many tables, including Clinical Studies, Clinical Study Objectives, Clinical Study Enrollment Criteria, and Clinical Study Versions. Select the record you want by entering the requested codes. The fields are list-enabled.

Creating Study Versions

A study typically passes through several iterations of refinement and can have several potential designs, depending on the parameters included in its protocol. These iterations of refinement can be modeled in Oracle Clinical as clinical study versions, each of which may be thought of as a potential protocol. The attributes that affect the design are held at the version level. When the protocol is finalized, you choose one of these versions to be the "live" version. Once live, you cannot change it or any other version of the study back to "not live." However, the live version can still be modified and developed.

Note:

RDC does not work with studies that have multiple study versions. Errors include (but are not limited to):

- selecting and using a patient when a patient with the same patient code exists in two or more study versions
- selecting and using a clinical planned event when a clinical planned event with the same name exists in two or more study versions

To use RDC for a versioned study, delete the clinical study versions that are not live: Delete any associated items (such as patient positions and clinical planned events), associated with the clinical study version.

Setting the live flag on a study makes that version the one to use throughout the study. The study timeline, treatment patterns, stratification, and randomization can all be defined separately for each version of the study design.

From the **Design** menu, select **Studies**, then select **Study Versions**. The Maintain a Clinical Study Version window appears, listing all current clinical study versions appropriate for your privileges. The system displays the clinical study and version, if any, of your current session. You can select another, and you can query to modify the list.

For more information, see:

- [Copy Logic for Clinical Study Versions](#)

Copy Logic for Clinical Study Versions

Oracle Clinical's copy logic distinguishes between copying a clinical study version within a clinical study and copying from one clinical study to another. When you copy a version within a study, you are usually creating a new version, a checkpoint in the study design process, or recording a change to the protocol. Oracle Clinical produces an exact copy of all attributes of the version, including randomization.

However, when you copy a version from one study to another, you are usually designing a new study. For this reason, Oracle Clinical performs the copy so that it looks as if the data is new. The system resets the creation user and timestamp to the current user and date, and the modified user and timestamp values are set to null. The system also sets the blinded attributes of a randomization to their default values.

Planning Study and Site Enrollment

You can project the enrollment expectations for studies and their study sites within the Maintain Study and Study Site Plans window. Each plan records a planned start date, planned enrollment target, and planned target date (date for achieving the enrollment target). You can modify plans until you finalize them by selecting the **Baselined?** field. To change a baselined study's plans you must supersede them with new plans. Furthermore, once enrollment for a study or study site has begun, you can no longer modify the planned start date in original or later plans.

For tracking actual study progress against the plan, use the study tracking reports. Specify the plan for the study from the Maintain Study and Study Site Plans window. From the **Design** menu, select **Studies**, then select **Study and Site Plans**. Select a study and modify the enterable fields as necessary. If you want to permanently record the plan for a study, select the **Baselined?** box.

The Study Plans window contains information about the Enrollment Plan for the study, including the Planned Start Date, Planned Target Date, and Planned Enrollment. Check that the Planned Target Date is greater than the Planned Start Date and that the Planned Enrollment is greater than zero.

Oracle Clinical calculates value of the **Actual Enrollment** field; you cannot manually update the field in this window. Oracle Clinical determines the actual enrollment based on the number of patients with a status of **ENROLLED** as calculated by the Patient Status Derivation Packaged Procedure. The system also maintains the **Plan #** field, incrementing it by one each time you create a new plan.

Oracle Clinical also maintains the Effective Date and Superseded Date. The current plan always has an Effective Date reflecting the time when the plan was created, and a Superseded Date of 15-AUG-3501. This date far in the future indicates that the plan has not been superseded, and is in fact the current plan. A plan with a Superseded Date of any other value was superseded at that time.

Creating a New Study Plan

Create a new study plan when you have already baselined (permanently recorded), a study plan, but need a new one to meet changed conditions. To create a new study plan, from the **Design** menu, select **Studies**, and choose **Study and Site Plans** for the Maintain Study and Study Site Plans window. Click the **Create New Plan** button. A new plan record appears, which inherits many of the values of the previous plan. The new record also has an incremented plan number (Plan #), a newly calculated Actual Enrollment value, and new Effective and Superseded Dates.

Each time you create a new plan, the system resets each of these fields to the current date: the **Superseded Date** for the old plan and the **Effective Date** for the new plan. It sets the **Superseded Date** for the new plan to 15-AUG-3501. Viewing the **Effective and Superseded Dates** fields of each of the study plans reveals the time periods over which the prior plans were effective.

Study Site Plans

Use this window to plan the enrollment details for a site. To maintain the records for a planned study site:

1. From the **Design** menu, select **Studies**, then choose **Study and Site Plans**, and click the **Sites** button. The Study Site Plans window appears.
2. Create site plans by clicking the **Create Initial Plans** button. Defaults from the values supplied in the study plan give the initial values for the site plans.
3. When new sites are added to the study after the planning process has started, create initial plans for the new study sites by clicking the **Create Initial Plans** button. As a result, Oracle Clinical creates plans only for study sites that do not yet have plans, and leaves existing plans untouched.

The actions you can perform in the study sites window are identical to the actions you can perform at the clinical study level.

Maintaining Clinical Study States

You can set or review several study-level settings that control data definition and collection in the Maintain Study States window. From the **Conduct** menu, select **Security**, then select **Clinical Study States**. These are the settings:

- **Enable Data Entry?** This box must be selected to enable data entry for the selected study.
- **Allow 1st Pass from Login?** Select this box to enable data entry operators to directly access the first-pass data entry window after completing a CRF's login information.
- **Frozen?** Indicates if the study is frozen at the data level. You perform this operation by running a batch job. (From the **Conduct** menu, select **Security**, then select **Freeze Study**.) This is a view-only field.
- **Second Pass Required?** Select this box to force a study to complete second-pass data entry to complete data collection.
- **Enable CRF Page Tracking?** Select this box to use Oracle Clinical's Page Tracking utility in a paper-based study. See [Using CRF Page Tracking](#).

You cannot select both **Enable CRF Page Tracking?** and **Flex Study Enabled?**.

- **VB Enabled?** Select this box to enable the Data Extract subsystem's View Builder utility. (See [The View Builder's Structure](#).)
- **DCI Forms Definition Enabled?** Select this box to enable DCI Forms definition for the selected study in the Definition subsystem. (See [Enabling DCI Forms](#).)
- **DCI Forms Entry Enabled?** Select this box to enable collecting data in RDC Onsite.
- **Flex Study Enabled?** Select this box if your study has an *adaptive* (or *flexible*) protocol. If you check this box, you can use the Enhanced DCI Books window to define rules that make specified CRFs and Intervals expected for a particular patient based on data collected for that patient at specified points in the trial. You cannot select both **Enable CRF Page Tracking?** and **Flex Study Enabled?**.

You can change this setting for an existing study if no Test or Production data exists (test data can have been entered and then deleted). If you change a study from flexible to nonflexible, the system ignores any rules defined for its DCI Book and you can no longer see them because the rules button in the Maintain Enhanced DCI Book window is inactive.

 **Note:**

The Enhanced DCI Books window includes Start Page renumbering, copying, and navigational enhancements that are helpful in any study. Studies with **Enabled CRF Page Tracking?** checked cannot use the Enhanced DCI Books window; they must use the original DCI Books window.

Studies with neither **Enable CRF Page Tracking?** nor **Flex Study Enabled?** checked can use Enhanced DCI Books, but the rule definition required for a flexible protocol is available only for studies with **Flex Study Enabled?** checked. See [Traditional and Enhanced DCI Books](#) for further information.

- **Key Template** Specifies a key template for the Data Extract View Builder.
- **Template Domain** Specifies the Key Template's Domain.
- **Surrogate Key Enabled?** To use Data Reload from Oracle Health Sciences Life Sciences Data Hub for Data Extract views in Oracle Clinical, activate this setting. This setting is available only if View Builder is enabled for the study. The surrogate key adds a field to your views that is guaranteed to be unique across all RDCM repeats in a given view and which does not change with data updates. After activating this setting, the new surrogate key appears in Data Extract view after you regenerate them.

 **Note:**

The new surrogate key does not appear in SAS views.

Also the OC function `vb.getViewColumnsTable`, which returns the columns in a Data Extract view, does not return the surrogate key column (which includes the Y or N value for this setting) in order to avoid possible impact on activities that use the function with SAS views.

You can also set a number of default study-level data entry settings from the Maintain Study Configuration window. From the Maintain Study States window, choose **Special**, then select **DE Config**, local database. The database-level settings are in effect unless you change them here; explicitly defined study-level settings override local settings, and user-level settings override both study and local settings. For many of these options, you can choose to enable them, disable them, or leave them in a "Not Set" state. In the Not Set state, the settings at the local database override settings here at the study level. These settings include:

- Second Pass Comparison Failure Alert
- Manual Discrepancy in Browse
- Resolve Discrepancies in Data Entry
- Privileged Update
- List of Values for Thesaurus Questions
- Univariate Failure Alert
- Initiate DE Session Using DCI Book

- Unhonored Patient Alert
- Prevent Second-pass Entry by First-pass Operator
- Browse Accessible Data Only
- DCI and DCM Date Required
- Default Height for Data Entry Page in DCM (character-based system)
- Default Width for Data Entry Page in DCM (character-based system)

For more information on managing data entry configuration and user preferences, see the *Oracle Clinical Administrator's Guide*.

3

Easy Study Design

Easy Study Design is a subset of the standard design modules that creates only a study's essential elements. In one Oracle window you can create a live study version. To open the window, from the **Design** menu, select **Studies**, then select **Easy Study Design**.

For more information, see:

- [Overview of Easy Study Design](#)
- [Creating a Clinical Study in the Easy Study Design Window](#)
- [Completing Easy Study Design](#)

Overview of Easy Study Design

Oracle Clinical's Easy Study Design window provides a simplified method to create a live version of a study. From this single window you can access streamlined versions of the main design utilities: planned study, clinical study, Intervals, events, patterns, and patient positions. After completing these essential components, you can refine the details of your study with any of the standard design forms.

For more information, see:

- [Requirements](#)
- [Using Easy Study Design](#)
- [Data Structure Compatibility](#)
- [Querying a Study](#)

Requirements

You can create a new study using Easy Study Design only under one of the following conditions:

- No planned study and no clinical study exists with the study name.
- A planned study exists with the study name, but no clinical study exists with the study name. (To plan a study, from the **Plan** menu, select **Planned Studies**.)
- A clinical study with the study name exists, but no versions exist.

After initial setup, you can use any of the standard design forms to add or modify other objects for your study. You can modify any object that originates in Easy Study Design with the equivalent standard design window, with these limitations:

- You can add new objects created with standard design forms.
- You cannot necessarily modify objects that originate in standard design forms as replacements for Easy Study Design objects.
- Some standard design-created objects include fields not available in the equivalent Easy Study windows. The system does not make these objects available to the equivalent

Easy Study Design window. You must use standard design forms to access these objects.

Using Easy Study Design

There are five design functions available from Easy Study Design: clinical studies, planned study intervals, clinical planned events, treatment patterns, and patient positions. You can use Easy Study Design to create all these components, some of them, or none of them.

Since the purpose of Easy Study Design is to provide a faster means of creating live studies, the window sets its version to live. In fact, you can only access studies with live versions in the Easy Study Design window.

Data Structure Compatibility

The structure of the data created by Easy Study Design is consistent with the data created by the standard study design modules. This means that studies created in Easy Study Design are accessible in the standard study design forms.

For instance, you can modify the clinical planned events for a study created in Easy Design via the Maintain Clinical Planned Events window. However, the same is not generally true working the other way: Objects created through standard study design forms are not always accessible from Easy Design. Because many fields are not accessible through Easy Study Design windows, if you want to modify studies that have data in these fields, access the data through standard study design forms.

Querying a Study

When you query in Easy Study Design, the system only returns its live version. A study created through the standard clinical studies modules can be queried in Easy Study Design only if it has a live version.

Creating a Clinical Study in the Easy Study Design Window

If you have access to the Maintain Planned Studies window, you can create a new planned study and maintain the planning information.

To create a study, follow these instructions:

1. From the **Design** menu, select **Studies**, then choose **Easy Study Design**.

All fields in this window are mandatory.

2. Enter a value for the study code.

You can use the list function to display all planned studies with no corresponding clinical studies. It is not necessary to pick a study from the list of values. Easy Study Design lets you create both the planned study and the clinical study if you have the necessary database privileges; you must have access to the **Maintain Planned Studies** option from the **Plan** menu. Without this access, you first plan the study.

- If the study code you enter exists in the planned studies table, the system sets the value of the **Study Title** field. In addition, the system populates the

Program Code, **Project Code**, **Organization Code**, and **Region Code** fields with the data from the planned study.

- The system prompts you for a different name if there is already a live version of a study with this name.
 - If no study exists with this name, you are creating a new live version of a clinical study.
3. Enter the version of the study. If a planned study exists and the system populates these fields, go to Step 7.

The system creates a live version when you save your work. Versions facilitate building and comparing variations on one basic study design. They permit altering one component, such as stratification, without forcing you to redesign the entire study.

4. If the **Study Title** field is empty, enter a name of up to 80 characters. The system displays the study title in lists of studies and on DCF reports.
5. Enter a program code for the study. A list of values is available for programs. After you select a program, the description of the program displays to the right of the program name.
6. Enter values for the **Project**, **Organization**, and **Region Code** fields. Lists of values are available, with descriptions.
7. Choose a time unit for events for the study. The default value is DAY. This is the time unit of the study's Intervals.
8. Enter the estimated maximum study duration in the unit of measure specified in the **Time Unit for Events** field.
9. Select the **Available for Replication Flag** box if you expect this study to be distributed to more than one location.
10. Enter a patient replacement rule in the **Patient Repl. Rule** field; a list of values is available. The default is NONE.
11. Save your changes.

When you save your changes, the system prompts designers with the appropriate role to specify the following:

- **Does the study require second-pass data entry?** Select **Yes** if your study will use Oracle Clinical data entry and you want to require that data entry operators enter data twice. You can then reconcile the two passes. Remote Data Capture (RDC) does not support second pass data entry. If you have access to the Maintain Study Statuses menu option, (from the **Design** menu, select **Studies**, then select **Study Statuses**), you can require second-pass entry.
- **Is this a Flexible study?** Select **Yes** if your study has an *adaptive* (or *flexible*) protocol. If you check this box, you can use the Enhanced DCI Books window to define Rules that make specified CRFs and Intervals expected for a particular patient based on data collected for that patient at specified points in the trial.

Select **No** if you do not need this functionality. You must check **No** if you plan to use the Page Tracking feature, which is useful in paper-based studies. You can change this setting and turn on Page Tracking in the Maintain Clinical Study States window under Conduct.



Note:

The Enhanced DCI Books window includes Start Page renumbering, copying, and navigational enhancements that are helpful in any study. Studies using Page Tracking cannot use the Enhanced DCI Books window; they must use the original DCI Books window.

See [Traditional and Enhanced DCI Books](#) and [Designing a Flexible Study](#) for further information.

The Clinical Studies window has the following navigation buttons for completing Easy Study Design tasks. Each task is described in a separate section below.

Table 3-1 Buttons in the Clinical Studies Window

Button	Objective
Intervals	Displays the Maintain Planned Study Intervals window. Intervals are the time units that make up the study timeline.
Events	Displays the Events window, where you specify the number and type of visits.
Treatment	Displays the Treatment Patterns window. Treatment patterns are sequences of one or more treatment regimens.
Create PP	Displays the Create Patients window. Patient positions are the blinded placeholder records that represent the patients in the study.
Delete PP	Displays the Delete Patients window, where you can eliminate unnecessary patient positions.
Display PP	Displays the Existing Patients window, which displays the details of all the patient positions in the study.

For more information, see:

- [Planning Study Intervals](#)
- [Defining Clinical Planned Events](#)
- [Defining Treatment Patterns](#)
- [Creating Patient Positions](#)

Planning Study Intervals

You can define time Intervals to set up a study schedule. See [Maintaining Intervals](#) and [Defining Intervals and CPEs for Flexible Studies](#) for more information.

From Easy Study's Clinical Studies window, click the **Intervals** button. In this window, you can create phases, which are the highest level Intervals. Phases can contain periods, which can contain sub-periods. You have full access to the Interval utility, limited to your current study. You have the same functionality available to define Intervals for your study as described in [Maintaining Intervals](#).

It is most efficient to define Intervals before clinical planned events (CPEs). However, if no Intervals exist for a study when you first invoke the Events window, the system creates two Intervals for you to use in creating your events.

When you create a study using Easy Study Design, the system automatically creates a Phase Interval called DEFAULT STUDY PHASE. If you are creating a nonflexible study and you do not need to organize CPEs into Intervals you can assign all your CPEs to this Interval. If you are creating a flexible study you cannot use this default Interval and must explicitly create Intervals.

For more information, see:

- [Creating Periods](#)
- [Resequencing Periods](#)
- [Creating Sub-periods](#)
- [Resequencing Sub-periods](#)

Creating Periods

Use periods to divide phases into smaller Intervals. From the **Design** menu, select **Studies**, then select **Easy Study Design**, click the **Intervals** button, and then the **Periods** button. (See [Maintaining Periods](#) for more information about periods.)

Resequencing Periods

You can change the order of defined periods within phases from the Easy Study Design window by clicking the **Intervals** button, then the **Periods** button, and the **Resequence** button. (See [Resequencing Phases, Periods and Sub-periods](#) for more information.)

Creating Sub-periods

You can divide periods into sub-periods from the Easy Study Design window. Click the **Intervals** button, then the **Periods** button, and the **Sub-Periods** button. (See [Maintaining Sub-periods](#) for more information.)

Resequencing Sub-periods

You can change the order of defined sub-periods within periods from the Easy Study Design window by clicking the **Intervals** button, then the **Periods** button, the **Sub-Periods** button, and then the **Resequence** button. (See [Resequencing Phases, Periods and Sub-periods](#) for more information.)

Defining Clinical Planned Events

A **clinical planned event** (CPE) usually corresponds to a visit. You can also define a CPE for the purpose of associating it with DCIs (corresponding to CRFs) that you want to have available in a DCI Book without associating them with a particular visit, such as AEs or ConMeds. If you are defining Processes in Oracle Clinical, you can create a CPE to schedule a Process during an Interval.

If you are working in a flexible study, see [Defining Intervals and CPEs for Flexible Studies](#).

For more information, see:

- [Create a Clinical Planned Event](#)

Create a Clinical Planned Event

To create a clinical planned event:

1. From the **Design** menu, select **Studies**, then select **Easy Study Design**, select a Study, then select **Events**. The Clinical Planned Events window opens.
2. Enter values in the following fields:
 - a. Enter an **Event Name** for the clinical planned event. The Event Name must be unique within this study version. You may want to take page numbering into consideration when you name your CPEs, especially in a flexible study; see [Sample Numbering Schemes](#).
 - b. Choose an **Interval Name** from the list of values. Each CPE occurs during an Interval, and you can choose a phase, period, sub-period, or the entire study as the Interval for this event.

 **Note:**

The system prevents you from changing a CPE's Interval if the study is flexible and has an active DCI Book that contains Interval rules. If the DCI Book is provisional, the system sets its validation status to Pending. When you activate the book or run the validation job explicitly, the job determines if the changes are valid.

- c. Enter a **Visit Number**. A CPE's visit number must be unique within this study version. The Visit Number for each CPE controls the order in which they occur in the DCI Book and in the study. CPEs with a lower Visit No. cannot occur after a CPE with a higher Visit No., so be sure to give CPEs the right Visit No. relative to one another. You can add a CPE later, but you must renumber all subsequent CPEs to maintain the correct relative order.

 **Note:**

The system prevents you from changing Visit No. order in flexible studies when the CPE is used in an active Enhanced DCI Book. It is possible to change visit order when the CPE is used in an active traditional DCI Book, but if the change makes the Book invalid you will not be able to use the Book for data entry.

You can always add a CPE to the end of the study. You can add a CPE before the end if you also renumber all subsequent visits.

- d. Indicate if the CPE is optional or required by checking or unchecking the **Optional** checkbox.
- e. If this event describes the termination visit, choose a **Termination Code**. Valid codes are "early" or "normal." You can change this field at any time, provided the study has at most one early and one normal termination visit.
- f. Populate the **Offsets from Interval Start** fields:

- **Minimum** enter the minimum amount of time that must elapse between the start of the Interval and the start of the event, in the time unit specified in the last field.
- **Maximum** enter the maximum amount of time that must elapse between the start of the Interval and the start of the event, in the time unit specified in the last field.
- Time Unit enter the unit that applies to the **Minimum** and **Maximum** fields.

3. Save.



Note:

You cannot access a clinical planned event previously created in the Maintain Clinical Planned Events window from the Easy Study Design window.

Defining Treatment Patterns

Patterns impose a centralized control over the codes that identify treatment patterns in analysis and reports. To add treatment patterns, click the **Treatment** button and choose from the list of values. If you do not require this control, define some dummy pattern codes. (For more information, see [Maintaining Treatment Patterns](#).) The Treatment Patterns window allows you to create, amend, or delete treatment patterns. When you enter the window the system displays all existing treatment patterns for the study.

You can delete a treatment pattern only if no treatment assignments exist for the treatment pattern. To add a treatment pattern, enter the code for the treatment pattern.

Treatment patterns are sequences of one or more treatment regimens. To add a treatment pattern, enter the code of the treatment pattern you want to add. Use the List function to select from the available codes. The system displays both the predefined code and description. Save your changes. Note: You can only delete a treatment pattern if it has no treatment assignments.

Creating Patient Positions

To create a range of patient positions, enter the range of patients for the type(s) of patient positions you want to create. To add patient positions, click the **Create PP** button and choose from the list of values. For example, to create ten screening patient positions, enter the low value for the range in the **Screening Patients Start Code** field, and the high value in the **Screening Patients End Code** field. If you are creating just one patient position, type the same code in both the **Start Code** and **End Code** fields. Click the **Create Patient Positions** button. (See [Maintaining Patient Positions](#) for more information.)

To view existing patient positions, click the **Display Existing Patient Positions** button.

For more information, see:

- [Displaying Patient Positions](#)
- [Deleting Patient Positions](#)

Displaying Patient Positions

You can display patient positions by study or by site.

Viewing by Study

The Existing Patients window shows you all patient positions assigned to a study. All fields in this window are display-only. You can also navigate to this window by clicking the **Display Existing Patient Positions** button, from either the Create Patients or the Delete Patients window.

The system displays all the patient assignments from this window, including any changes you make in the Create or Delete Patient Positions windows. Note: If the system displays a 'Please see details' message in the **Investigator Name** field for a range, the range is assigned to more than one study site. Click the **Display Existing Patient Positions by Site** button to view the patient positions as they are arranged across sites.

Viewing by Site

The **Display Existing Patient Positions by Site** button in the Existing Patients window takes you to the Existing Patients by Sites window. This window sorts a particular range of patient positions by site.

This configuration can show multiple patient position assignments in the event that the same patient positions have been assigned to more than one site.

Deleting Patient Positions

From the Easy Design - Clinical Studies window, you can delete patient positions. Click the **Delete PP** button. Enter the range of patient of the type(s) of patient positions you want to delete. If you are deleting just one patient position, enter the patient's code in both the **Start Code** and **End Code** fields. Click **Delete Patient Positions**.

To see the existing patient positions, click the **Display Existing Patient Positions** button. See [Sites, Investigators, and Patient Positions](#) for more information on patient positions. The fields in the Delete Patients window are identical to standard design's Create Patients window.

Completing Easy Study Design

To make your study operational you can complete these tasks from within the standard design modules:

- Assign the study to one or more study sites. See [Assigning Sites to a Study](#) .
- Assign an Investigator to each of your study sites. See [Assigning Investigators to Study Sites](#) .
- Allocate patient positions to the study sites. See [Assigning Patient Positions to Sites](#).

To refine the definitions of the objects you create in Easy Study Design, you can use the standard study design forms.

4

Sites, Investigators, and Patient Positions

This chapter describes defining sites, Investigators, and patient positions, and assigning them to a study. A study may already have sites assigned at the planning level (see [Maintaining Site Records](#)).

For more information, see:

- [Maintaining Site Records](#)
- [Maintaining Investigator Records](#)
- [Maintaining Patient Positions](#)
- [Treatment Assignments](#)
- [Blind Breaks](#)

Maintaining Site Records

Oracle Clinical refers to the locations where patients receive treatment as **sites**. You create site records, then you assign them to studies. A **study site** is a site that has been assigned to a study. Like Investigator records, site records are not restricted to particular studies.

For more information, see:

- [Creating Site Records](#)
- [Assigning Sites to a Study](#)

Creating Site Records

You can create site records from the Maintain Study Sites window. A study site record is independent of study associations: you can assign a site to any studies. To create a site record, from the **Design** menu, select **Investigators and Sites**, and select **Sites**. You can change site records until you designate a site record as active.

The **Site Code** value can be up to ten characters, and must be unique within a domain. The system determines the owning location. The **Name** field can be up to 60 characters in length. Select the **Active** box when you are ready to make the site record available for assignment to studies.

Assigning Sites to a Study

You can assign sites to a study from the Maintain Study Sites window. From the **Design** menu, select **Investigators and Sites**, then choose **Study Sites**. In this window you can describe in detail how to employ the site in your study. This window is also the starting point for assigning Investigators and patient positions to a study.

 **Note:**

If you are using the Oracle Clinical - Siebel Clinical integration for your study you do not need to define study sites. The definition is populated automatically from Siebel Clinical.

If you have already selected a study during a session, the system takes it by default. If you intend to work on another study, be sure to change studies first. If you have not selected a study during this session, the system first presents a **Select a Study** dialog box. The system then displays all of the site records currently assigned to the selected study.

To add a site assignment to a study, use the list function to select another site in a new record row. The system populates most of the fields. Leave the **Study Site** field blank to force the system to generate the code value for you.

As a convenience, the Study Sites window has buttons that take you to related windows in the system. These extra buttons are:

- **Site Details** Displays the address and owning database. (Defined in the Maintain Sites window. See [Creating Site Records](#) for details.)
- **Assignments** Clicking this button takes you directly to the Maintain Investigators window, where you can assign Investigators to a site by selecting from a list of active Investigators. (See [Assigning Investigators to Study Sites](#) for more information about this window.)
- **Patients** You can link a study's patient positions to a site. Select the **Assign?** box for each existing patient position to assign to the selected study site code. The system populates all other fields.
- **New Patients** Assign patient positions codes to a site by range. Enter the number of patients and the start number. Then click the **Assign Patients** button to assign them to the study site.

Maintaining Investigator Records

As with sites, you can create general records to represent the Investigators who are responsible for collecting data at the sites in any study.

For more information, see:

- [Creating Investigator Records](#)
- [Assigning Investigators to Study Sites](#)

Creating Investigator Records

To create and maintain an Investigator record, work in the Maintain Investigators window. From the **Design** menu, select **Investigators and Sites**, then choose **Investigators**. In the Investigators window you can create, update, or delete Investigator records. See the next section for assigning an Investigator to a study site—a site assigned to a particular study.

 **Note:**

If you are using the Oracle Clinical - Siebel Clinical integration for your study you do not need to define Investigators. The definitions are populated automatically from Siebel Clinical.

Assigning Investigators to Study Sites

 **Note:**

If you are using the Oracle Clinical - Siebel Clinical integration for your study you do not need to assign Investigators to study sites. The assignment is imported automatically from Siebel Clinical.

A study site has exactly one Investigator in charge; however, that assignment can change over time, and a single Investigator might be responsible for several sites. To assign an Investigator to a site:

1. From the **Design** menu, select **Investigators and Sites**, then choose **Study Sites**. The system displays all sites associated with the selected study.
2. Select a site.
3. Click the **Assignments** button. The system displays a history of Investigator assignments to the selected site, if any. Use the list function to select an Investigator record.
4. Enter an Investigator code. You can use the list function to locate a particular Investigator.
5. Enter information regarding the assignment like start and end dates, and the number of patients required.
6. Save. The system designates the new Investigator as current, removing that designation from the Investigator previously in that position.

To view details about a particular Investigator, click the **Investigator Details** button. (See [Creating Investigator Records](#) for more information).

Maintaining Patient Positions

Patient positions are codes that represent the patients enrolled in a study. They are used as the unique ID for patients enrolled in a study.

To see consolidated information about the patient positions in a study, navigate to the **Design** menu, select **Patient Positions**, then **Patients**, and then click the **Patient Positions** button.

This section describes the following Patient Position maintenance tasks:

- [Creating Patient Positions](#)
- [Marking Patients Eligible for Source Data Verification](#)
- [Assigning Patient Positions to Sites](#)
- [Changing Owning Locations](#)

- [Retrieving Patient Positions](#)
- [Deleting Patient Positions](#)
- [Enrolling Patients in a Study](#)
- [Linking Patient Positions in Different Studies](#)
- [Assigning Patient Positions to DCI Books](#)

Creating Patient Positions

You can create patient position code assignments that end in consecutive, system-generated numbers, or you can use your own format.

For more information, see:

- [Creating Patient Positions with System-Generated Codes](#)
- [Creating Patient Positions with User-Defined Codes](#)

Creating Patient Positions with System-Generated Codes

Use the Create and Delete Patient Positions window to create multiple patient positions at once, with codes that are either entirely numerical or that have a single character at the beginning.

To specify the number of patient positions to create:

1. From the **Design** menu, select **Patient Positions**, then select **Create/Delete Patients**.
2. Select the study to modify, if necessary.
3. Click the **Create Positions** button.

The system displays the current Screening, Patients, and Replacement patient positions assigned to the study on the left side of the window.

4. On the right side of the window, make sure the **Location to Own Patients** is correct. For each grouping of patient positions:
 - a. Specify the number of patient positions to create in the **Number to Create** field.
 - b. Specify the code assignment of the first patient position to create in the **Start [Scr|Repl] Patient** field.

You can specify just a number, or you can add an optional single-letter alphabetical prefix such as S for Screening patients and R for Replacements, or the code name of the study or site.

Note:

Do NOT include a slash (/) in the patient code as that will cause the Patient Data Report generation to fail.

The system generates consecutive integers, appended to your prefix, if any. For example:

Current Number	Number to Create	Start Patient	Positions Created
0	5	1	1, 2, 3, 4, 5
0	5	R1	R1, R2, R3, R4, R5

- c. Click the **Create Positions** button.
5. Click the **Back** button.

If you enter a patient code that is already in use, the system returns an error.

If any patient position has a patient code that matches a treatment code on a treatment assignment, the system links them. The system reports how many Screening, Normal, and Replacement patient positions it creates.

To display the existing patient position blocks in a study, from the **Design** menu, select **Studies**, choose **Easy Design**, and click the **Display PP** button. To display individual patient position details, from the **Design** menu, select **Patient Positions**, and choose **Patients (Patient Positions)** button). Oracle Clinical generates the patient position codes in each block, incrementing each successive code by one. The system displays an alert upon completion of the task.

See [Deleting System-Generated Patient Positions](#).

Creating Patient Positions with User-Defined Codes

Use the Alpha Patient Codes window to create patient position codes if the Oracle Clinical-generated codes do not conform to your organization's coding format. You can create codes of mixed characters and numbers with a maximum length of 10. You must create patient positions one at a time.

Note:

The only way to delete patients positions with alpha codes is by selecting one patient at a time in this window and choosing **Delete Record** from the **Data** menu.

1. From the **Design** menu, select **Patient Positions**, then select **Alpha Patient Codes**.
2. Select the study to modify, if necessary.
3. Click the **Patient Positions** button. The system displays all existing patient positions, including site/Investigator assignments.
4. From the **Data** menu, choose **Insert Record**.
5. Enter the new patient position code in the column labeled Patient. The code can have up to ten characters. You must enter the complete patient position code. The system does not append anything.

Note:

Do NOT include a slash (/) in the patient code as that will cause the Patient Data Report generation to fail.

6. Select the **Scr?** box to designate the patient position as a Screening position.

7. Select the **Repl?** box to designate the patient position as a Replacement position.
8. You can assign the position to a site in this window. Use the list function to select a study site.
9. If you assign the patient position to a site, you can use the list function to assign an Investigator from this site.

Repeat Steps 4-9 as often as necessary to enter all the patient positions you need.

10. Save.

See [Deleting Alpha Code Patient Positions](#).

Marking Patients Eligible for Source Data Verification

If you are using a Patient Source Data Verification plan, you must choose a method to make particular patients eligible for 100% source data verification (SDV). Patients are considered to be eligible for SDV when an eligibility date is entered for their patient position. If the SDV plan for a study site specifies an initial number of patients and/or a selection rate, patients are not considered for automatic selection until they become eligible. If you manually select patients for SDV, either in the RDC SDV Plan interface or through a batch job, RDC does not present those patients as requiring verification until they are also eligible.

When you tag a patient as eligible through any of the procedures presented in this section, you must enter an Eligible Date with a non-blank value. If the patient is evaluated and considered ineligible for SDV based on your eligibility criteria, then you must enter a date in the Ineligible Date field.



Note:

If you receive an error message saying that updates are pending for one or more patients in the study and to try again later, you can resolve the problem by running the Execute Pending Patient Updates job under **Conduct, Security, Execute Pending Patient Updates**.

You can make Patient Positions eligible using one of the following methods:

- [Manual Set Up](#)
- [Validation Procedure](#)
- [Batch Load Job](#)
- [Introducing a Patient SDV Plan in an Ongoing Study](#)

Manual Set Up

From **Design**, select **Patient Positions**, then select **Patients**, and then click the **Patient Positions** button. You can enter an eligibility date in the Patient Positions window for one or more patients at the same time.

Validation Procedure

You can make patients eligible for SDV as data is actually entered for them and they meet the criteria you define by creating a validation or derivation procedure for this purpose that populates a column in the PATIENT_POSITIONS table:

- Populate the SDV_ELIGIBLE_DATE column to make the patient eligible for automatic selection for SDV on the date entered.
- Populate SDV_INELIGIBLE_DATE column to prevent the patient from being selected for SDV.

For example:

- **Eligible:** If the Demography CRF is completed only after the Screening CRF is completed satisfactorily, you could write a procedure to make the patient eligible when data is entered and saved for the Demography CRF.
- **Ineligible:** If the responses in the Screening CRF are unsatisfactory, make the patient ineligible. If you have an existing screening procedure, you may wish to simply update that procedure. (RDC reports, but does not otherwise use this value.)

Finally, you can have your procedure populate the SDV_PROC_VERS column to indicate which version of your procedure derived the patient eligibility or ineligibility.

The eligibility procedure must be written to ensure that the SDV Eligible Date and SDV Ineligible Date fields do not both have not-null values at the same time. If one date is entered, the other must be null.

Using the OCL_UTILS Utility

Call the OCL_UTILS.UPDATE_SDV_PAT procedure from the utility to populate the SDV Eligibility or SDV Ineligibility fields dynamically. The OCL_UTILS procedure is compiled in the RXC user schema and located in the server code (\$RXC_INSTALL).

1. Write a procedure that will be triggered by data you choose. See **Procedure Example** below.
2. Define your procedure to call the OCL_UTILS.UPDATE_SDV_PAT procedure, providing the input parameters as shown in [Table 4-1](#). Pass a single date to the procedure. Whether to treat the date as an Eligibility date or an Ineligibility date depends on the flag setting.

Table 4-1 Parameters in OCL_UTILS.UPDATE_SDV_PAT procedure

Parameter	Description	Example Values
Mode	T - Test mode P - Production Mode	rxcpdst.v_mode
patient_position_id	Position ID for the patient that should be updated	rxcpdst.patients_rec.
Proc Version	Version of the procedure	rxcpdst.v_procedure_version_sn
SDV Eligible Date	Date of SDV Eligibility OR Ineligibility	date

Table 4-1 (Cont.) Parameters in OCL_UTILS.UPDATE_SDV_PAT procedure

Parameter	Description	Example Values
SDV Eligibility Flag	If Y, treat the date as an Eligibility date. If N, treat the date as an Ineligibility date.	Y or N

Procedure Example

For example, using expression as SEX = F and A as alias for a DCM Question Group, you can add the following lines to your procedure to update the SDV Eligibility Date, SDV Ineligibility Date, and the Procedure Version number for a patient:

```

If A.SEX = 'F'
THEN
ocl_utils.update_sdv_pat(rxcpdstd.v_mode,rxcpdstd.patients_rec.patient_positio
n_id, rxcpdstd.V_PROCEDURE_VERSION_SN,to_date(DE.DCM_DATE,'YYYYMMDD'),'Y');
ELSE
ocl_utils.update_sdv_pat(rxcpdstd.v_mode,rxcpdstd.patients_rec.patient_positio
n_id, rxcpdstd.V_PROCEDURE_VERSION_SN,to_date(DE.DCM_DATE,'YYYYMMDD'),'N');
END IF;
.
COMMIT;

```

See [Validation and Derivation Procedures](#).

Batch Load Job

You can use this option to submit a Patient SDV Import. Select **Design**, then **Patient Positions** and then **Load Patient SDV Attributes**. Enter the study for which to run the batch job. The parameters required are:

- Name of the Patient SDV File.
- Allow batch upload to over-write eligibility values? Choose Y for the values specified in the input file to overwrite current values in the Patient Positions table.

Patient SDV File

You can load a Patient SDV File to accomplish either or both of the following functions, depending upon the parameters you elect to use for each patient code:

- Make patients eligible or ineligible for SDV
- Select patients for 100% SDV

Make sure the data in the file you use matches the study for which you import it, otherwise the job will fail. For more on the location of the file see "Setting Up File and Image Viewing" in the *Oracle Clinical Administrator's Guide*.

The input file content must be comma delimited and any comment inserted must be preceded by the hash key (#). The file must have the following fields for each record:

- **Study** (mandatory)
- **Patient Code** (mandatory)
- **SDV Eligible Date** (optional) in the DD-MON-YYYY format

- **SDV Ineligible Date** (optional) in the DD-MON-YYYY format
- **Procedure Version** (optional). This field would ordinarily be used if you are using a validation procedure to set SDV eligibility. It allows you to record the version responsible for making a patient SDV eligible or ineligible.

However, you may want to establish a convention for values to be placed in this field when Eligible or Ineligible dates are either manually updated or imported. For example, '0' could be used to indicate that the date was not specified by the eligibility procedure

- **SDV Requirement** (optional). There are two valid values: SDVSelect and SDVExclude:

Processing

The Import processes each patient individually, as follows:

- If values are specified for one or more of the 3 eligibility parameters (and the patient doesn't already have a value for the corresponding parameter, or you've indicated existing values should be over-written), the patient positions table is updated.
- If a value is specified for SDV Requirement, an update is made to a new or existing draft Patient SDV plan, as shown in the table below:

Table 4-2 Values for the SDV Requirement parameter and results in the Draft and Production Plans

Import file value for Parameter SDV Requirement	Pending Action in Draft Plan	Selection Status when the Plan Becomes Production
SDVSelect	Import	Imported
SDVExclude	Import Exclusion	Import Excluded

- An action may not be taken for a patient position in some circumstances as listed below. In these cases, a warning message is issued to the batch job's log file.
 - The study is frozen
 - Allow batch upload to over-write eligibility values? = n, and the patient already has a value for an eligibility parameter specified in the file
 - Both an eligible and an ineligible date are specified for a patient
 - A patient is already selected for SDV in the existing draft or published SDV plan
 - The specified patient has not been assigned to a site

Introducing a Patient SDV Plan in an Ongoing Study

Following are two options for introducing a patient SDV plan in an ongoing study.

- **Create an eligibility procedure for an existing DCI**
This will calculate eligibility for both existing and new patients.
- **Use a different approach for current vs. new patients**
For new patients, create a new DCI with a new eligibility procedure.
For existing patients, manually specify or import the SDV eligibility status.

Once you have implemented your eligibility methodology, there are two other steps to be completed before SDV patient auto-selection can begin for a study-site:

- "Schedule the Pending Patient Updates job" in the Oracle Clinical Administrator's Guide

- Publish a patient SDV plan for the study site

For information on how patient auto-selection works, refer to "Auto-selecting Patients for SDV" in *Oracle Clinical Remote Data Capture Onsite User's Guide*.

Also, see more about "Publishing SDV plans for all Sites" in Oracle Clinical Conducting a Study.

Assigning Patient Positions to Sites

This section describes several approaches to assigning patient positions to sites. Since only one Investigator can be in charge of a site for a particular study, by assigning a patient position to a study site, you also assign it to an Investigator. You can make this assignment from two different environments in Oracle Clinical.

For more information, see:

- [Assigning Patient Positions from the Study Sites Window](#)
- [Assigning Patient Positions from the Patient Positions Window](#)
- [Assigning Patient Positions by Range](#)
- [Assigning New Blocks of Patient Positions](#)

Assigning Patient Positions from the Study Sites Window

To assign patient positions to sites and Investigators:

1. From the **Design** menu, select **Investigators and Sites**, then choose **Study Sites**.
2. Select the study, if necessary.
3. Click the **Patients** button. The system displays the current patient positions and their site assignments.
4. Assign the patient positions to the site by selecting the **Assign?** box.
5. Save.

Assigning Patient Positions from the Patient Positions Window

1. From the **Design** menu, select **Patient Positions**, then select **Study Site Relations**.
2. Select the study, if necessary.
3. Click the **Patients** button. The system displays the existing patient positions available to the site.
4. Assign the patient positions to the site by selecting the **Assign?** box.
5. Save.

Assigning Patient Positions by Range

You can also assign a range of positions to a study site. To select a range of patient positions or screening patient positions to link to the selected study site:

1. From the **Design** menu, select **Patient Positions**, then choose **Study Site Relations**.

2. Click the **Sites** button.
3. Select the study site.
4. Click the **Patient Ranges** button. The left side of the window displays the current state of the system and refreshes each time you make a change.
5. Define the treatment range on the right side of the window.
6. Click the **Assign Patients** button. The system removes the instructions as it acts on them.

Assigning New Blocks of Patient Positions

From the **Design** menu, select **Investigators and Sites**, then select **Study Sites**, and click the **New Patients** button. Enter the number of patients and the start number. Then click the **Assign Patients** button to assign them to the study site.

Changing Owning Locations

If you are replicating data among many locations, you must create all patient positions required for a study at the study-owning location. However, users at other locations can use only patient positions that are owned by their location. You must therefore change the ownership of the patient positions required by all other locations.

After you assign a patient position to a non-study-owning location, you cannot change its owning location again unless you first reassign it to the study-owning location; see [Retrieving Patient Positions](#).

To change a specified number of patient positions from the current location to another:

1. From the **Design** menu, select **Patient Positions**, then select **Create/Delete Patients**.
2. Select the study to modify, if necessary.
3. Click the **Change Owning Location** button.

The system displays the current Screening, Patients, and Replacement patient positions assigned to the study on the left side of the window. On the right side of the window, for each grouping of patient positions, enter:

- a. the number of patient positions to change in the **Number to Update** field.
 - b. the starting assignment code of the first patient position in the **Start** field.
4. Click the **Change Owning Location** button.

Retrieving Patient Positions

If you are replicating data among many locations, the study-owning location controls the assignments of all patient positions. If a patient position owned by a different location must change ownership, the study-owning location must reclaim, or retrieve, ownership of the patient position first. This utility is only available at the study-owning site.

To retrieve patient positions, from the **Design** menu, select **Patient Positions**, then select **Retrieve Patients**.

1. Use the list function to select the site that currently owns the patient position. The system populates the **Source Study Title** field.

2. Use the list function to select the study that owns the patient position. If there is only one study at that site, the system populates the field for you.
3. Click the **Retrieve Patients** button.
4. Save.

Deleting Patient Positions

You can delete a patient position only if its patient enrolled date is not set, and no received DCIs exist for the patient. If you delete a patient position, the system breaks any links to treatment assignments or study sites.

To view existing patient positions, from the **Design** menu, select **Patient Positions**, then choose **Patients**.

You can delete multiple patient positions, including those with an alphabetical prefix, at the same time if:

- they were generated by the system (not created individually in the Alpha Patient Codes window)
- they are sequential

If the patient positions have alpha codes, you must delete them one by one in the Alpha Patient Codes window (see [Deleting Alpha Code Patient Positions](#)).

For more information, see:

- [Deleting System-Generated Patient Positions](#)
- [Deleting Alpha Code Patient Positions](#)

Deleting System-Generated Patient Positions

To delete system-generated patient positions:

1. From the **Design** menu, select **Patient Positions**, then select **Create/Delete Patients**.
2. Select the study to modify, if necessary.
3. Click the **Delete Positions** button.

The system displays the current Screening, Normal, and Replacement patient positions assigned to the study on the left side of the window.

On the right side of the window, for each grouping of patient positions, specify:

- a. the number of patient positions to delete in the **Number to Delete** field.
 - b. the code of the first patient position to delete in the **Number to Delete Start** field.
4. Click the **Delete Positions** button.
 5. If necessary, repeat this process for patient positions not included in these sequences.

Deleting Alpha Code Patient Positions

To delete patient positions with user-specified alpha codes (see [Creating Patient Positions with User-Defined Codes](#)), you must go to the Alpha Patient Codes window and delete them one at a time:

1. From the **Design** menu, select **Patient Positions**, and choose **Alpha Patient Codes [Patient Positions]**.
2. Select the patient position you want to delete.
3. From the **Data** menu, choose **Delete Record**.
Repeat Steps 3 and 4 as many times as necessary to delete all the patient positions you need to delete.
4. Save.

Enrolling Patients in a Study

Patient positions are codes that represent the people you enroll in a study. They are used as the unique IDs. You enroll a patient in a study by entering any of the following data against a patient position:

- Patient enrollment date
- Patient's last name
- Patient's initials
- Patient's data of birth

 **Note:**

Although Patient Initials and Date of Birth are optional in Oracle Clinical, you may want to make a policy to always enter them because they are required by Oracle Clinical Labs functionality and by Siebel Clinical; see [Lab Ranges](#) and [Integrating with a Clinical Trial Management System](#).

You can enroll patients in several ways, including:

- During study design in the Maintain Patient Positions window. See [Adding Patient Position Details](#) for more information.
- As a batch job. See [Batch Load Patient Enrollment Dates](#) for more information.
- During initial login and data entry in Oracle Clinical. See *Oracle Clinical Conducting a Study* for more information.
- During data entry in Oracle Clinical Remote Data Capture. See the *Oracle Clinical Remote Data Capture User's Guide* and *Oracle Clinical Remote Data Capture Classic Data Entry User's Guide* for more information.
- Using a derivation procedure that reads the response value of a Question that collects the enrollment date and populates the value of the PATIENT_ENROLLMENT_DATE column in the PATIENT_POSITIONS.

You can view enrollment information in **Labs**, **Lab Assignment Criteria**, **View Patient Positions**. See [Missing Data in Patient Positions](#) for more information.

The system requires that you update the `PATIENT_STATUS_CODE` column of the `PATIENT_STATUSES` table to determine when patients are enrolled in a study for the purpose of the enrollment form and reports. You can define additional patient statuses and update them as appropriate for each user. See [Maintaining Patient Statuses](#) for more information.

In this section, see:

- [Adding Patient Position Details](#)
- [Batch Load Patient Enrollment Dates](#)

Adding Patient Position Details

To enroll patients in a study you must associate them with a patient position. You can do this in the Maintain Patient Positions window. To access the Maintain Patient Positions window, from the **Design** menu, select **Patient Positions**, then **Patients**, and then click **Patient Positions**. If necessary, select a study from the Special menu.

The Maintain Patient Positions window displays the patient positions that are owned by your location. The following fields are display-only:

- **Patient:** A unique code for an existing patient position.
- **Scr'ing?:** (Screening) If selected, the patient position was created for a Screening patient.
- **Repl-ment?:** (Replacement) If selected, the patient position was created for a Replacement patient.

You can enter values for a particular patient in the following fields:

- **Enrollment Date:** Date (DD-MON-YYYY) when the patient enrolled in the study. This is optional except when the Patient Position is associated with a Lab Assignment Criteria. You cannot change this field if the **Frozen?** is checked. You can specify search criteria in this field. You can use a derivation procedure—based on the response to a DCM Question—to complete this field.
- **Birth Date:** This is optional except when the Patient Position is associated with a Lab Assignment Criteria. You cannot change this field if **Frozen?** is checked. For query tasks, you can specify search criteria in this field.
- **Sex:** The patient's gender. This is optional. Acceptable values are **M** (Male) and **F** (Female). The system may use the value in this field to determine the appropriate normal range for a patient's laboratory responses. You cannot change this field if **Frozen?** is checked. You can specify search criteria in this field and can use a derivation procedure—based on the response to a DCM Question—to complete this field automatically.
- **Inits:** The patient's initials. This is optional. It accepts up to 4 characters of mixed case. You cannot change this field if **Frozen?** is checked. For query tasks, you can specify search criteria in this field.
- **Frozen?:** If checked, the system prevents any changes to the data for this patient. This is a display-only field.
- **Efficacy:** If checked, this patient's data is included in an efficacy analysis.
- **Safety:** If checked, this patient's data is included in a safety analysis.

- **I/E Date:** The date (DD-MON-YYYY) when the decision was made to include or exclude this patient's data in efficacy and/or safety analyses. You can edit this field only if the patient's **Frozen?** field is unchecked.
- **Term Date:** The date (DD-MON-YYYY) the patient was terminated from this clinical study. This is an optional field. You can use a derivation procedure, based on the response to a DCM Question, to complete this field automatically.
- **Reported Reference:** An optional, additional patient identifier (up to 25 characters), which can be displayed in RDC Onsite. See "Adding a Customized Reference Field to the Search Pane." in *RDC Onsite Administrator's Guide*. You can edit this field only if the patient's **Frozen?** field is unchecked.
- **First Name:** The patient's given name. This field is optional. It can contain up to 15 characters of mixed case. You can edit this field only if the patient's **Frozen?** field is unchecked. You can specify search criteria in this field.
- **Last Name:** The patient's family name. This field is optional. It can contain up to 15 characters of mixed case. You can edit this field only if the patient's **Frozen?** field is unchecked. You can specify search criteria in this field.
- **Death Date:** The date (DD-MON-YYYY) the patient died.
- **Owning Location:** The code of the database location that owns this patient position.
- **DCI Book Name:** The DCI Book assigned to this patient position. The list of values includes active DCI Books associated with the study.

Batch Load Patient Enrollment Dates

If you have an ASCII data file of patients' enrollment dates containing patient codes and dates, either from a phone enrollment or external randomization system, you can use the Load Patient Enrollment Dates PSUB submission window to load them into the system.

Load the file into the directory specified in the BDL_ROOT setting of the OCL_STATE local reference codelist. If BDL_USERDIRS in the same codelist is set to Y, load the file into the user-specific subdirectory for the user who will run the job. See the *Oracle Clinical Administrator's Guide* for more information.

Use the following format for batch loading patient enrollment dates:

- Study Code – 15 characters
- Patient Code – 10 characters
- Enrollment Date – YYYYMMDD
- Sex (optional) – M or F
- Birth Date – YYYYMMDD (optional)

To run the batch load, from the **Design** menu, select **Patient Positions**, then select **Load Enrollment Dates**. The system sets the enrollment date on the patient position. The system ignores a date if it matches the date from the file. If the dates are different, the system records the date and generates an error in the log file.

 **Note:**

Do not leave an empty line in the load file. The system treats each line in the load file as a record. If there is an empty line, the system takes that as an empty record and displays an error.

Linking Patient Positions in Different Studies

You can define patient positions in different studies to be the same person. The Link Patient Position utility enables you to link the records for the two patient position codes together. You can link any number of patients together, but if you attempt to link patient positions with different dates of birth or sex, the system prompts you to verify your request. Two situations where you may need to do this are:

- Tracking a patient's participation in Phase 1 or volunteer studies
- Rollover studies where patients are in two successive studies

From the **Design** menu, select **Patient Positions**, and select **Link Patient Positions**. You can indicate that the data collected from more than one patient position code originates from the same patient. Your company must write an extract macro, incorporating the primary key `CLINICAL_SUBJECT_ID`, to process the link you make here. For instance, you can combine screening data recorded against a screening patient position code with randomized data recorded against a regular patient position code, or link data from a short term and rollover study. Choose a study, then click the **Patient Positions** button. Select the patient to link to another patient position.

To link patient positions:

1. Following the path given earlier, select the study, if it differs from the study of the current session.
2. Click the **Patient Positions** button.
3. Select the patient position to link to another patient position.
4. Click the **Link to Other Patients** button.
5. Enter the study name that contains the target patient position. Use the list function to view a list of available studies.
6. Enter the patient position code for the target patient position.
7. Save.
8. Repeat this procedure for each link.

Since your study may require that you not divulge information about patients that could identify them, the system limits you to indicating that the same patient fills two or more patient positions.

Assigning Patient Positions to DCI Books

Oracle Clinical 4.6 adds the option to assign or reassign groups of Patient Positions by range, site, or study.

Navigate to Design, Patient Positions, Patients. Select a study and then click the Patient Positions button to open the Maintain Patient Positions window. Click the **Assign DCI Book** button to open the Maintain DCI Book Assignments window.

For example, if an active study definition changes because of a protocol amendment, groups of patient positions may require changes to a unique set or sequence of data collection. In previous releases of Oracle Clinical, you assigned patient positions individually, which could be difficult for large studies. Oracle Clinical 4.6 allows you to assign groups of patient positions in one operation.

For more information, see:

- [Assigning Patient Positions by Range](#)
- [Assigning Patient Positions by Site](#)
- [Assigning Patient Positions by Study](#)

Assigning Patient Positions by Range

To assign a sequential range of patient positions:

1. Choose Patient Range in the Patient Selection Panel.
2. Enter a Patient Position or choose one in the **from** list of values.
3. Enter a Patient Position or choose one in the **to** list of values.
4. Enter, or choose All, Unassigned, or a valid Book name from the Patients Assigned To Book list of values.
5. Enter or choose a Target Book from the **Target Book** list of values.
6. Click **Assign Book**.

Assigning Patient Positions by Site

To assign all of the patient positions from one site:

1. Click **All Patients at Site** button.
2. Enter a site or choose one in the **Select Site** list of values.
3. Enter, or choose All, Unassigned, or a valid Book name from the Patients Assigned To Book list of values.
4. Enter or choose a Target Book from the **Target Book** list of values.
5. Click **Assign Book**.

Assigning Patient Positions by Study

To assign whole study populations to a new casebook:

1. Click **All Patients for this Study** button.
2. Enter, or choose All, Unassigned, or a valid Book name from the Patients Assigned To Book list of values.
3. Enter or choose a Target Book from the **Target Book** list of values.
4. Click **Assign Book**.

Treatment Assignments

In this section:

- [Assigning Selected Treatment Assignments to a Site](#)
- [Linking Treatment Assignments](#)

Assigning Selected Treatment Assignments to a Site

From the **Design** menu, select **Patient Positions**, choose **Study Site Relations**, click the **Sites** button, and click the **Treatment Assignments** button.

You can assign selected treatment assignments from all available treatment assignments for this study from this menu path. Use the Treatment Ranges utility, below, to assign sequences of treatment assignments that are coded consecutively, or the **Randomizations** button to narrow the selection of treatment assignments to a particular randomization.

To assign selected treatment assignments:

1. Select a randomization to assign randomized treatment assignments.
2. Click the **Treatment Assignments** button. You have reached this stage by selecting **Patient Positions** from the **Design** menu, then selecting **Study Site Relations**, and clicking the **Sites** button. You can also reach this display by selecting **Patient Positions** from the **Design** menu, then choosing **Study Site Relations**, clicking the **Sites** button, and then the **Randomizations** button; this narrows the selection to treatment assignments from a particular randomization. The remaining instructions are the same for both menu paths.
3. Select the **Assign?** box for each treatment assignment. The system populates the **Site** field with the current site selection.
4. Save.

Linking Treatment Assignments

Typically, you link patient positions to study sites, and their treatment assignments with them. You must link treatment assignments to a study site only when they do not match patient positions.

To assign or remove individual treatment assignments from a study site:

1. From the **Design** menu, select **Patient Positions**, then choose **Study Site Relations**.
2. Select the live clinical study version of interest. If your currently selected study does not have a live version, the system prompts to see if you want to view all studies. If your study has a live version, but you do not have it currently selected, the system prompts to see if you want to switch to the live version.
3. Click the **Sites** button. The system displays the Maintain Study Site Relationships window, listing all the study's sites.
4. Select a site. You can only select a study site while in this window; you cannot insert, update, or delete a study site here. (To assign sites to studies in the

Investigator subsystem, from the **Design** menu, select **Investigators and Sites**, then choose **Study Sites**.)

5. Choose between viewing all available treatment assignments or those of a randomization:
either: Click the **Treatment Assignments** button. The system displays all treatment assignments available for the study and site.
or: Click the **Randomizations** button. The system displays the randomizations assigned to the study.
 - a. Select a randomization.
 - b. Click the **Treatment Assignment** button. The system displays all treatment assignments assigned to the selected randomization.
6. Select the **Assign?** box for each treatment assignment you want to link to your study site. Deselect the **Assign?** box for each treatment assignment you do not want linked to your study site.
7. Save.

For more information, see:

- [Linking Treatment Assignments by Range](#)
- [Linking Treatment Assignment Ranges to a Site](#)

Linking Treatment Assignments by Range

Typically, you can assign patient positions and their treatment assignments to study sites. You have to assign treatment assignments to a study site only when they do not match patients automatically. To select treatment assignments to link to the selected study site, work in the Link Treatment Assignments by Range window:

1. From the **Design** menu, select **Patient Positions**, then select **Study Site Relations**.
2. Click the **Sites** button.
3. Select the study site.
4. Click the **Treatment Ranges** button. The left side of the window displays the current state of the system and refreshes each time a you make a change.
5. Define the treatment range on the right side of the window.
6. Click the **Assign Treatments** button. The system removes the instructions as it acts on them.

Linking Treatment Assignment Ranges to a Site

From the **Design** menu, select **Patient Positions**, choose **Study Site Relations**, click the **Sites** button, and then the **Treatment Ranges** button.

You can assign ranges of consecutively coded treatment assignments to sites. Following the path given earlier, select study and site. The window displays the site's current treatment assignments on the left side of the window. The system assigns the individual treatment assignment codes according to the number and start code number you specify on the right side of the window.

Blind Breaks

Circumstances sometimes require going against data blinding controls to access information on treatment patterns for a particular patient.

If an Investigator opens an occluded label for a patient, you record the blind break this represents in this window. But you cannot disclose the treatment pattern assigned to a patient in this window.

Oracle Clinical provides an emergency mechanism for obtaining a patient's treatment details. If you use this mechanism, the system generates a blind break record. The system stores blind breaks you document and system-generated blind breaks together.

For more information, see:

- [Creating, Reviewing, or Canceling Blind Breaks](#)

Creating, Reviewing, or Canceling Blind Breaks

Associated with the Blind Breaks form are three windows, each accessed from a corresponding navigation button.

Note that you cannot delete records of blind breaks. You can cancel them, but the system retains a record of canceled breaks.

To create or review a blind break:

1. From the **Design** menu, select **Patient Positions**, then choose **Blind Breaks**.
2. Select the clinical study associated with the patient positions you want. If you have already accessed a clinical study during this Oracle Clinical session, the system presents it; otherwise this window lists all clinical studies.
3. Click the **Patients** button. The system displays the Patient Positions window.

In the Patient Positions window, you can choose to do the following tasks. Save your work when you finish.

- Viewing an existing blind break
 1. Select the patient position code with the blind break. The patient positions appear in patient code sequence.
 2. Click the **Show Blind Breaks** button.

If there are no blind breaks for the selected patient position, this window does not appear; instead, a message states that no blind breaks exist. You can only select a blind break while in this window. You cannot insert, amend, or delete blind breaks here. The blind breaks appear in order of their creation.

- Canceling an existing blind break:
 1. Select the patient position code with the blind break. The patient positions appear in patient code sequence.
 2. Click the **Show Blind Breaks** button.
 3. Select the blind break to modify.
 4. Click the **Modify Blind Breaks** button.

5. Enter a description of why you are canceling the blind break.
- Creating a new blind break:
 1. Click the **Create Blind Break** button.
 2. Enter the information about the blind break.

5

Treatments and Schedules

This section first describes creating and modifying treatments, which are definitions of drugs and how to administer them. You compose a drug definition by defining its active substance and dose form. By defining the details about a drug's dosing, you create a single regimen. By assigning a single regimen, regimens, or combinations of single regimens (combination regimens), to Intervals in the study timeline, you create a treatment pattern. The latter part of this section describes creating the Intervals for a study's timeline and creating a treatment pattern.

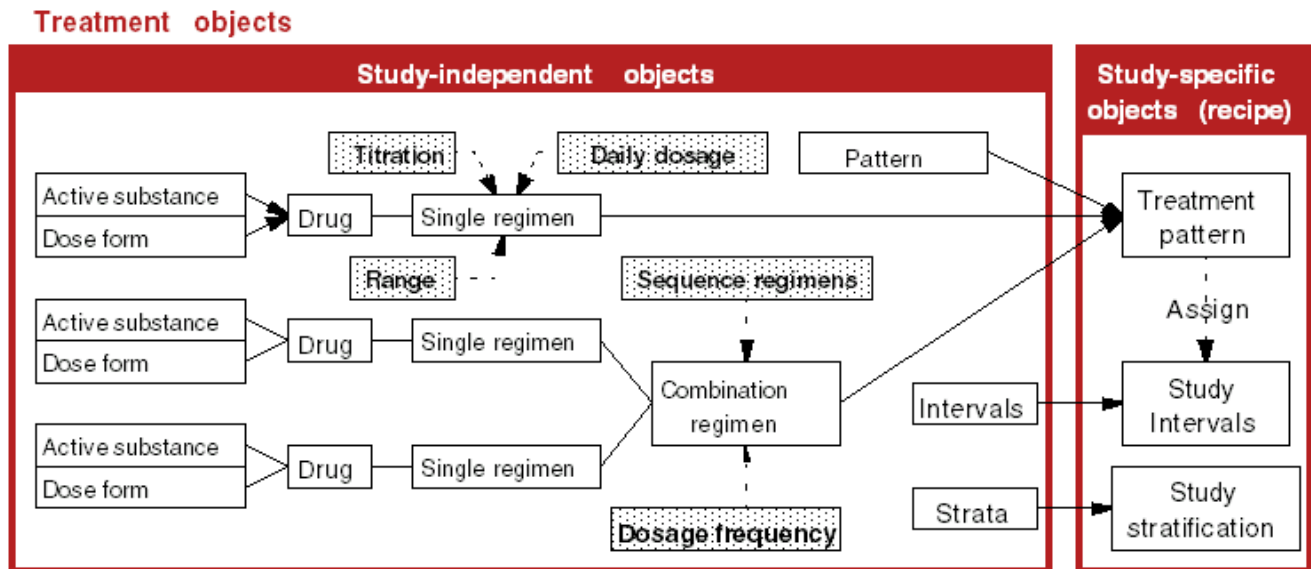
In this section:

- [Treatments](#)
- [Defining a Study Schedule](#)
- [Creating Clinical Planned Events](#)

Treatments

The Treatment Objects diagram (Figure 5-1) illustrates some of the possible relationships you can create with Oracle Clinical. The gray-filled boxes represent optional specifications you can add to the definitions.

Figure 5-1 Treatment Objects Diagram



Treatments represent the medications a patient should take and the instructions for their administration. The term **single treatment regimen** describes taking one medication at a time. For situations where patients take more than one medication at a time, you collect single treatment regimens into **combined treatment regimens**.

For more information, see:

- [Maintaining Active Substances](#)
- [Maintaining Drugs](#)
- [Maintaining Single Treatment Regimens](#)
- [Creating or Modifying a Single Treatment Regimen](#)
- [Maintaining Daily Dosage Details for a Treatment Regimen](#)
- [Maintaining Combined Treatment Regimens](#)
- [Maintaining Patterns](#)
- [Maintaining Treatment Patterns](#)
- [Maintaining Treatment Pattern Codes](#)

Maintaining Active Substances

An **active substance** is a chemical compound that is part of a medication. Active substances are study-independent objects and once defined, they become available for assignment in planned objects. The Maintain Active Substances window enables you to define the formulation of your active substances and record drugs you are using from other sources, and these active substances to programs. You can also delete active substances in this window, but only if they are not assigned to any treatment regimen.

To define a new active substance:

1. Navigate to **Design, Treatments**, then **Active Substances**. The Maintain Active Substances window opens in multi-view format.
2. Add a record, then supply values for the following fields:
 - **Name** – A unique name, of up to 70 characters.
 - **Active?** – Select this box to make this active substance available for defining new drugs.
 - **Code** – The identifier of the active substance, with up to 10 characters.
 - **Sub Code** – An optional identifier extension of up to 10 characters. You can use subcodes to categorize related active substances under the same code.
3. Save. Oracle Clinical populates the **Substance Id** field with a unique ID number for this active substance.

Maintaining Drugs

In Oracle Clinical, the term **drug** refers to a medication defined by an active substance and a dose form or delivery method. A drug definition consists of an active substance and its delivery method, or dose form. The term **product master** is an industry synonym. Each distinct active substance/dose form combination must have a unique name. You may need to create drug definitions for one or all of the placebos in your inventory, depending on your packaging and labeling standards.

To define a drug:

1. From the **Design** menu, select **Treatments**, then select **Drugs**. The Maintain Drugs window opens.

2. Add a record, then supply the following fields:
 - **Drug Name** – Unique name, of up to 70 characters.
 - **Active?** – Select this field if the drug is available.
 - **Code** and **Sub Code** – The active substance's identifiers. The system populates these fields when you choose an active substance.
 - **Dose Form** – Choose the dosage medium from the list of values. The values are set in a reference codelist.
 - **Strength** – The amount of active substance in the drug.
 - **UOM** – The Unit of Measurement in which the drug's strength is measured.
 - **Based on Active Substance** – Choose an active substance from the list of values.
3. Save. Oracle Clinical populates the **Drug Id** field with a unique ID number.

Maintaining Single Treatment Regimens

The treatment regimens you create or update are for the use of all clinical studies. You can create and update single treatments, assign dosage by body factors, define titration models, and set minimum and maximum daily dosages.

Creating or Modifying a Single Treatment Regimen

A **single treatment** is a regimen that describes how to take a single drug in one particular way. A **single treatment regimen** is a specific prescription for taking a single drug. This is somewhat like a prescription for taking a drug, except that it does not specify duration. A **combined treatment** describes how to take one or more drugs in one or more ways.

To create or modify a single treatment regimen:

1. From the **Design** menu, select **Treatments**, then choose **Single Regimen**. The Maintain Single Treatments window opens.
2. Select a drug to assign to a treatment regimen.
3. Click the **Treatments** button.
4. Define the details of the single treatment regimen. You can only define simple linear titration models here. Select a single treatment regimen to make the following modifications:
 - Click the **Single** button to view all the fields for a regimen on one window.
 - Click the **By Range** button to assign dosage by body factors, such as weight. (See [Specifying Dosage by Range](#).)
 - Click the **Titration** button to define complex titration models. You can define titration dosage steps for a nonlinear progression. The system displays the titration steps in dose sequence order. (See [Maintaining Non-linear Titration Steps for a Treatment Regimen](#).)
 - Click the **Daily Doses** button to define the minimum and maximum daily dose in the treatment regimen. (See [Maintaining Daily Dosage Details for a Treatment Regimen](#).)
5. Save.

For more information, see:

- [Specifying Dosage by Range](#)

- [Maintaining Non-linear Titration Steps for a Treatment Regimen](#)

Specifying Dosage by Range

The Dosages by Range subwindow enables you to define dosages according to linearly progressing patient characteristics. For example, you can specify dosage ranges according to age, where you specify the dosage to be 100 Mg for someone up to three years old, 150 Mg for a patient between three-plus years up to five years old, and 200 Mg for a patient between five-plus years up to seven. The system sorts any existing range definitions first by sex, then by maximum value.

To set a regimen's dosage by range:

1. With the Maintain Single Regimens window open, choose a regimen and click the **By Range** button. The Dosages by Range for *Regimen Name* window opens.

Navigate to **Design, Treatments, Single Regimen**, then click the **Treatments** button, and the **By Range** button. Use this window to define dosages according to linearly progressing patient characteristics.

2. Choose a **Range Factor**, which provides a unique code for the criteria that controls the dosing. A list of values is available, and the available values are populated by navigating to **Design, Strata**, and then **Factors**.
3. Enter a row in the table to specify each range. These rows includes the following fields:
 - **Sex** – Dosing ranging requires you to specify the patient's sex. Choose M for Male or F for Female.
 - **Maximum Value** – A number, of up to seven digits, that is unique in this regimen. The system interprets a range to represent the next largest value above any smaller maximum value to this one.
 - **Dosage** – Enter the dosage for this maximum value. It must be a number, of up to eight digits.
4. Save.
5. Click the **Back** button to return to the Maintain Single Regimens window.

Maintaining Non-linear Titration Steps for a Treatment Regimen

The Titration window enables you to define dosages for non-linear characteristics.

To define each titration step for a regimen:

1. With the Maintain Single Regimens window open, choose a regimen and click the **Titration** button. The Non-linear Titration Steps for Regimen *Regimen Name* window opens, displaying all existing titration steps for the selected treatment pattern in dose sequence order.
2. For each titration step, populate a new row in this window. Each row includes the following fields:
 - **Dose Sequence** – Enter a number of up to ten digits, where zero is the starting dose, each negative number is an incrementally lower dose, and a positive number is an incrementally higher dose.
 - **Dose Quantity** – The amount of the drug to be administered to the patient, up to 8 numeric characters, to be administered to the patient.

- **Time Offset** – The number of default time units to wait since the last administration.
3. Save.
 4. Click the **Back** button to return to the Maintain Single Regimens window.

Maintaining Daily Dosage Details for a Treatment Regimen

You can use the Daily Dosage Details window to define the minimum and maximum allowable quantities of a drug for one 24-hour period.

To define the daily dosage details for a treatment regimen:

1. With the Maintain Single Regimens window open, choose a regimen and click the **Daily Doses** button. The Daily Doses for Regimen *Regimen Name* window opens, displaying all of the daily dose records for this treatment regimen.
2. Add a row. The window automatically populates the **Substance Id** field with the active substance's ID number, and if they exist, also populates the **Substance Code** and **Sub Code** fields. None of these fields are editable.
3. In the **Min Daily Dose** field, enter the minimum daily dose of the drug. This entry can be a number of up to eight digits.
4. (Optional) In the **Max Daily Dose** field, enter the maximum daily dose of the drug. This entry can be a number of up to eight digits.
5. Save.
6. Click the **Back** button to return to the Maintain Single Regimens window.

For more information, see:

- [Changing the Drug Assigned to a Treatment Regimen](#)

Changing the Drug Assigned to a Treatment Regimen

To change the drug assigned to a treatment regimen you use the Maintain Product Master for Single Treatments window. You must complete this form as part of the study replication process. From the **Design** menu, select **Treatments**, then choose **Drug for a Regimen**. By default, the system assigns replicated single treatment regimens to a drug called NOT AVAILABLE. You can then use this module to assign the regimens to the appropriate drug in your inventory.

The default multiple listing window of this form does not display all the details you can specify about the treatment regimen. Click the **Single** button to specify route type and frequency type.

Maintaining Combined Treatment Regimens

Combined treatment regimens are the combination of two or more single treatment regimens. You can create and assign combined treatment regimens, sequence the order of administration, and assign daily dosages.

For more information, see:

- [Creating a New Combined Treatment Regimen](#)
- [Modifying a Combined Treatment Regimen](#)
- [Sequencing Regimens in a Combination Treatment Regimen](#)

Creating a New Combined Treatment Regimen

To create a new combined treatment regimen:

1. Navigate to **Design, Treatments**, then **Combined Regimen**. The Maintain Combined Treatment Regimens window opens, displaying all existing combined treatment regimens.
2. Enter a unique **Name** for the combined treatment regimen, of up to 60 characters.
3. In the **Label Abbreviation** field, you can enter an abbreviation or code for the regimen. The label abbreviation can be up to 20 characters long.
4. In the **Time Unit** field, choose from the list of values the unit of time that defines how the single regimens in this combined regimen should be spaced out. The time unit is typically "minutes." In query mode, you can use this field to specify search criteria. You can change this value at any time.
5. (Optional) Enter **Special Instructions** that describe how to administer the regimen. This field can store up to 200 characters.
6. Save.
7. Proceed to [Modifying a Combined Treatment Regimen](#) to define the subelements of this combined treatment pattern.

Modifying a Combined Treatment Regimen

To modify a combined treatment regimen:

1. From the **Design** menu, select **Treatment**, choose **Combined Regimens**, then choose a combined treatment regimen. You can click the **Single** or **Multi** buttons to toggle the window between single- and multi-record view.
2. Click the **Drugs** button to open the Select a Drug for the Single Regimen window. From this window, you can select a drug as a starting point in adding a new single treatment to the list of combined treatment regimens.
3. Within the Select a Drug for the Single Regimen window, you can also click the **Single Treatments in Drug** button to display the single treatment regimens associated with this drug. You can add single treatments to the drug here, or control their inclusion to the combined treatment regimen by selecting or clearing the **Included?** field.
4. Click the **Treatments** button to list the single treatment regimens already assigned to the combined treatment and follow these steps:
 - a. Select a single regimen to add to the combined treatment regimen.
 - b. Add or remove regimens from the combination. You can specify whether a single regimen is optional in the combined regimen.
 - Click the **Remove** button to remove a single regimen from the combined regimen.
 - Click the **Linkage** button to define the sequence to administer the single treatment regimens. You can also define the minimum and maximum wait times, which are expressed in the units assigned to the combined treatment regimen.
5. Click the **Daily Doses** button to record dosing details.

Sequencing Regimens in a Combination Treatment Regimen

This option enables you to specify the sequence in which the single treatment regimens in a combined treatment regimen should be taken. It also enables you to specify the wait time between the single regimens.

To sequence regimens in a combination treatment regimen, follow these steps:

1. Navigate to **Design, Treatments**, then **Combined Regimen**, then select the combined regimen to sequence.
2. Click the **Treatments** button. The system displays the existing single treatments.
3. Select a single treatment regimen to sequence.
4. Click the **Linkage** button. The system displays the selected treatment in the **Current Treatment** field.
5. Use the list function to enter the names of the preceding and following single treatment regimens.
6. Specify the minimum and maximum wait times between treatments. The system assigns the same unit of measure used to define the treatment.
7. Select **Optional** if the single regimen is not a mandatory part of the combination.
8. Enter any special instructions, of up to 200 characters.
9. Save.

Maintaining Patterns

Patterns impose a centralized control over the codes that identify treatment patterns in analysis and reports. If you do not require this control, define some dummy pattern codes.

This section describes how to define a new pattern. You can also use the Maintain Patterns window to modify existing patterns and to delete them from the database. All fields in the window are editable even after you save the pattern record.

To create a new pattern:

1. From the **Design** menu, select **Treatments**, then select **Patterns**. The Maintain Patterns window opens, displaying all patterns in the database.
2. Add a record.
3. Enter a unique **Code** to identify this treatment pattern. The code can be up to eight characters long.
4. Enter a **Description** of this treatment pattern. This value can be free form text, and up to 70 characters long. The description automatically appears every time you enter the code for this treatment pattern.
5. Enter **Text** that describes the treatment pattern at length. This value can be as long as 200 characters.
6. Select the **Active?** box to make this pattern record active, or clear it to make the record inactive. Active pattern codes appear when you invoke the list of values from the **Code** field in the Maintain Treatment Patterns window.
7. Save.

Maintaining Treatment Patterns

A **treatment pattern** represents the treatment to be given to a patient. If your only interest is in having treatment patterns to randomize, then you must only define a pattern code and description. However, you can define treatment patterns to include the treatment regimens in the treatment pattern and their relationship to planned study Intervals. You can then use this information to define the intended treatment. You cannot change the pattern code from this window once it has been used in a randomization. However, designers with the appropriate privileges can change the code by navigating to **Design, Treatments**, and then selecting **Treatment Pattern Codes**. Select a study, if it differs from the study of your current session, and click the **Treatment Patterns** button.

1. Navigate to **Design, Treatments**, and then **Treatment Patterns**. Select the study version, then click the **Treatment Patterns** button to reach the Treatment Patterns window.
2. Choose a **Pattern Code** from the list of values. When you select one, the window populates the corresponding **Treatment Pattern Name**.
3. Enter a **Label Code** for this treatment pattern.
4. Save. Oracle Clinical commits this treatment pattern to the database.
5. Click the **Regimens in Treatment Pattern** button to assign treatment patterns regimens. See [Assigning Treatment Pattern Regimens](#).
6. Click the **Drugs** button to display the drugs that are assigned to treatment patterns. See [Displaying Drugs Assigned to Treatment Patterns](#).
7. Click the **Copy Treatment Pattern** button to display the drugs that are assigned to treatment patterns. See [Copying Treatment Patterns](#).
8. Save.

For more information, see:

- [Assigning Treatment Pattern Regimens](#)
- [Displaying Drugs Assigned to Treatment Patterns](#)
- [Copying Treatment Patterns](#)

Assigning Treatment Pattern Regimens

When you click the **Regimens in Treatment Patterns** button in the Treatment Patterns window, the Treatment Pattern Regimens window appears, listing the regimens assigned to the selected treatment pattern. You can control the order of administration of treatment regimens by assigning them to the study's planned Intervals. Assigning regimens to Intervals also controls how long the treatment regimens should be taken.

The Treatment Pattern Regimens window has two navigation buttons:

- **Details of This Treatment Regimen** displays the Treatment Regimen Details window.
- **Assign Regimen to Intervals** displays the Study Intervals Assigned window.

The Study Intervals Assigned window has two function buttons: **Assign Regimen to this Interval**, and **Deassign Regimen to this Interval**. They perform the functions

they describe once you have selected an Interval for assignment to/deassignment from the treatment regimen you selected earlier. See [Defining a Study Schedule](#).

Displaying Drugs Assigned to Treatment Patterns

When you click the **Drugs** button in the Treatment Patterns window, the Drugs window appears for the selected study version. Selecting a drug can help in locating a regimen.

The Drugs window of the Maintain Treatment Patterns form has one button, **Single Treatments in Drug**, which displays the treatment regimens window for the selected drug.

You can locate a regimen to assign to a treatment pattern by tracing the drug associated to it:

1. Select the drug used by the selected treatment pattern.
2. Click the **Regimens for this Drug** button.
3. Select a regimen.
4. Click the **Add this Regimen to Treatment Pattern** button. The system displays the currently assigned regimens in this treatment pattern. This option displays all of the attributes for the treatment regimen assigned to the treatment pattern.

Copying Treatment Patterns

Use this option to copy an existing treatment pattern. The system copies the set of treatment regimens composing the treatment pattern and any of their assignments to planned study Intervals.

When you click the **Copy Treatment Patterns** button in the Treatment Patterns window, the treatment patterns for the study version you selected are copied and appear on the list in the window with the date copied. The system then prompts you for the pattern code to assign to the new treatment pattern.

Maintaining Treatment Pattern Codes

The Maintain Treatment Pattern Codes module supports all of the tasks allowed in the Maintain Treatment Patterns module, plus one significant additional option. You can update the treatment pattern code even *after* a study has been randomized in the Maintain Treatment Pattern Codes module, which is not allowed in the Maintain Treatment Patterns module.

The pattern code links the actual medication given to a patient with the treatments used in analysis.

To maintain a treatment pattern code:

1. From the **Design** menu, select **Treatments**, then choose **Treatment Pattern Codes**. The Maintain Treatment Patterns and Pattern Codes window opens, with the Clinical Study Versions window selected.
2. Select a study if it differs from the study of your current session, then click the **Treatment Patterns** button. The Treatment Patterns window opens.
3. Modify the treatment patterns and codes in the same manner as in [Maintaining Treatment Patterns](#).

Defining a Study Schedule

In this section:

- [Maintaining Intervals](#)
- [Maintaining Phases](#)
- [Maintaining Periods](#)
- [Maintaining Sub-periods](#)
- [Assigning Planned Processes and Procedures](#)

Maintaining Intervals

You can organize the expected progress of patients through a study into **Intervals**, creating a schedule, or timeline. The most general Interval is the **phase**. If necessary, you can create **periods**, which are divisions within a phase. You can create also **sub-periods** within periods. All three Interval types have the same attributes, except that you can view some randomization details at the phase level.

You can change the name of a phase at any time. You can also add a phase to a study at any time. You can delete a phase from a study if no CPEs are assigned to it; from the **Design** menu, select **Schedule**, then choose **Events**. The system cautions you if you attempt to delete a phase with periods assigned to it. In query mode, you can specify search criteria in this field.

Oracle Clinical imposes no restrictions on how you model the schedule. However, the way you define Intervals has implications for other parts of study design:

- **Flexible Studies** If you are creating a flexible study, bear in mind that you can enable Intervals and bypass to Intervals based on see [Defining Intervals and CPEs for Flexible Studies](#).
- **Page Tracking** If you are using Page Tracking (for paper-based studies), define Intervals to enforce minimum and maximum Intervals for when a visit can occur; see [Using CRF Page Tracking](#).

To create an Interval:

1. From the **Design** menu, select **Schedule**, then select **Intervals**. The Clinical Study Versions window opens.
2. Choose a study version and click **Phases**. The Maintain Intervals window opens.
3. Add a row.
4. Enter the **Phase Name**, which uniquely identifies a phase within a study. Sample phase names include Screening, Baseline, and Dosing.

When you enter the phase name, the window populates its **Short Name** automatically.

5. Define the details of the phase:
 - a. Choose a **Phase Type** from the list of values. The phase type classifies the phase according to its major objective, and the candidate values are drawn from the installation reference codelist PLAN STUDY INT TYPE CODE. In query mode, you can use this field to specify search criteria.

- b. Choose a **Time Unit** for this phase from the list of values.
 - c. Choose a **Blind Type** code that describes the type of blinding for this phase of the study. The candidate values in this list are drawn from the installation reference codelist BLIND TYPE CODE. In query mode, you can use this field to specify search criteria.
 - d. Enter the **Minimum Duration** value, which is a number that represents the minimum length of time for this phase of the study. This field, in conjunction with the **Time Unit** and **Maximum Duration** fields, describes the timespan for the phase.

The **Minimum Duration** must be less than or equal to the value of the **Maximum Duration** field. The system uses the value in this field to identify Off-Target DCMs.
 - e. Enter the **Maximum Duration** value. The system uses the value in this field to identify Overdue DCMs.
 - f. Select the **Opt?** box if you want to indicate that the phase is optional for patients in the study. You can change this setting at any time.
 - g. (Optional) Specify an **End Date** in DD-MON-YYYY format. This value is the date by which all patients are expected to complete this phase of the study.
6. Save.

Maintaining Phases

Companies typically use phases to classify different protocol objectives, such as screening and dosing. You can also control study design criteria like blinding at this level.

Your company specifies the types of phases, typically with values like Screening, Baseline, or Post_Study. The system does not restrict how you define these Intervals.

To create a phase:

1. Navigate to **Design, Schedule**, and then **Intervals**.
2. Select a study. The system orders the studies by study code sequence, then by version sequence within each study code. Fields here are display-only.
3. Click the **Phases** button to open the Phases window.
4. Specify the phases's details. You can query the multi-record window to view the phases assigned to the selected study. You can also click the **Single** button to see details of a selected study in one window.

When you create a phase for a clinical study version, the system also creates a clinical study history to record any changes in assigned randomization access status types. The system records the phase's creator, the time of creation, and the time at which it received its randomization access planned study Intervals.

You can also select and delete a phase in this window. You can delete a phase only if there are no CPEs linked to it. The system also deletes any associated periods or sub-periods. Oracle Clinical alerts you to confirm that you want to delete the phase.

5. Save.

There are two navigation buttons at the bottom of the window:

- The **Periods** button opens a window where you can subdivide a study's phases. See the following section for instructions.

- The **Resequence** button opens a window where you can re-order the study's phases. See [Resequencing Phases, Periods and Sub-periods](#).

Maintaining Periods

You can divide a phase into periods, if necessary. If you delete a period, the system deletes its sub-periods. You can only delete a period if there are no CPEs (visits) assigned to it.

To create, update, or delete periods of the selected phase:

1. From the **Design** menu, select **Schedule**, then select **Intervals**.
2. Select a study. Studies are displayed by study code sequence, then by version sequence within each study code. Fields here are display-only.
3. Click the **Phases** button to open the Phases window.
4. Select the phase to modify periods. You can only view periods linked to the phase you select.
5. Click the **Periods** button. The system displays the details of the periods currently assigned to the selected phase.
6. Enter a new period, delete, or modify the current periods' details.

Selecting the **Opt?** box sets the Interval as optional for patients in the study. Clear the box to make the Interval mandatory.

There are two navigation buttons at the bottom of the window:

- The **Sub-periods** button opens a window where you can subdivide a study's periods. See the following section for instructions.
- The **Resequence** button opens a window where you can re-order the study's phases. See [Resequencing Phases, Periods and Sub-periods](#) for instructions.

Maintaining Sub-periods

You can further divide study Intervals by adding sub-periods to periods. To create, update, or delete sub-periods of the selected period, click the **Sub-periods** button in the Periods window. The fields in the Sub-periods window function in the same way as those in the Periods window. See [Maintaining Periods](#) for instructions for this window, substituting the term "sub-periods" for "periods."

For more information, see:

- [Resequencing Phases, Periods and Sub-periods](#)

Resequencing Phases, Periods and Sub-periods

This section describes resequencing phases, periods, and sub-periods of a clinical study version. You may need to add a phase, period, or sub-period, or change the sequence, among existing ones. Each Phases, Periods, and Sub-periods window has a **Resequence** button.

To resequence:

1. Select the phase, period, or sub-period to resequence.
2. Select the **Resequence** button on the toolbar.

3. To mark the insertion point in the series, enter the name of the phase, period, or sub-period to precede the one you named in Step 1.

For example, to change the phase sequence A-B-C-D-E to the new sequence A-C-B-D-E:

- a. Select C on the Phases window.
- b. Select the **Resequence** navigation button.
- c. Enter A as the new previous phase.

To make a phase the first phase—that is, to make D first in the sequence ABCD:

- a. Select D and resequence it to come after A, giving ADBC.
- b. Select A and resequence it to come after D.

Assigning Planned Processes and Procedures

In this section:

- [Maintaining Processes](#)
- [Maintaining Clinical Procedures](#)

Maintaining Processes

The use of Oracle Clinical processes is optional and is not included in the Easy Study Design utility. When you define a Clinical Planned Event (CPE) you must assign a process to it, but you can use a default value for most or all of your CPEs.

A *process* can serve two purposes:

- Since Oracle Clinical imposes no restrictions on what constitutes an event, assigning processes to events can, for example, help identify screening visits or dosing visits (though assignment to Intervals can accomplish the same thing), or differentiate between CPEs that correspond to visits and those used for AEs or ConMeds.
- **Grouping Clinical Procedures** You can define treatment details called "procedures" and assign one or more of them to a process.

You can delete a clinical planned process only if it is not assigned to any CPE.

To assign a process to link a version of a study:

1. Navigate to **Design, Schedule**, and then **Processes**.
2. Select a study version.
3. Click the **Processes** button. The system displays only those clinical planned processes linked to this version, in order of process name.
4. Create a new process, or modify or delete an existing process. If you delete a clinical planned process, the system also deletes all its clinical procedures.

Note:

The time unit in the Clinical Processes window is not the same as the time *when* the events should take place. This time unit defines *what happens* at the event.

The Clinical Processes window has one navigation button, **Procedures**, which displays the Clinical Procedures window.

Maintaining Clinical Procedures

Clinical procedures specify time ranges within a process when treatment regimens should take place. You can define the minimum and maximum time units relative to the start of the procedure or other procedures. You can also specify that a regimen take place under certain circumstances, which you define in the **Trigger Condition** field. Typically, a process contains more than one procedure.

From the **Design** menu, select **Schedule**, choose **Processes**, click the **Processes** button, and then the **Procedures** button to open the Procedures window. It lists all clinical procedures associated with the selected process. The clinical procedures display in description sequence.

Enter values for one or both of the following sets of timing specifications. These fields are informational only (unlike similar fields in CPEs). They are not actually used by any functions.

- **Min Offset Start** and **Max Offset Start** (range since the start of the procedure)
- **Min Offset Prev.** and **Max Offset Prev.** (range since the end of the last procedure)

 **Note:**

Trigger Condition has nothing to do with the trigger condition that is part of a Rule defined for a flexible study in an enhanced DCI Book.

Creating Clinical Planned Events

A **clinical planned event** (CPE) usually corresponds to a visit. You can also define a CPE for the purpose of associating it with DCIs (corresponding to CRFs) that you want to have available in a DCI Book without associating them with a particular visit, such as AEs or ConMeds. If you are defining Processes in Oracle Clinical, you can create a CPE to schedule a Process during an Interval.

If you are working in a flexible study, see [Defining Intervals and CPEs for Flexible Studies](#) for more information.

To create a clinical planned event:

1. From the **Design** menu, select **Schedule**, then select **Events**. The Study Versions window opens.
2. Choose a study version and click **Events**. The Maintain Clinical Planned Events window opens, displaying all CPEs in the database.
3. Insert a row, then enter a title for the event in the **Event Name** field. The Event Name must be unique within this study version. You may want to take page numbering into consideration when you name your CPEs, especially in a flexible study; see [Sample Numbering Schemes](#).
4. Specify the event details:

- a. Choose a **Process Name** for the event from the list of values. A process is a collection of clinical planned procedures performed at a CPE; a process occurs at each CPE.
The values in this list are populated in the Maintain Processes window.
- b. Choose an **Interval Name** from the list of values. Each CPE occurs during an Interval, and you can choose a phase, period, sub-period, or the entire study as the Interval for this event.

 **Note:**

When you create a study using Easy Study Design, the system automatically creates a Phase Interval called DEFAULT STUDY PHASE. If you are creating a flexible study you cannot use this default Interval and must explicitly create Intervals.

 **Note:**

The system prevents you from changing a CPE's Interval if the study is flexible and has an active DCI Book that contains Interval rules. If the DCI Book is provisional, the system sets its validation status to Pending. When you activate the book or run the validation job explicitly, the job determines if the changes are valid.

- c. Enter a **Visit Number**. A CPE's visit number must be unique within this study version. The Visit Number for each CPE controls the order in which they occur in the DCI Book and in the study. CPEs with a lower Visit No. cannot occur after a CPE with a higher Visit No., so be sure to give CPEs the right Visit No. relative to one another. You can add a CPE later, but you must renumber all subsequent CPEs to maintain the correct relative order.
 - d. If this event describes the termination visit, choose a **Termination Code**. Valid codes are "early" or "normal." You can change this field at any time, provided the study has at most one early and one normal termination visit.
 - e. Select the **Dispensing Visit?** field if you want to indicate that the visit includes the administration of a study drug.
 - f. Populate the two pairs of Minimum/Maximum offset parameters:
 - **Offsets from Interval Start** mark the event's time frame from the start of the Interval.
 - **Offsets from Previous Event** mark the event's time frame from the last event.
 - g. Enter a **Label Code**, which is a one-character code to print on the kit label to represent this event.
 - h. (Optional) Enter a **Trigger Event**, which describes the conditions that must exist before this event can occur.
5. Save.

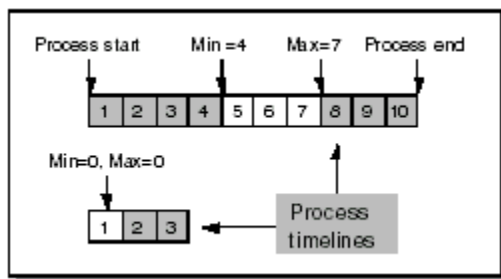
For more information, see:

- [Using CPE Offsets](#)

Using CPE Offsets

The *minimum offset* is the minimum number of time units that must elapse before a particular CPE can take place. The *maximum offset* is the maximum number of time units that can elapse before a particular CPE must take place.

The following diagram shows how to use offsets to specify the time frame within which a planned event should take place:



For a CPE that can occur on day 5, 6, or 7, the minimum offset is 4 and the maximum is 7. A CPE on Day 1 has a minimum and maximum offset of zero.

Oracle Clinical reports missing CPEs to facilitate the DCM tracking process. The system automatically calculates the minimum and maximum offset of each CPE from the start of the study.

The system derives the calculation in the following manner:

- The minimum offset from the start of the study is the sum of the minimum duration for the Intervals that occur before the start of the Interval to which the CPE belongs, plus the offset for the CPE within the Interval.
- Similarly, the system calculates the maximum offset based on the maximum duration and offset.

For more information, see:

- [Examples of Offsets](#)

Examples of Offsets

Following are examples of calculating when a CPE should occur within a clinical study version.

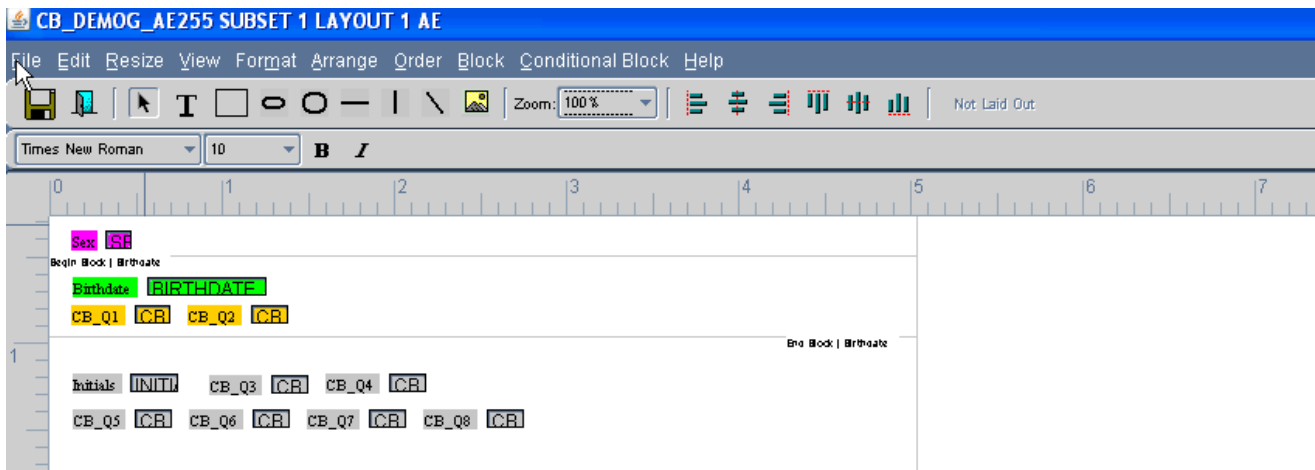
Simple examples:

CPE1 occurs 6 days into the study.

CPE2 occurs 9 days into the study.

CPE3 occurs 5 days into Phase 2, which comes after the 14 days of Phase 1.

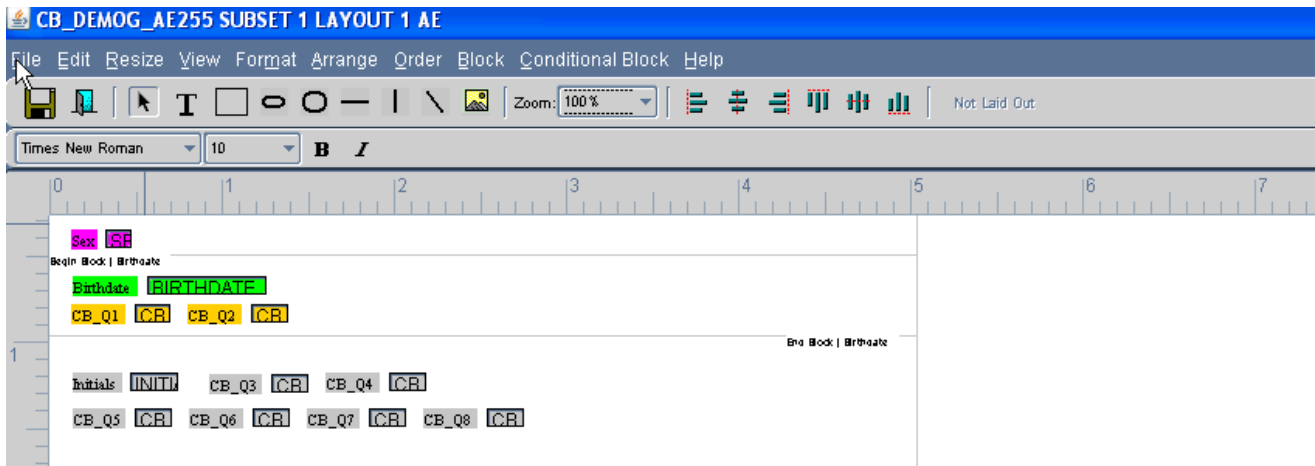
So, visit 3 occurs $14 + 5 = 19$ days into the study.



Different Minimum and Maximum Example

CPE1 occurs 6 to 8 days into the study.
 CPE2 occurs 9 to 11 days into the study.
 CPE3 occurs 5 to 8 days into phase 2, which comes after the 14 days of phase 1.

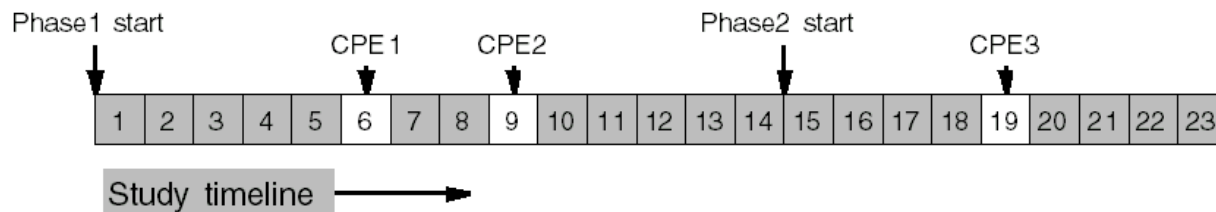
So, visit 3 occurs after a minimum of $14 + 5 = 19$ days into the study and by a maximum of $14 + 8 = 22$ days into the study.



Offsets from Interval Example

CPE1 occurs 6 days into the study.
 CPE2 occurs 3 days after visit 1; so visit 2 occurs $6 + 3 = 9$ days into the study.
 CPE3 occurs 5 days into Phase 2, which comes after the 14 days of Phase 1.

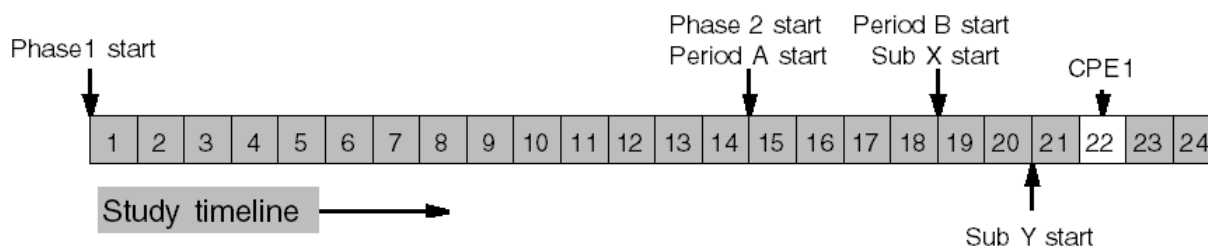
So, CPE3 occurs $14 + 5 = 19$ days into the study.



Periods and Sub-periods Example

CPE1 occurs 2 days into Sub-period Y of Period B of Phase 2.
 Sub-period Y comes after Sub-period X, which is 2 days long.
 Period B comes after Period A, which is 4 days long.
 Phase 2 comes after Phase 1, which is 14 days long.

So, CPE1 occurs after Phase 1 + Period A + Sub-period X, plus 2 days into Sub-period Y: $14 + 4 + 2 + 2 = 22$ days.



It is possible to mix the different cases. For example, minimum and maximum offsets and duration can be combined with periods and sub-periods. Also, an event can always be defined as occurring a minimum and maximum time after the previous event.

6

Stratification and Randomization

In Oracle Clinical the term **strata** means groupings of patients that have common characteristics; they are representations of particular expressions of factors. **Randomization** is the process of hiding patients' treatments by assigning treatment patterns to patient positions in an unpredictable order. **Patient positions** represent potential study participants.

Once defined, you can use strata in any study, as needed. You can construct them by combining factors. Each factor describes a single characteristic, such as sex or age. Implementing the system's stratification utilities can minimize confounding a study's statistical results due to the uneven distribution of treatments to patients.

In this section:

- [Stratification](#)
- [Randomization](#)
- [Randomization Maintenance](#)

Stratification

Oracle Clinical provides stratification utilities to ensure that the groups you select are mutually exclusive and represent a full cross-section of your study's population. For example, you could stratify a study according to the following groups:

male smokers over 65

male nonsmokers over 65

males 40 to 65

female smokers 45 to 65

females over 65

You create strata with a view to preserving statistical significance. For example, superficially the following two examples of stratification for a study seem to be the same. However, these stratification patterns impose different constraints on randomization.

- two strata: male smokers; male nonsmokers and women
- three strata: male smokers; male nonsmokers; women

The first could result in a randomization where 100 patient positions include male smokers, but 200 other patient positions include *either* male nonsmokers *or* females.

The second stratification pattern could result in a randomization where 100 patient positions include male smokers, and 200 include male nonsmokers and females. Both stratifications could be useful, but each would have a different statistical significance.

For more information, see:

- [Maintaining Factors](#)
- [Maintaining Single Strata](#)

- [Maintaining Combination Strata and Combined Combination Strata](#)
- [Maintaining a Study Stratification Factor](#)

Maintaining Factors

As a first step in creating strata, clinical study enrollment criteria, clinical study termination criteria, and treatment regimens, you create factors. Factors are attributes of a patient you can measure, rather than the actual values of attributes in individual cases.

To reach the Maintain Factors window, from the **Design** menu, select **Strata**, then select **Factors**.

For more information, see:

- [Glossary of Terms for Factors](#)

Glossary of Terms for Factors

Ranges represent variables requiring further qualification to make them meaningful to the study. Examples are AGE, WEIGHT, and BLOOD PRESSURE, which require maximum and minimum values to specify their roles in the study.

Values, on the other hand, are variables such as BLOOD GROUP, which require only a single value to make them meaningful.

The **Range/Value** box in the Factors window designates a variable as either a range or a value. You can change the flag only if the factor is not used by any clinical study enrollment criteria, clinical study termination criteria, treatment regimen, or strata; otherwise you get an error message.

A retired factor can still relate to existing strata, but cannot be used to create a new stratum. A retired factor can be reactivated by setting the **Active?** box back to selected.

You can delete a factor only if it is not being used by any strata, clinical study enrollment criteria, clinical study termination criteria, or treatment regimen. You get an error message if you try to delete a factor that is in use. For a factor, define a code, description, either a value or range, and whether it is available for assignment to strata. You must create all of the factors needed to compose each stratum.

Maintaining Single Strata

To create, modify, and delete single strata, created for use in all clinical studies, use the Maintain Single Strata window.

To access this window, from the **Design** menu, select **Strata**, then **Single Strata**; this initially displays the Factors window.

For more information, see:

- [Selecting a Factor](#)
- [Selecting a Stratum](#)

Selecting a Factor

You select a factor from the Factors window list of all active factors before you can proceed to create a single stratum. This is the factor for which you create or update the single stratum.

You cannot insert, update, or delete factors from this window. You can only select a factor for which to create a single stratum

Selecting a Stratum

The **Strata** button in the Factors window of the Maintain Single Strata window displays the Strata window. Here you can create, update, or delete single strata for the factor you selected in the Factors window.

A new stratum can only be created for an active factor.

You can delete a single stratum only if it is not used in any combined strata or combined combination strata, and if no study stratification factors assign it to a clinical study version. To delete the stratum, select **Delete Record**.

The minimum/maximum values and the single value are mutually exclusive: if your factor is of type RANGE, you must enter minimum and maximum values; if the factor is of type VALUE, you must enter a single value. Select a factor to define single strata. You can create the factor records by selecting **Strata** from the **Design** menu, and choosing **Factors**. Then click the **Strata** button.

Maintaining Combination Strata and Combined Combination Strata

You link single strata to create combination strata, logically linked by AND, or combined combination strata, logically linked by OR.

To link single strata with an AND statement, open the Maintain Combination Strata window: from the **Design** menu, select **Strata**, then choose **Combination Strata**.

To link single strata with an OR statement, open the Maintain Combination Strata window: from the **Design** menu, select **Strata**, then choose **Combined Comb Strata**.

Because the two modules operate identically in every other respect, the following instructions apply to both modules.

Each window has one field, **Description**, which is required and must be unique. Enter a description of the stratum, indicating the presence of AND or OR logic.

The forms for maintaining Combination Strata and Combined Combination Strata each have an associated Available Factors window and a Strata for Factor window, reached by pressing the **Factor** or **Strata** button, respectively. These windows allow you to change the selected Combination or Combined Combination Stratum by adding or removing constituent (strata).

For more information, see:

- [Assigning Strata to Combination or Combined Combination Strata](#)
- [Displaying Strata Assigned to Combination or Combined Combination Strata](#)

Assigning Strata to Combination or Combined Combination Strata

Select a factor and then click **Strata** to see the strata assigned to the factor. This window displays the strata for the factor selected earlier. Click the **Add** or **Remove** button to create the set of strata in the strata you are creating.

The following buttons appear in the Strata for Factor window. Add or remove special factor-related strata to the selected combination stratum with these buttons:

- **Add** assigns the selected stratum to the strata.
- **Remove** removes the selected stratum.

Displaying Strata Assigned to Combination or Combined Combination Strata

Use this function to display the strata already assigned to a combination strata or combined combination strata and allow them to be removed.

Maintaining a Study Stratification Factor

The assignment of a stratum to a clinical study version is a *study stratification factor*. To create and maintain study stratification factors, see:

- [Selecting a Clinical Study Version](#)
- [Maintaining Strata](#)

Selecting a Clinical Study Version

To assign a stratum to a clinical study version, navigate to **Design, Strata, Stratification**. Select a clinical study version to work with from the list. The system selects the most recently accessed clinical study version during the current session. If you have not selected a study, the window displays all existing clinical study versions.

Maintaining Strata

To reach the Study Factors window, which lists the strata assigned to the study, navigate to **Design, Strata, Stratification**, and click the **Stratum Assigned to the Study** button. **Stratum Description** is the lone field for this window. It is a display-only description of the stratum that the factor is for.

Two function-specific buttons appear in the Study Stratification Factors window:

- **Factors** displays the Factors window. Assign one or more of the associated strata to your study.
- **Details of the Stratum** displays the Strata window in single-record format. You can also reach the Strata window from the Factors window, via the Strata window. Select a stratum to work with while in this window.

The following buttons are in the Strata window:

- **Details of the Stratum** displays the window Details of Selected Stratum. It shows the details of a stratum assigned to a study or of a stratum for a factor before assignment to a study.

- **Assign Stratum to Clinical Study** enables you to add a stratum to the list of assigned strata.

Randomization

A study's treatment patterns specify the drugs to be administered and the medical procedures to be performed. You assign them to patient positions, creating treatment assignments.

For more information, see:

- [Some Concepts](#)
- [Access](#)
- [Choosing a Randomization Type](#)
- [Deleting, Locking/Unlocking, and Verifying Randomizations](#)
- [Batch Loading a Randomization](#)

Some Concepts

Principle 1

The treatment code of a treatment assignment must be unique within a clinical study version, and so randomizations must not overlap. Recipes (instructions creating a randomization, including the treatment patterns involved, their ratio, and the number of related treatment assignments) are overlapping when they contain the same treatment assignment.

For example, Recipe A creates treatment codes 1 to 10; Recipe B creates 1 to 20. Your recipes can overlap, if necessary, while you experiment with alternate recipes. But you cannot use overlapping recipes when creating randomizations. To prevent overlap, when you enter parameters for a randomization the system validates *within a recipe* to trap for overlapping definitions. However, because the system does not detect overlaps *between recipes* until you randomize, it is then that you receive an error if you try to create a treatment assignment with a code already in use.

Principle 2

Oracle Clinical assumes that, due to experimental design requirements, your company separates the role of an unblinded randomization analyst, who generates a randomization, from the blinded person who enters the randomization recipe.

Principle 3

The sequence for entering randomization recipes and attributes is flexible; you can continue to modify recipes until you generate a randomization from them. Once it is generated, however, you can only make changes that expand the randomization but that do not change the existing treatment assignments.

Access

Access refers to tasks involved in an interim analysis. An interim analysis is when a quick assessment is made of the study results to decide if it should continue or if it should be stopped because of lack of efficacy (or safety).

For more information, see:

- [Interim Analysis](#)

- [Randomization Privileges](#)
- [Maintaining Blinding in a Study](#)

Interim Analysis

Several functions control who can perform an interim analysis, when, and on what data. In particular, the randomization access status code controls who is allowed to see the randomization in a study. It can have the following values:

Value	Meaning
Open	The study is <i>open label</i> from the start; everybody can see the randomization details.
Closed	The study is currently blinded, so only unblinded users can see the randomization details.
Release	The study was initially blinded (<i>Closed</i>) but is now over and the randomization opened, or released, for analysis.
Access	Allows specified users access to the randomization to perform an interim analysis. (See Creating or Revoking an Access View .)
Multi	Access controlled at the phase level and not at the study level.

A dummy randomization is available to establish the analysis environment. It has no correlation to the real randomization; however, you can use it to test reporting and extracting facilities. The system creates the dummy randomization when you randomize the study. The actual randomization becomes available for analysis when the study's blinding stage is complete.

Randomization Privileges

Most functions within the Randomization subsystem require the privilege `RXC RAND`. Typically companies limit `RXC RAND` privileges to a small group of specialized users who are not members of the study management teams.

Maintaining Blinding in a Study

While a study is still blind, only privileged users have access to its randomization. If an emergency situation occurs, the system provides ways to disclose a patient's treatment pattern. (See [Disclosing Patient Treatment Assignments](#).) If disclosure occurs, the system generates a record of the disclosure as a *blind break*.

Choosing a Randomization Type

Oracle Clinical offers four types of randomization: version-level, strata-level, version block-level, and strata block-level—each described in the following sections.

If you create either a version-blocked or strata-level blocked randomization, and you intend to create only one randomization recipe, you can use the easy randomization utility available for these randomization types. From the **Design** menu, select **Version Blocked Randomization** (or **Strata Blocked Randomization**), then select **Easy Randomization**. Except for allowing only one recipe, the Easy Randomization utilities are identical to the equivalent standard Randomization utilities; if you want different recipes—for example, different ratios of treatment patterns for different patients—use the standard utilities.

For more information, see:

- [Version-level Randomization](#)
- [Strata-Level Randomization](#)
- [Version Block-Level Randomization](#)
- [Strata Block-Level Randomization](#)

Version-level Randomization

The version type of randomization generates sequential treatment assignments for the same treatment pattern. There is nothing random about this method—the recipe explicitly defines what you produce. For example, kits 1 to 10 on Treatment Pattern A, 11 to 20 on Treatment Pattern B, etc. Designers typically apply version randomization to Phase I open label-type studies.

The system assigns the following type codes to version-level randomizations:

- **SVTP**: Study Version Level, Treatment Pattern
- **SVTPX**: Study Version Level, Treatment Pattern, Extra Assignments Only

The system maintains the recipe within the treatment pattern records. For each treatment pattern, you can specify how many treatment assignments you require and the code from which to start coding them.

To create a version-level randomization, from the **Design** menu, select **Randomization**, choose **Version Randomization**, then select **Treatment Patterns**. Choose a study, then click the **Treatment Patterns** button. Specify the parameters of your recipe and use the **Regimens in Treatment Pattern** and **Drugs** buttons to access windows to further specify the randomization. Click the **Copy Treatment Pattern** button to copy a pre-existing treatment pattern within your study.

To create a version-level randomization from a specific treatment assignment, from the **Design** menu, select **Randomization**, choose **Version Randomization**, then select **Treatment Assignments**.

Strata-Level Randomization

The strata-level type of randomization is similar to the version-level randomization described above, except that each stratum in the study can have its own set of treatment assignments. For example, males can use kits 1 to 10 on Treatment Pattern A, 11 to 20 on Treatment Pattern B; females can use kits 100 to 110 on Pattern A, etc. This is also designed to apply to Phase I open label-type studies.

The following type codes are assigned to stratum-level randomizations:

- **STFP**: Stratum Level, Treatment Pattern
- **SFTPX**: Stratum Level, Treatment Pattern, Extra Only

From the **Design** menu, select **Randomization**, choose **Strata Randomization**, and then select **Treatment Patterns**. Choose a study, then click the **Treatment Patterns** button. Specify the treatment pattern information, then use the **Regimens in Treatment Pattern** and **Drugs** buttons to access windows to further specify the treatment pattern. Click the **Copy Treatment Pattern** button to copy a treatment pattern already existing in your study.

To maintain the strata assigned to a study, from the **Design** menu, select **Randomization**, choose **Strata Randomization**, and then **Stratification**. Choose a study, then click the

Stratum Assigned to the Study button to view the descriptions of the strata assigned to the study. Click the **Factors** button to associate the strata to a factor; click the buttons, **Details of the Stratum**, and **Treatment Patterns** to refine the randomization.

To create a strata-level randomization from a specific strata, from the **Design** menu, select **Randomization**, choose **Strata Randomization**, and then **Treatment Assignments**. Select the strata and click the **Randomize** button.

Version Block-Level Randomization

The version block-level type of randomization generates a random set of treatment assignments. Within the randomization, each block of treatment assignments contains the same combination of treatment patterns but in different sequences. For example, two patterns A and B and a block size of four could produce ABBA BABA ABAB BBAA ABBA, etc.

The following type codes are assigned to study version block definition-level randomization:

- **SVBD**: Study Version Level, Block Definition
- **SVBDX**: Study Version Level, Block Definition, Extra Only

To create a version-blocked level randomization, from the **Design** menu, select **Randomization**, choose **Version Blocked Randomizations**, and then **Treatment Patterns**. Choose a study, then click the **Treatment Patterns** button. Specify the parameters of your recipe and use the **Regimens in Treatment Pattern** and **Drugs** buttons to access windows to further specify the randomization. Click the **Copy Treatment Pattern** button to copy a pre-existing treatment pattern within your study.

To create a version block-level randomization, from the **Design** menu, select **Randomization**, choose **Version Randomization**, then select **Block Definition**. Choose a study, then click the **Block Definitions** button. Specify the parameters of your recipe, then click the **Treatment Pattern** button to further specify the randomization. Click the **Copy Treatment Pattern** button to copy an existing treatment pattern in your study.

To create a version block-level randomization from a specific treatment assignment, navigate to **Design, Randomization, Version Blocked Randomization**, and then **Treatment Assignments**.

Strata Block-Level Randomization

The strata block-level type of randomization is similar to the version block definition-level described above, except that the randomization belongs to a stratum of the study.

The following type codes are assigned to the stratum version block definition-level randomization:

- **SFBD**: Stratum Level, Block Definition
- **SFBDX**: Stratum Level, Block Definition Extra Only

To create a strata block-level randomization, from the **Design** menu, select **Randomization**, choose **Strata Blocked Randomization**, and then **Treatment Patterns**. Choose a study, then click the **Treatment Patterns** button. Specify your treatment patterns, and use the **Regimens in Treatment Pattern** and **Drugs** buttons to access windows to further specify the treatment pattern. Click the **Copy Treatment Pattern** button to copy a treatment pattern that exists in your study.

To create a version block-level randomization, from the **Design** menu, select **Randomization**, choose **Strata Blocked Randomization**, and then **Block Definition**. Choose a study, then click the **Block Definitions** button. Specify the parameters of your recipe, then click the **Treatment Pattern** button to further specify the randomization. Click the **Copy Treatment Pattern** button to copy a treatment pattern existing in your study.

To create a strata block defined-level randomization from a specific set of treatment assignments, from the **Design** menu, select **Randomization**, choose **Strata Blocked Randomization**, and then **Treatment Assignments**. Select a strata and block definition, then click the **Randomization** button.

Deleting, Locking/Unlocking, and Verifying Randomizations

Oracle Clinical randomization includes utilities to remove randomizations, to lock—that is, to prevent changes to randomization treatment assignments—and to unlock locked randomizations. Open the Randomizations window by navigating to **Design**, **Randomization**, **Randomization Type**, then **Treatment Assignments**, and clicking the **Show Randomizations** button.

To test for changes to a randomization, click the **Verify** button in this window. Oracle Clinical regenerates the randomization using the original seed number, and creates a list of differences between the original randomization and the verification randomization. The report is named `randval.out`.

Note:

Randomization reports are formatted in landscape mode, so be sure to configure your page setup to this mode before printing a hard copy of the report.

Batch Loading a Randomization

Oracle Clinical provides utilities to transfer the details of a randomization in or out of Oracle Clinical. These utilities handle the following situations:

- Downloading a copy of a study's randomization details from Oracle Clinical for use in another application
- Loading legacy randomization details into Oracle Clinical
- Unusual randomization requirements that Oracle Clinical's standard functionality cannot emulate

Note:

If you batch load a legacy 'Block' Randomization into Oracle Clinical, it will not result in a block randomization. The batch load process will never populate `TREATMENT_ASSIGNMENTS.RANDOM_BLOCK_ID` or create blocks. If you want a block randomization, you should use the Oracle Clinical functionality. In addition, if you download an Oracle Clinical system-generated randomization, edit it, and batch load it back into Oracle Clinical, the previously populated `RANDOM_BLOCK_ID` is no longer populated.

Before you can batch load a randomization into Oracle Clinical, you must define the treatment patterns and strata. Create strata by navigating to, **Strata**, then **Stratification**. Define treatment patterns by navigating to **Design, Treatments**, then **Treatment Patterns**. Batch loading a stratification randomization is the same as manual creation, except that you cannot specify the number of treatment kits.

See [Maintaining a Study Stratification Factor](#) for a description of stratification factors.

Load the file into the directory specified in the RAND_ROOT setting of the OCL_STATE local reference codelist. If RAND_USERDIRS in the same codelist is set to Y, load the file into the user-specific subdirectory for the user who will run the job. See the *Oracle Clinical Administrator's Guide* for more information.

In this section:

- [Downloading a Randomization](#)
- [Validating and Loading a Randomization](#)
- [Handling a Legacy Randomization](#)

Downloading a Randomization

You can create an ASCII file of all the randomization details in a study. Navigate to **Design, Randomization, Randomization Batch Load**, then **Download Randomization**.

This PSUB submission function requires one parameter, Description, which the system applies as the name of the ASCII output file. The formats for the file are shown in this section.

- **The name of the Randomization File:** must include, and defaults to, the directory specified in the RAND_ROOT setting of the OCL_STATE local reference codelist. If RAND_USERDIRS in the same codelist is set to Y, it is stored in the user-specific subdirectory for the user who runs the job. See the *Oracle Clinical Administrator's Guide* for more information.
- **The Study Code of a Study**

Only the output file name is passed as input. This file is stored in

Editing the Randomization Download

If you intend to force in a randomization (download, edit, and reload a randomization), the system limits you to the following changes:

- Altering the treatment pattern assigned to a treatment assignment code
- Creating new treatment assignments
- Deleting treatment assignments
- Changing the replacement and/or latest flags

File Format

The tables in this section specify the format of the data for the different randomization types.

First Card Type - All Randomization Types

From:	To:	Parameter
1	- 10	"STUDY"
11	- 25	Study Code
26	- 35	Version Code

Stratum Card Type - For Stratified Randomizations Only

One card for each stratum involved in the randomization.

From:	To:	Parameter
1	- 10	"STRATUM"
11	- 21	Stratum Id
22	- 51	Stratum Description
53	- 61	Starting Treatment Assignment Code
62	- 73	Ending Treatment Assignment Code

Non-stratum Card Type -For Non-stratified Randomizations Only

One card for the entire randomization.

From:	To:	Parameter
1	- 10	"NOSTRATUM"
11	- 51	Null
52	- 61	Starting Treatment Assignment Code
62	- 71	Ending Treatment Assignment Code

Treatment Pattern Card Type - For All Randomizations

One card for each treatment pattern involved in the study.

From:	To:	Parameter
1	- 10	"PATTERN"
11	- 25	Pattern Code
28	- 80	Treatment Pattern Description

Treatment Assignment Card Type - For All Randomizations

One card for each treatment assignment in the randomization.

From:	To:	Parameter
1	- 10	"TRTASG"
11	- 20	Treatment Assignment Code
21	- 30	Patient Code
-	31	Latest Assignment Flag
-	32	Replacement Patient Flag

From:	To:	Parameter
33	- 47	Owning Location Code

Match the structure of the force-in file as one of the following:

- STUDY
 - STRATUM
 - * PATTERN
 - * TRTASG
 - * TRTASG
 - * etc.
 - * PATTERN
 - * TRTASG
 - * PATTERN
 - * TRTASG
 - STRATUM
 - * PATTERN
 - * TRTASG
 - * PATTERN
 - * TRTASG
 - * PATTERN
 - * TRTASG

OR

- STUDY
 - NOSTRATUM
 - * PATTERN
 - * TRTASG
 - * TRTASG
 - * PATTERN
 - * TRTASG
 - * TRTASG
 - * PATTERN
 - * TRTASG
 - * TRTASG

Validating and Loading a Randomization

This form reloads an ASCII file containing the randomizations for a study, checking that it meets all randomization requirements. It is a standard PSUB submission form.

From the **Randomization** menu, select **Randomization Batch Load**, and choose **Validate and Load Randomization**. The system checks that the treatment patterns exist within the clinical study and that there are no existing locked treatment assignments. It then deletes all the existing randomization information and re-creates it from the file.

Copy the input randomization file into the directory specified in the RAND_ROOT setting of the OCL_STATE local reference codelist. If RAND_USERDIRS in the same codelist is set to Y, copy the file into the user-specific subdirectory for the user who will run the job. See the *Oracle Clinical Administrator's Guide* for more information.

Handling a Legacy Randomization

When you have a randomization created outside the system—a converted randomization—Oracle Clinical reads the randomization from the file and loads it into the database, translating external codes into the corresponding design codes for strata and treatment patterns. The system takes this input and creates treatment assignments and randomization records. Oracle Clinical also validates that the load process ran successfully.

Before you can execute the load, the clinical study, clinical study version, treatment patterns, and, possibly, strata, have to exist.

To load and validate a converted randomization:

1. Navigate to **Design, Randomization, Randomization Batch Load**, and **Load a Converted Randomization**.
2. Reformat a flat file from the existing legacy randomization system into Oracle Clinical's format. See "Converted File Format."
3. Copy the input randomization file into the directory specified in the RAND_ROOT setting of the OCL_STATE local reference codelist. If RAND_USERDIRS in the same codelist is set to Y, copy the file into the user-specific subdirectory for the user who will run the job. See the *Oracle Clinical Administrator's Guide* for more information.
4. Use the Maintain Conversion Specifications module to inform Oracle Clinical how to translate the treatment pattern and strata codes supplied by your legacy system into Oracle Clinical codes. See "Entering Conversion Parameters." Some randomization systems treat replacement patients as different strata. Oracle Clinical does not require this, so if a stratum exists only to separate replacement patients from normal patients, you can flag it as a replacement stratum and map it to the same stratum in Oracle Clinical as the normal stratum from the legacy system.
5. Load the converted randomization into Oracle Clinical.
6. Check the output report for any errors encountered in loading the data.
7. Optionally, run a report to read the original data file and compare it with the Oracle Clinical database to confirm that the data loaded correctly. See "Validating a Loaded Randomization."

Unlike downloading, editing, and loading Oracle Clinical data—processes that identify everything with Oracle Clinical codes and structure—converting and loading legacy data allows for code translation and for merging strata. It is more powerful but also more complex. Oracle recommends using the download format, if possible.

Entering Conversion Parameters

When converting randomization data from legacy systems into Oracle Clinical, you need to create conversion specifications to match the external data to Oracle Clinical parameters.

If the treatment pattern in the legacy system is based on the same code as the treatment pattern in Oracle Clinical, there is no need to enter a conversion parameter.

From the **Randomization** menu, select **Randomization Batch Load**, choose **Enter Conversion Parameters**, and then select the study. The Version window of Maintain Conversion Specifications has two buttons:

- **Strata** opens the Strata window, where you map the strata from the legacy system to the corresponding target Oracle Clinical codes. You can map each target stratum to more than one legacy stratum. Since some legacy systems use strata differently from Oracle Clinical's normal and replacement assignments, you can set a flag to indicate that Oracle Clinical merges the strata into Oracle Clinical's replacement assignments.
- **Patterns** opens the Treatment Pattern Conversions window of the Maintain Conversions window, where you map the legacy treatment pattern code (**Old Pattern** field) to the corresponding Oracle Clinical code. You can have only a one-to-one correspondence.

Converted File Format

The file format must conform to the following requirements:

- Left-align all character fields, with no leading zeros.
- Right-align and zero-fill all numeric fields.
- Left-align the version code, adding trailing spaces to fill the 10-character code.
- Treatment and patient codes are character fields but are limited to a single, leading, non-numeric character. Right-align these fields, and do not add leading zeros; for example, S1001.

Clinical Study Card Type

From:	To:	Parameter	Type	Length
1	-10	"STUDY"	CHAR	9
11	- 25	Study Code	CHAR	15
25	- 34	Null	CHAR	10
35	- 45	Starting Seed number	NUMBER	11
46	- 56	Ending Seed number	NUMBER	11
57	- 59	Block size	NUMBER	3

Stratum Card Type

From:	To:	Parameter	Type	Length
1	- 9	"STRATUM"	CHAR	9
11	- 14	Investigator code	NUMBER	5
15	- 16	Stratum number	NUMBER	2
17	- 31	Stratum abbreviation	CHAR	15
32	- 41	Starting patient number	CHAR	10
42	- 51	Ending patient number	CHAR	10

Treatment Pattern Card Type

From:	To:	Parameter	Type	Length
1	- 9	"PATTERN"	CHAR	9
10	- 24	Pattern Code	CHAR	15
18	- 80	Pattern Description	CHAR	63

Patient Card

From:	To:	Parameter	Type	Length
1	- 9	"PATIENT"	CHAR	9
10	- 19	Patient Code	CHAR	10
20	-34	Owning Location	CHAR	15

Converted file structure

- STUDY
 - STRATUM
 - PATTERN
 - * PATIENT
 - * PATIENT
 - * etc.
 - PATTERN
 - * PATIENT
 - PATTERN
 - * PATIENT
 - STRATUM
 - PATTERN
 - * PATIENT
 - PATTERN
 - * PATIENT
 - PATTERN
 - * PATIENT
 - PATTERN
 - * PATIENT

Validating a Loaded Randomization

If required, you can run a report to read the original data file and compare it with the Oracle Clinical database to confirm that the load was correct. From the **Design** menu, select **Randomization**, choose **Randomization Batch Load**, and then **Validate Loaded Randomization**. The Randomization Take-on display of the PSUB submission form appears. Enter the name of the loaded randomization file in the **Description** field.

Randomization Maintenance

You can perform the following tasks by navigating to **Design, Randomization**, and then **Randomization Maintenance**:

- [Replicating a Randomization](#)
- [Mirroring Randomization](#)
- [Maintaining Randomization Statuses](#)
- [Creating or Revoking an Access View](#)
- [Cohort Views](#)
- [Managing Patient Treatment Assignments](#)
- [Linking Patient Treatment Assignments](#)
- [Disclosing Patient Treatment Assignments](#)

Replicating a Randomization

Replicating randomizations is a similar process to replicating clinical studies. The system records information you enter in this module into system tables, including Treatment Pattern Block Definitions, Treatment Assignments, and Randomizations.

You cannot replicate a randomization between studies. You must replicate the study first, then add the randomization to it.

To replicate a study's randomization:

1. From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, and then **Replicate Randomization**. The Replicate a Clinical Study form appears, displaying the Study Selections window.
2. Click the **Replicate Randomization** button.
3. Use the list function to select from the studies with randomizations available for replication.
4. Use the list function to select the study to which to replicate the randomization (the target study).
5. Click the **Replicate Randomization** button.

Mirroring Randomization

Mirror randomization takes an existing randomization and creates a new randomization of replacement treatment assignments starting at a new code. For example:

1 to 10 NORMAL, 100 to 110 REPLACEMENT.

The main use is when a site receives supplies for a group of patients and a corresponding set of replacement patients. Then there is an instruction that if, for example, patient 7 drops out, enroll the next patient into 107. This maintains balance in the study's randomization.

1. From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, and then **Mirror Randomization**.
2. Choose the study to which to add mirrored treatment assignments.
3. Click **Show Randomizations**. The Copy Randomizations window displays the selected study's randomization details.
4. Click **Copy**. A pop-up window appears where you can specify the starting code.

Maintaining Randomization Statuses

This section describes updating the status of the randomization. The information you enter here is a basis for creating clinical study history records.

To update clinical study statuses:

1. From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, and then **Randomization Statuses**.
2. Select the study of interest.
Use the list and query functions, if necessary, to locate the study.
3. Enter the study status code.

To modify the study's randomization access, click the **Phases** button. A window appears titled Amend the Randomization Access Status.

Clicking the **Amend the Study Status** button lets you change the planning status only, and even to do this, you must have planning attributes privileges. You can modify the Randomization Access Status only if you have access to the randomization attributes. The studies display in study code sequence. You can scroll or query to the study of interest. You can only modify a clinical study's status; you cannot insert or delete studies here.

When you click the **Phases** button, the system displays the phases of the live version of the current study. You can work with only the live version of a study while you are in the Amend Study Statuses module. If there is no live version of the current study, the system alerts you with a message.

Creating or Revoking an Access View

An access view enables specified users to see the true randomization within a study without un-blinding the study to everybody. The randomization only appears in the view while the randomization status of the study is set to ACCESS.

This technique controls who can perform an interim analysis, for which study, and when. If necessary, you can edit the view to additionally limit the data it displays. For example, you can limit it to display only the first 50 patient treatment assignments, or only patients who have completed the study.

Note:

If you are working in a UNIX environment, put double quotes around the \$ sign in the account names. For example, OPS"\$"ABCD.

For more information, see:

- [Creating an Access View](#)
- [Revoking an Access View](#)

Creating an Access View

The PSUB form Create Access View creates an access view for a study. It is a standard PSUB submission form. To access this form, from the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, and then **Create Access View**.

Revoking an Access View

The PSUB form Revoke Access Grants revokes the select privilege from a user on an access view for a study. It is a standard PSUB submission form. To access this form, from the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, and then **Revoke an Access View**.

Cohort Views

Oracle Clinical provides the cohort view to show patients assigned to their correct **cohorts** (groups) without revealing which treatment pattern each cohort is for.

For more information, see:

- [Creating a Cohort View](#)
- [Revoking Cohort View Grants](#)

Creating a Cohort View

To access this form, from the **Design** menu, select **Randomization**, then **Randomization Maintenance**, and choose **Create Cohort View**. The view has the same security safeguards as the Access View. You can use the PSUB submission form to create the view.

The system assigns dummy codes to the study's treatment patterns by a hidden algorithm. The algorithm decodes the real treatment pattern into letters—A, B, C, etc., then generates the cohort view.

To access the cohort view, the study must have randomization access status, regardless of your privileges.



Note:

If you are working in a UNIX environment, put double quotes around the \$ sign in the account names. For example, OPS"\$"ABCD.

Revoking Cohort View Grants

Use the Revoke Cohort Grants PSUB submission form to control the grants to access a study's cohort view. From the **Design** menu, select **Randomization**, then **Randomization Maintenance**, and choose **Revoke Cohort View**.

Managing Patient Treatment Assignments

Take the Patients window as a starting point to manage patient positions. From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, select **Patient Treatment Assignments**, and click the **Patients** button. The system lists the patient positions in patient code sequence.

Select the patient position to manage. You can only select a patient position while in this window. You cannot insert, update, or delete a patient position.

The operations you can perform from this window are:

- [Disclosing a Patient Position's Treatment Pattern](#)
- [Changing the Latest Treatment Assignment to a Different Treatment Pattern](#)
- [Replacing a Patient Position with Another Patient at the Same Site](#)
- [Replacing a Patient Position with Another Patient in the Same Study](#)
- [Linking a Patient Position to an Existing Treatment Assignment](#)
- [Creating a New Treatment Assignment](#)
- [Removing a Treatment Assignment](#)

Disclosing a Patient Position's Treatment Pattern

From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, select **Patient Treatment Assignments**, click the **Patients** button, and the **Disclose** button, selecting the appropriate study and patient position, to the Treatment Patterns Assigned to Patient form. The system displays the treatment pattern records in assignment time sequence.

You can also navigate to the Amending Treatment Patterns Assigned to Patient window via the **Change...** button. See the next section for instructions on changing treatment pattern assignments.

Changing the Latest Treatment Assignment to a Different Treatment Pattern

To change the treatment pattern assigned to a patient position without creating a new treatment assignment, you work from the Amending Treatment Patterns Assigned to Patient window. From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, select **Patient Treatment Assignments**, click the **Patients** button, and the **Change** button, selecting the appropriate study and patient position. (This window is also available from the **Change** button in the Treatment Patterns Assigned to Patient window.)

Changing an assignment indicates that the treatment a patient receives is different from the treatment the randomization assigned to the patient. This is the way you can correct mistakes in packaging or labeling.

1. Enter the treatment code to change in the **Treatment Code for this Assignment** field.
2. Complete the **Date** and **Reason for Change** fields.
3. Select the row of the pattern to change the treatment to from the **Code/Pattern** fields.
4. Click the **Change Select Treatment Assignment To This Treatment Pattern** button. You must click this button to save your work.

Replacing a Patient Position with Another Patient at the Same Site

You can replace the current patient position with a replacement patient already assigned to the same site. From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, select **Patient Treatment Assignments**, click the **Patients** button, and then the **Site Replace** button, selecting the appropriate study and patient position. The system displays the Patient Replacement at the Same Site popup window, with the replacement patient controlled by the Patient Replacement Rule Type.

A code on the clinical study version, the Patient Replacement Rule Type, controls how this part of the module operates. Study designers set it in either the Clinical Studies or Easy Study Design modules. The Patient Replacement Rule Type has one of three values:

- **NONE** – No replacement of patient positions allowed.
- **NEXT** – The next sequential replacement patient position not currently enrolled, regardless of treatment.
- **SAME** – The next replacement patient position not currently enrolled that is assigned the same treatment pattern as the one being replaced.

The **Replace Selected Patient With This One** button confirms the patient position replacement and sets the patient enrolled date on the replacement patient to the current date.

Replacing a Patient Position with Another Patient in the Same Study

To replace the current patient position with a replacement patient not linked to any study site, from the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, select **Patient Treatment Assignments**, click the **Patients** button, and then the **Study Replace** button, selecting the appropriate study and patient position. The system displays the Patient Replacement not assigned to any site popup window, with the replacement patient controlled by the Patient Replacement Rule Type.

A code on the clinical study version, the Patient Replacement Rule Type, controls how this part of the module operates. Study designers set it in either the Clinical Studies or Easy Study Design modules. The Patient Replacement Rule Type has one of three values:

- **NONE** – No replacement of patients allowed.
- **NEXT** – The next sequential replacement patient position not currently enrolled, regardless of treatment.
- **SAME** – The next replacement patient position not currently enrolled that is assigned the same treatment pattern as the one being replaced.

Clicking the **Replace Selected Patient With This One** button confirms the patient position replacement and sets the patient enrolled date on the replacement patient to the current date.

Linking a Patient Position to an Existing Treatment Assignment

To link the current patient position to an existing treatment assignment, from the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, select **Patient Treatment Assignments**, click the **Patients** button, and then the **Link Asg**

button, selecting the appropriate study and patient position. A popup window appears, Link Treatment Assignment.

This function is most useful in studies where the Kit codes or Treatment Assignment codes are not the same as the patient codes. The differences could be a result of a second randomization in the study, or of management of limited drug supplies.

Clicking the **Link Selected Patient To This Treatment Assignment** button confirms the patient position linkage and assigns the new treatment code to the patient.

Creating a New Treatment Assignment

You may have patients who receive more than one randomization during a study and, at the end, you have to assign the patients a reporting treatment pattern for analysis. For these, and similar circumstances, Oracle Clinical provides a utility to create a *new* treatment assignment to assign to a patient position.

From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, select **Patient Treatment Assignments**, click the **Patients** button, and then the **Create Assign Treatment to Patient** window provides a new treatment assignment code field and a list acceptable treatment patterns to assign to it. After you enter the fields and select a treatment pattern, you must click the **Create a New Treatment Assignment Using This Treatment Pattern** button to confirm the patient treatment assignment and create a new treatment assignment to allocate the selected treatment pattern to this patient.

Removing a Treatment Assignment

You can break the link between a patient position and a treatment assignment. The treatment assignment is then available for reassignment. From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, select **Patient Treatment Assignments**, click the **Patients** button, and then the **Unlink Assign** button, selecting the appropriate study and patient position. In the Unlink the Treatment Assigned for Patient window, you see a list of all the treatment assignments for the selected patient position. Select the treatment assignment you want to unlink, and click the **Unlink/Remove This Treatment Assignment From The Selected Patient** button, confirming the patient treatment unlink and breaking the link between the patient position and the treatment assignment.

Linking Patient Treatment Assignments

Oracle Clinical provides a separate menu path to the Link Patients and Treatment Assignments window: From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, then select **Link Patient Treatment Assignments**, click the **Patients** button, and then the **Link Assign** button.

This method can be used to allow a blinded user to link patient positions to treatment assignments without knowing what treatment pattern is involved.

Disclosing Patient Treatment Assignments

Oracle Clinical provides a facility for study management members who must disclose the treatment assigned to a patient for emergency medical reasons, but do not have RXC_RAND privileges (see [Randomization Privileges](#)), and do not have access to the Maintain Treatment Assignment, Disclose Assign menu path. The system creates a blind break with each disclosure.

From the **Design** menu, select **Randomization**, choose **Randomization Maintenance**, then select **Disclose Patient Treatment Assignments**, click the **Patients** button, selecting the appropriate study and patient position. The treatment pattern records display in assignment time sequence. Select the patient and click **Disclose**.

If you have the role `RXC_DER`, the system generates a blind break record before displaying the patient's current treatment assignment. Select **Yes** to confirm the disclosure and continue, or **No** to cancel the request and exit. This action always creates a blind break and should be your last choice. Use the **Disclose** option in the Maintain Treatment Assignments form to avoid creating a blind break whenever possible.

Part II

Defining Data Collection Objects

Part II describes creating data collection objects, data entry systems, and systems for testing data. It includes:

- [Questions and Question Groups](#)
- [Global Library Activities](#)
- [DCMs and DCIs](#)
- [Defining DCI Books](#)
- [Annotating CRFs](#)
- [Creating Graphic Layouts](#)
- [Copy Groups](#)
- [Data Extract](#)
- [Lab Ranges](#)
- [Validation and Derivation Procedures](#)
- [Test Mode](#)
- [Integrating with a Clinical Trial Management System](#)

7

Questions and Question Groups

An Investigator collects all study data as responses to clinical study Questions.

These activities take place within Oracle Clinical's Global Library subsystem. For information on other Global Library functions, see [Global Library Activities](#) and [DCMs and DCIs](#)

This chapter contains the following topics:

- [Creating and Maintaining Questions](#)
- [Creating and Maintaining Question Groups](#)
- [Creating and Using DVGs](#)

Creating and Maintaining Questions

Oracle Clinical *Questions* correspond to the Questions on a CRF for a patient visit. You must define an Oracle Clinical Question for each Question on a CRF.

Question definitions are reusable.

In the Oracle Clinical Global Library, you can create *Question Groups* that contain multiple logically related Questions to be collected in the same section of a CRF. You then create DCMs (Data Collection Modules) that contain multiple logically related Question Groups to be collected together. You can create DCIs (Data Collection Instruments) that contain multiple logically related DCMs. In general, a DCI corresponds to a CRF.

The same Question can be included in many Question Groups. The same Question Group can be included in many DCMs, and so on. You can create standard DCMs and DCIs in the Global Library for reuse in multiple studies, and modify them as necessary for particular studies.

There are three windows in the Global Library in which you see the same Question definitions. Which of the three windows you can see depends on your user privileges:

- **Query Questions** (from the **Glib** menu, select **Questions**, then **Qry Questions**). You can query and view all Global Library Questions but cannot make any changes.
- **Provisional Questions** (from the **Glib** menu, select **Questions**, then **Prov Questions**). You can view all Global Library Questions and create and modify Questions with a status of Provisional, but you cannot change a Question's status to Active. You can delete only provisional Questions, and only when they are not included in a Question Group or DCM.
- **Questions** (from the **Glib** menu, select **Questions**, then **Questions**). You can do all Question-related Global Library tasks, including all the activities possible in the Provisional Questions window, plus changing a Question's status to Active or Retired, and changing some attributes of Active Questions.

Global Librarian

To see the Questions window you must have the database role Global Librarian. The Global Librarian is responsible for organizing and cleaning the set of Questions in the Global Library: defining Question categories and medical evaluation qualifiers to apply to each Question,

determining which provisional Questions to promote to Active status for use in production, and which Active Questions to retire.

The Global Librarian is also responsible for defining values in two installation references codelists (under the Admin menu) that affect Questions: Question Category Type Code and Medical Eval Type Code. The system uses values defined in these codelists to populate the lists of values for the Question Categories and Medical Evaluation Type attributes for Questions.

For more information, see:

- [Creating a New Question](#)
- [Assigning Question Categories](#)
- [Extended Text Questions](#)
- [Assigning DVGs to a Question](#)
- [Defining Details for Complex and Question Set Questions](#)
- [Defining Extended Attributes](#)
- [Deleting a Provisional Question](#)

Creating a New Question

The Maintain Questions window has two viewing modes: Multi displays multiple Questions and their attributes, using one row per Question; and Single displays one Question at a time so that you can see all its attributes without scrolling. The Multi window has several buttons the Single window lacks; see [Assigning Question Categories](#), [Defining Details for Complex and Question Set Questions](#), and [Defining Extended Attributes](#).

To create a Question:

1. Complete these fields:
 - **Name:** A unique name within the Question's domain, up to 20 characters.
 - **Domain:** The Question's domain. See [Maintaining Domains](#).
 - **Status:** There are three valid statuses:
 - **Provisional (P).** The default status for new Questions. All attributes of provisional Questions can be modified, but they cannot be assigned to Question Groups or used in any way. Provisional Questions can be deleted.
 - **Active (A).** Active Questions are available for use: they can be assigned to Question Groups and DCMs, and used in test and production studies.
 - **Retired (R).** Retired Questions cannot be assigned to Question Groups, but Questions already in use in Active Question Groups can continue to be used.

 **Note:**

You can change a Question's status from Active to Retired at any time.

You **cannot** change a Question's status from Active to Provisional if the environment is replicated; that is, if there are more than one source locations set to Active in the installation reference codelist SOURCE LOCATION CODE; or if the Question is used by any other object:

- * included in an active Question Group
- * has Lab Ranges defined
- * has Preferred Lab Units in a Preferred Conversion Group
- * has Lab Test Units defined
- * is used as a DCM Qualifying Question
- * is used as a DCM Question

- **Medical Evaluation Type:** A classification for how to evaluate medical responses.
- **Intent:** A unique description of the meaning or intended use of the Question. The text you enter here is included in the list of values for the Questions field in the Question Groups window; it helps identify the Question when you are selecting Questions to include in a Question Group.
- **Question Type:** From the list of values, select a type. Depending on the type you select, the system enables or disables other Question attribute fields for the Question you are defining. The type determines what data type is allowed for the Question, as noted. For example, if you select a Question type of Char, Unit, or Thes Validated, all of which must have a data type of CHAR, the fields for date/time, decimal places and lower bound and upper bound are disabled. Choices include:
 - **Unit** (Data type must be CHAR.) Informational datatype useful for identifying Questions used to capture unit information related to another Question, such as a lab Question.
 - **Lab Test** (Data type must be CHAR or NUMBER.) Defines the Question as collecting lab data and requires collection of the lab name for each received DCM. A Question of type LAB TEST can have a lab range defined in the Lab Subsystem; see [Defining Textbook Ranges](#). If a DCM Question is based on a lab test Question and the DCM Question is used in a Procedure, lab range information for that Question is automatically available for use in the procedure.

If you are using normalized lab data, you must define a Question to display the lab test name with a Question type of Lab Test; see [Using Normalized Lab Data](#) for details.
 - **Date Time** (Data type must be DATE or TIME.) Use for a date or time Question.
 - **Non-Lab** (Data type must be CHAR or NUMBER.) Non-lab Questions with a data type of NUMBER can have Lower Bound and Upper Bound values defined at the DCM Question attribute level as well as in the global library.
 - **Complex.** (Data type must be CHAR, NUMBER, DATE or TIME.) Complex Question values are derived by a macro from other Question responses and appear only in data extract views. They do not have to be defined as a DCM Question. Selecting **Complex** activates the **Details** button. See [Defining Details for Complex and Question Set Questions](#) .

- **Char** (Data type must be CHAR.) Use for character-data type Questions.
- **Thes(aurus) Validation** (Data type must be CHAR.) For use with the Oracle Thesaurus Management System (TMS) only. Thesaurus Validation Questions are not collected at a patient visit, but receive data derived from TMS, such as the dictionary term to which the verbatim term is classified. If a DCM Question is based on a Thes Validated Question and the DCM Question is used in a Procedure, the derived TMS information for that Question is automatically available for use in the Procedure. You must select the **Derived?** check box for the Question and include it as a child Question in a Question Set; see [Defining Details for Complex and Question Set Questions](#) .
- **Question Set** (Data type must be CHAR, NUMBER, DATE or TIME.) For use with the Oracle Thesaurus Management System (TMS) only. Use for Questions you will use to collect TMS verbatim terms—responses collected at a patient visit that will be classified to a standard dictionary term in TMS. For example, you might collect a response to the Question CONCOMMITANT_MEDICATION at a visit and check the response against the dictionary WHO-Drug.

You must include the Question as the parent Question in a Question Set, with a child Question, or variable, for each term you want to derive from TMS back into Oracle Clinical; see [Defining Details for Complex and Question Set Questions](#) and the *Oracle Thesaurus Management System User's Guide* for more information.

Selecting **Question Set** activates the **Details** button.

- **Extended_Text** (Data type must be CHAR.) Extended Text Questions can store a response of 10,000 characters instead of 200 characters, which is the maximum length for all other character-based Question types. See [Extended Text Questions](#) for restrictions and additional information.
- **Data Type:** Data Type is a database classification: Character, Number, Date, and Time. The system generates a univariate discrepancy for responses to this Question that are of the wrong data type. For information about discrepancies, see the chapter "Using the Discrepancy Database" in the *Oracle Clinical Conducting a Study* manual.
- **Date Time Type:** Specifies the expected precision of the response for a Question with Data Type of DATE or TIME.
 - For Questions with a data type of DATE, select **DMY** to include the day, month, and year; **MY** to include only the month and year; or **Y** to include only the year.
 - For Questions with a data type of TIME, select HMS to include hours, minutes, and seconds; or HM to include only hours and minutes.
- **Date Time Fmt (Format):** The Date/Time Format must specify at least as much precision as specified by the Date/Time Type field.
- **Len (Length):** Set the maximum length for responses to the Question. You can make the length longer when you add it to a study DCM, **but never shorter**. The system generates a univariate discrepancy for responses to this Question that are too long. For information about discrepancies, see the *Oracle Clinical Conducting a Study* manual.

 **Note:**

All Extended Text Questions have a predefined length of 10,000 characters. However, the system cannot display the full text of the response in most places in the user interface. **The length you define determines the number of characters displayed** in these places in the user interface, including Oracle Clinical RDC Onsite discrepancy details.

Note that **the system reserves the first 24 characters** for the attachment prefix, so if you define a length of 30, the system displays the prefix plus the first six characters of the response.

- **Dec Plc (Decimal Places):** Specifies the expected maximum number of digits to the right of the decimal point for a response to a NUMBER Question. The number in this field counts toward the total number you specify in the Length field. Enter 0 to specify an integer response. This value is the Question's default Decimal Place value when you copy the Question to Question Groups or DCMs. You can change this field only for a provisional Question, or, for an active Question, only to increase it.
- **SAS Name:** The Question's unique identification, within the Domain, for to access data through the SAS package; up to 8 characters in length, cannot end with a number, and can be comprised only of uppercase letters, numbers, underscores, or hashes. If the Question has multiple occurrences in a single Question Group, this field can be only up to 7 characters. You can change this field only for a provisional Question, or, for an active Question, only to increase it. See [Data Extract](#) for more information.
- **SAS Label:** SAS variable label for data extraction. The Intent field is the default value for this field. You can change this field only for provisional and active Questions.
- **Extract Macro:** Specify the Extract Macro associated with this Question, if any. A list of values is available.
- **Question Set Name:** For Questions of type Question Set, enter the name of the Question Set to which you want to link this parent Question.
- **Safety Question?:** Check if this is a safety-related Question.
- **Derived?:** Select if this Question serves as the recipient of a response generated by a Derivation Procedure. All Derivation Procedures must have a target derived Question, which cannot collect a response in any other way. (See [Defining Procedure Questions](#) for more information.)

Select also for all Questions to be used as TMS Question Set child Questions, because their responses are derived from TMS. You can also define a TMS Question Set parent Question as derived if its value is to be generated by an Oracle Clinical derivation procedure. See the *Oracle Thesaurus Management System User's Guide* for information on using parent Questions this way.

 **Note:**

Oracle supports only the amount of derivation that can be accomplished within a single batch validation. Oracle does not support using an Oracle Clinical derivation procedure to populate the value of another derived parent Question with the value of the original derived parent Question.

- **Upper Case?:** If selected, the system records the response in upper case characters.
- **Sight Verify?:** Select this to display the Pass-1 response during Pass-2 data entry.
- **Default Prompt:** The free form text (up to 60 characters), you enter is the prompt text associated with the response field when the Question is rendered in a data entry form. You can change this value later.
- The **DVG** and **Alpha DVG** fields: You can limit the Question's responses to a static list of values, or Discrete Value Group (DVG). The system generates a univariate discrepancy for responses to this Question that are not included in the DVG. See [Assigning DVGs to a Question](#).

Even if you do not want to assign a DVG to this Question now, you can select **DVG Modifiable?** (and/or **Alpha DVG Modifiable?**) to allow the option to add a DVG to this Question at a later time, even after the Question's status is Active, within a particular Question Group.

- **Lower and Upper Bound:** fields (Optional; applicable only to Questions of data type Number): Enter an appropriate upper and lower limit for the response to this Question. The system generates a univariate discrepancy for responses to this Question that are above the upper bound or below the lower bound.
- **Valid(ation) Failure Type:** During data entry, if a response to the Question fails validation, the discrepancy record the system generates receives this classification.
Choices include: Verify, Normal, Critical, Monitor Alert, Safety, Thesaurus-PT, Thesaurus_SCT.
- **Upper Case?:** If you check this field the characters in the response to a DVG-associated Question are forced to uppercase. You can change this field only for Provisional and Active DCMs.
- **Pass 2 Sight Verify:** If you check this field, the system completes the Second Pass Data Entry response field with the response from the first pass, allowing the response to be visually verified instead of being rekeyed. If this field is unchecked, the second pass field is left blank and requires reentry. You can change this field only for Provisional and Active DCMs.
- **Status Comment:** If you change the Question's status, enter the reason.
- **Retirement Reason:** If you change the Question's status to Retired, select the reason from the list of values.

2. Save.

Assigning Question Categories

A Question cannot have Active status unless you associate it with at least one Question category. A Question category is a broad search classification, such as Demography or General Physical, that you can use to find existing Questions. You define these values in the QUESTION CATEGORY TYPE CODE installation reference codelist; see the Reference Codelists chapter of the *Oracle Clinical Administrator's Guide* for editing instructions.

Question Categories: A Question cannot have Active status unless you associate it with at least one Question category.

1. With the Question selected, click **Question Categories**. If the Medical Evaluation Type of the Question is also a valid Question category, it is automatically assigned to the Question.
2. From the list of values in the Question Category Code field, select as many Question categories as appropriate.
3. Save.

Extended Text Questions

Extended Text Questions support lengthy freeform text responses; for example, a description of a serious adverse event or a patient case history.

In RDC Onsite, an Extended Text Question field is denoted by a yellow note icon. When a user clicks the icon, double-clicks within the field or right-clicks and selects **Show Extended Text** from the right-click menu, a pop-up window appears in which the user can enter or paste up to 10,000 characters.

For more information, see:

- [Extended Text Question Usage Restrictions](#)
- [Extended Text Question Definition Restrictions](#)
- [Attachment Prefix](#)
- [Database Storage](#)

Extended Text Question Usage Restrictions

The following restrictions apply:

- Responses to Extended Text Questions can be entered and updated only in RDC Onsite. You cannot use Oracle Clincial, Mass Changes or Batch Data Load to enter or update values for Extended Text Questions. (However, Batch Data Delete can delete Extended Text Question responses.)
- The system cannot display the full response text in many parts of the user interface. Instead, it displays the value in the RESPONSES table VALUE_TEXT column: the beginning of the response plus information about the version of the attachment that includes the full text; see [Database Storage](#). This truncated response is displayed in Oracle Clinical, Discrepancy Details in RDC Onsite, Audit History in RDC Onsite, and some sections in Patient Data Reports. In the case of the RDC Onsite Audit History pane and Patient Data Reports, the full text is available in a separate window or page.
- Validation Procedures have access only to the truncated response value text stored the the RESPONSES table; see [Database Storage](#).
- Data Extract does not support Extended Text Questions.
- Do not use an Extended Text Question as an indicator Question, as a trigger Question in a flexible study, or as a source Question in conditional branching.
- Special characters entered as part of an extended text response may not be displayed properly in the user interface.

Extended Text Question Definition Restrictions

The system enforces the following restrictions:

- You cannot assign a Discrete Value Group (DVG) to an Extended Text Question.
- You cannot define a default response for an Extended Text Question.
- You cannot define an Extended Text Question as Mandatory.
- You cannot define an Extended Text Question as a derived Question.
- You cannot define a Question Group as repeating if the group contains an Extended Text Question.
- Oracle Clinical automatically sets the Enterable check box to unchecked for an Extended Text Question. You cannot change it. However, although the Question is identified as non-enterable, you can use RDC Onsite to enter and update data for an extended text field. You cannot use Oracle Clinical, Mass Changes, or Batch Data Load to enter or update data in an extended text field.

Attachment Prefix

In the places where the system may not be able to display the full response to an Extended Text Question—Oracle Clinical, Discrepancy Details in RDC Onsite, Audit History in RDC Onsite, and some sections in Patient Data Reports—the system displays a prefix:

ATTACHMENT V<X.Y>:

X is the version number; the number of consecutively different changes saved by a user in RDC Onsite. The first time a user enters and saves a response for a patient, the version number is 1. If a user updates the response and saves again, the version number is 2 unless the actual value has not changed; the system compares the existing value against the value being saved and increments the version number only if there is a difference. The version number can have up to four digits.

Y is the subversion number; the system increments it each time the focus leaves the extended text field but before the user saves the record. It is used for internal auditing.

The system reserves 24 characters, the maximum possible, for the attachment prefix string whether all 24 characters are used or not.

Database Storage

Oracle Clinical stores responses to Extended Text Questions in two database tables; a truncated version in the RESPONSES table and the full text in the RESPONSE_LOBS table. The truncated version is displayed in several places in the user interface where the system cannot display the full text.

RESPONSES

The RESPONSES table VALUE_TEXT column stores the 24-character [Attachment Prefix](#) plus the response entered, up to the defined Question length, truncating any remaining response text. The number of response text characters stored depends on the Question length:

Number of characters in truncated response text = Question length - 24

For example, if you define an Extended Text Question to have a length of 30 and a patient response to the Question begins "A 72-year-old male patient," the system stores (and displays in the user interface) the attachment prefix and the first six characters of the response, for example:

ATTACHMENT V<2.3>: A 72-y

RESPONSE_LOBS

The RESPONSE_LOBS table stores the full text, a maximum of 10,000 total characters, in the VALUE_TEXT_CLOB column. The version number is stored in the VERSION_SN column.

Assigning DVGs to a Question

You can limit the valid responses to a Question to a DVG. For information about DVGs, see [Creating and Using DVGs](#).

Questions used as Qualifying Questions to distinguish multiple occurrences of a DCM in a DCI must have a DVG. See [Configuring Qualifying Questions](#).

The first set of fields, labeled **DVG**, refer to both internal and thesaurus DVGs. The second set of fields refer to alpha DVGs. Otherwise, the functionality of the two sets of fields is the same:

- **(Alpha) DVG Modifiable?** This box controls whether or not any changes can be made to the DVG settings for this Question within the context of a particular Question Group or DCM. If the box is deselected here, it is also deselected whenever the Question is included in a Question Group or DCM, and no changes can be made within those contexts.

If the box is selected here, it will be selected by default for the Question within a Question Group, in the Global Library and in a DCM. Within those contexts, it is then possible to change the DVG or deselect the box. See [Adding DVGs to Active Questions, Question Groups, and DCMs](#).

The Global Librarian can always change this setting here, even after the Question is set to Active. When the box is selected, it is possible to change the DVG assignment.

Note:

The Global Librarian can assign a different subset of the same DVG at any time, even if this box is not deselected.

- **(Alpha) DVG Name** To assign an internal or thesaurus DVG or an alpha DVG to a Question, choose a DVG from the list of values in the **DVG Name** field.
- **(Alpha) DVG Domain** The system enters the name of the DVG's domain.
- **(Alpha) DVG Subset #** Specify the subset of the DVG that you want to assign to this Question. The default is Subset 1. See [Creating a DVG Subset](#).

Note:

The base subset, which contains the complete list of values defined for the DVG, is Subset 0 (zero).

Defining Details for Complex and Question Set Questions

The **Details** button in the Maintain Questions window is enabled only for [Question Set Questions](#) (used only in TMS) and [Complex Questions](#) (used only in Data Extract). In both cases you associate Questions with variables already defined for the current Question.

For more information, see:

- [Question Set Questions](#)
- [Complex Questions](#)

Question Set Questions

In this window, associate the variables defined in the current (parent) Question's Question Set with the derived Questions that will receive the variables' values from TMS. You must first define the Question Set, the parent Question (of Question type Question Set), and each derived Question (of Question type Thes Validated).

The system populates a row for each variable defined for the Question Set.

1. In the Question Name column for each variable, from the list of values select the name of the derived Question to receive the variable value from TMS. The list of values includes only Questions whose Question type is Thes Validated.
2. Save.

Complex Questions

In this window, associate the Extract macro variables with the Questions whose values the macro combines to derive the value of the current (Complex) Question. You must first define the Extract macro, the Complex Question, and the Questions to be used as variables in the macro.

The system populates a row for each variable defined for the current (Complex) Question's Extract macro.

1. In the Question Name column for each variable, from the list of values select the name of the Question whose value will be used as the variable value.
2. Save.

Complex Questions are not DCM Questions; they are created for the purpose of data extract. They are not part of default definition, so they have to be defined in the Global Library. Once a Complex Question is defined, it references other Questions as well as the Extract macro that combines them into a single value.

When you write an Extract macro with more than one variable, you accommodate that Extract macro by creating a Complex Question, which can be broken down into simple Questions, each of which can then be associated with one of the variables.

Defining Extended Attributes

Oracle Clinical provides additional attributes that can be associated with Questions to increase the information available in data extract views. The attributes are columns in

the RESPONSES table. By including them in a Question definition, you make it possible to extract the column value for a Question response.

By default, the system creates one attribute: VALIDATION_STATUS, which contains information on whether any discrepancies are logged against the response, and if so, their current status.

The other attributes available are:

- AUDIT_COMMENT_TEXT. If the record was updated after it was made accessible in the Update mode of data entry, the comment, if any, added by the operator. Value is populated on the version prior to the new version, or, in the case of deletions, on the final version.
- DATA_CHANGE_REASON_TYPE_CODE. Code indicating reason a new version of the response was created. Value is populated on the version prior to the new version, or, in the case of deletions, on the final version.
- DATA_COMMENT_TEXT. Investigator comment, if any, for the response.
- DISCREPANCY_INDICATOR. Box indicating whether a univariate discrepancy, a manual discrepancy, or both exist for the response. Possible values are U, M, and B, respectively.
- DVG_LONG_VALUE. Applicable only to Questions associated with a DVG.
- DVG_NUMBER. Applicable only to Questions associated with a DVG.
- DVG_SHORT_VALUE. Applicable only to Questions associated with a DVG.
- EXCEPTION_VALUE_TEXT. The full value of the response is stored in this column if a discrepancy has been created of the type that indicates that the value is inconsistent with the database use of the DCM Question. In particular:
 - Data type discrepancies store the value here with the value text null.
 - Length discrepancies store the full value here with the value text containing null for numbers and containing the text truncated to the DCM Question length for characters.
- FULL_VALUE_TEXT
- TERM_COL1. Applicable only to Questions associated with a thesaurus DVG.
- TERM_COL2. Applicable only to Questions associated with a thesaurus DVG.
- TERM_COL3. Applicable only to Questions associated with a thesaurus DVG.

Deleting a Provisional Question

You can remove a Question from the Global Library if it meets the following criteria:

- It is a provisional Question. You cannot remove active or retired Questions.
- The Question is not part of a Question Group. If it is, you must first remove the Question from the list of Questions in the Question Group. See [Managing Question Groups](#).
- The Question is not part of a DCM.

From the **Glib** menu, choose **Questions**, then select **(Prov) Questions**. Click the **Delete Record** button, then save your work. If you have not yet saved the Question, there is nothing to delete and you can clear the record by clicking the **Clear Record** button.

Creating and Maintaining Question Groups

Question Groups organize Questions in a study and collect related data by grouping related Questions. For instance, Question Group DEMOG may collect demographic data about patients and contain the Questions: BIRTHDATE, RACE, and SEX.

You can group multiple Question Groups in a single Data Collection Module (DCM) and multiple DCMs in a single Data Collection Instrument (DCI), which generally corresponds to a CRF. You can define a Question Group either in the Global Library or in a particular study DCM, but to promote consistency and reusability, define Question Groups in the Library and then copy them to study DCMs, where you can modify them for the needs of the study. Library Question Groups should be as general and complete as possible so that designers can easily modify them to work in any clinical study.

For example, a Library Question Group ALLERGIES might contain many Questions, each collecting data medically useful in various studies. If a study designer copied and modified ALLERGIES into a cardiovascular study study DCM, the designer might confine it to four Questions that gather a patient's history of allergic reactions to medication (HX_ALLERGIC_SENS, MED_NAME_II, ALLERGIC_REACTION, and FIRST_INCIDENT_YEAR). The designer would designate the other Questions in ALLERGY as "Not Collected."

For more information, see:

- [Creating a New Question Group](#)
- [Assigning Questions to a Question Group](#)
- [Defining Indicator Branching](#)
- [Creating a Question Group by Copying](#)
- [Deleting Provisional Question Groups](#)
- [Managing Question Groups](#)

Creating a New Question Group

Some choices of attributes you make in the library version of Question Groups impact what you are able to do when you include a Question Group in a study DCM and DCM subset. This section addresses some of these attributes:

If you disable COLLECTED at the library level, you cannot enable it in the study DCM Question Group.

If you enable both REQUIRED and COLLECTED at the library level, you must enable both in the study DCM and enable COLLECTED at the subset level.

If you enable REQUIRED but disable COLLECTED at the library level, you do the same at the study level, and you must enable COLLECTED in the DCM subset.

If you disable both REQUIRED and COLLECTED at the library level, then you cannot enable COLLECTED at the study level, but COLLECTED is enabled at the subset level.

If you disable REQUIRED and enable COLLECTED at the library level, but do not enable COLLECTED at the study level, then COLLECTED is also not enabled at the subset level.

If DERIVED is enabled, then ENTERABLE and MANDATORY must be disabled.

If MANDATORY is enabled, then ENTERABLE must be enabled. If ENTERABLE is enabled, then DISPLAYED must be enabled.

The maximum number of Questions in a Question Group is equal to 255, minus the number of fields in the key template. If the system does not present the **Prov Question Groups** menu item to you, see [Assigning Questions to a Question Group](#).

To create a new Question Group:

1. From the **Glib** menu, choose **Question Groups**, then select **Prov Question Groups** or **Question Groups**.
2. Enter values as follows:
 - **Name** – Must be unique within the given domain. Query only to view listings of existing Question Groups for guidance in naming.
 - **Domain** – The system supplies the first domain in your domain list. Use the List function to choose another, if necessary.
 - **Status** – There are three valid statuses:
 - **Provisional (P)**. The default status for new Question Groups. All attributes of provisional Question Groups can be modified, but they cannot be assigned to DCMs or used in any way. Provisional Question Groups can be deleted.
 - **Active (A)**. Active Questions are available for use: they can be assigned to DCMs and used in test and production studies.
 - **Retired (R)**. Retired Question Groups cannot be assigned to DCMs but Question Groups already in use in Active DCMs can continue to be used.

 **Note:**

You can change a Question Group's status from Active to Retired at any time.

You can change a Question Group's status from Active to Provisional only if the Question Group is not used as a DCM Question Group. In addition, you cannot change a Question Groups's status from Active to Provisional in a replicated environment; that is, if there are more than one source locations set to active in the installation reference codelist SOURCE LOCATION CODE.

- **QG Type** – The kind of data gathered by the Question Group. A list of values is available.
- **Expand?** – If checked, you can add more Questions to a saved Question Group.
- **Description** – A description of the Question Group to help other people know if they should use it.
- **Status Comment** – (Optional) If you change the Question Group's status, provide a reason.

- **Retirement Reason Type** – Required when you change the Question Group's Status to R. Select a value from the list. These values are set in the RETIREMENT REASON TYPE CODE installation reference codelist.
3. Save your work, or click the **Exit** button to discard it.
 4. Click the **Question Group Questions** button to populate the Question Group. See [Assigning Questions to a Question Group](#) for details. The other fields in this window are for Question Group maintenance:
 - Use the status comment to explain changing a Question Group's status.
 - Select a retirement reason if you choose to prevent a Question Group from being used in new studies.
 5. Click the **Standards Affiliations** button to assign the standards affiliations to the Question Group. For information on standards affiliations, see the *Oracle Clinical Getting Started* manual.

Assigning Questions to a Question Group

Once you create a new provisional Question Group, assign Questions to it. A Question must meet the following conditions to be assigned to a Question Group:

- The Question must exist in the Global Library. If it does not, create the new Question first. See [Creating a New Question](#).
- The Question Group must be provisional. (To add Questions to Active Question Groups, see [Managing Question Groups](#).)

To assign Questions to a Question Group:

1. From the **Glib** menu, select **Question Groups**, then choose **Prov Question Groups**.
2. Query and select the Question Group where you want to assign Questions.
3. Click the **Question Group Questions** button to display the Question Group Questions window.
4. Enter the name of the Question in the **Question Name** field. The system fills in the Question's current values when you select another field.
5. Fill in all mandatory fields. Depending on your computer's display configuration, you may have to perform several scroll actions to see all fields.

Note:

You can assign the same Question to the Question Group multiple times if needed. If there is only one occurrence of a Question in the Question Group, the system give it an Occurrence Sequence Number (Occ SN) of zero (0). If there are multiple occurrences, the system gives the first one an Occ SN of 1, the second an Occ SN of 2, and so on.

Defining Indicator Branching

Indicator branching allows you to control whether or not Questions in a Question Group are collectible for a particular patient depending on that patient's response to the first Question in the Question Group.

You mark the first Question in the Question Group as the indicator Question and specify the response value that, if entered for a patient, makes the rest of the Questions in the Question Group collectible.

If a patient response to the indicator Question matches the specified indicator value, the remaining Questions in the Question Group are collectible. Otherwise, the remaining Questions in the Question Group remain disabled, and the branch is to the first Question outside of the Question Group.

- In RDC Onsite the Questions are enabled and the data entry form branches to the next Question.
- In Oracle Clinical, there is no change in the appearance of the Questions but the remaining Questions in the Question Group become enterable.

If a patient response to the indicator Question does not match the specified indicator value:

- In RDC Onsite the remaining Questions in the Question Group remain disabled and the data entry form branches to the first Question outside the Question Group.
- In Oracle Clinical, there is no change in the appearance of the Questions but the remaining Questions in the Question Group remain unenterable.

For example, define the first Question, SEX, to collect the patient's gender. If a patient's response to SEX is FEMALE, then in RDC Onsite the remainder of the Questions in the Question Group, which cover pregnancy, become collectible for the patient. If the patient is male, these Questions are not collectible for the patient. The next Questions available in data entry are in the next Question Group in the DCM, if any, or the next DCM, if any.

You define indicator Questions and values in the Question Group Questions window in the Global Library.

Restrictions:

- indicator Questions must be the first Question (according to sequence number) in the Question Group. If you do not define it first, on Save the system repositions the Question as first.
- Only one value can make the rest of the Questions in the group collectible
- An indicator Question cannot also be a conditional branch source Question, and should not be an Extended Text Question.

Creating a Question Group by Copying

To create a new Question Group similar to an existing Question Group, you can copy an existing one. You can then add or change attributes to make the new Question Group fulfill all its purposes. Note that as soon as you make a copy, it exists as a distinct record; changes to the copy do not affect the original Question Group.

1. Navigate to **Glib**, then **Question Groups**, and **Prov Question Groups**.

2. Select **Copy Ques Group** from the **Special** menu to view the Locate Question Groups window.
3. Query to identify an existing Question Group with the attributes you want.
4. Select the Question Group.
5. Click the **OK** button to copy the Question Group.
The system closes the Copy window, redisplaying the Question Groups window. Its first field contains the name of a new record the system generates by prefixing COPYOF to the name of the original Question Group.
6. Change the name of the copied Question Group. If you do not change the name, the system retains the COPYOF prefix.
7. Save.

Deleting Provisional Question Groups

You cannot delete an Active or Retired Question Group. To delete a Provisional Question Group:

1. From the **Glib** menu, select **Question Groups**, then choose **Prov Question Groups**.
2. Query and select the Question Group you want to delete.
3. Click the **Delete Record** button.
4. Save.

Managing Question Groups

Question Group management concerns setting the status of Question Groups, rejecting provisional Question Groups, defining sets of Question Group types, and other administrative tasks.

You can find the Question Group you want to work with by selecting **Locate QG by SA** (Standards Affiliation) from the **Special** menu. (See the *Oracle Clinical Getting Started* for information on standards affiliations.) For example, the standards affiliation type CARDIOVASCULAR may locate the Question Groups ECG_TRACE, ECG_ABNORM, EXER_STAGE, TREADMILL, and STOP_REASON.

For more information, see:

- [Setting Question Group Status](#)
- [Creating Lists of Values of Question Group Types](#)

Setting Question Group Status

Only the Global Librarian, who has access to the full-privilege Question Groups, can set the status of a Question Group to Active or Retired. A Question Group must have one or more Questions before you can make it Active. When you retire a Question Group, the system prompts you to fill in the **Retirement Reason** field.

Creating Lists of Values of Question Group Types

To create lists of values (LOVs) of Question Group types, you (or someone who has the correct privileges) must modify the installation reference codelist DCM DCI QG TYPE CODE. Note that Question Groups have the same type codes as DCMs and DCIs.

Examples of Question Group types are: DEMOGRAPHY, MEDICAL HISTORY, PHYSICAL EXAM, ADVERSE EVENTS, MEDICATION, and COMPLETION. See the Reference Codelist chapter of the *Oracle Clinical Administrator's Guide* for instructions on maintaining reference codelists.

Creating and Using DVGs

A Discrete Value Group or DVG is a list of the values (LOV) that constitute acceptable responses to a Question. You can assign a DVG or DVG subset to a Question to limit the allowable responses to the Question. If a data entry operator enters a different value as a response to the Question, the system creates a univariate discrepancy of type DVG and alerts the data entry operator (see the "Using the discrepancy database" chapter in the *Oracle Clinical Conducting a Study* guide for information on discrepancies).

For example, if a Question is associated with the DVG Yes_No, whose only acceptable values are Yes and No, and a data entry operator mistakenly enters "Male" as a response to the Question, the system alerts the operator, who can then check his or her work. If the operator does not correct the response, the discrepancy remains in the system, and you can use the discrepancy management system to clean the data.

DVGs are reusable; you can assign the same DVG to multiple Questions. For example, a DVG with the values Yes and No is appropriate for many Questions.

For more information, see:

- [Qualifying Questions](#)
- [DVG Subsets](#)
- [Alpha DVGs](#)
- [Thesaurus DVGs](#)
- [Creating a DVG](#)
- [Adding Values to a DVG](#)
- [Creating a DVG Subset](#)
- [Maintaining a DVG](#)
- [Adding DVGs to Active Questions, Question Groups, and DCMs](#)
- [Applying New DVGs to Existing Data](#)
- [Migrating DVG Changes into Layouts](#)

Qualifying Questions

When you create a DCM with a Qualifying Question you must associate a DVG with the DCM Qualifying Question. The system uses the Question response—which must be one of the values in the DVG—to distinguish one received DCM from another collected at the same visit. See [Configuring Qualifying Questions](#).

DVG Subsets

For any DVG, you can create one or more subsets that include only a subset of the values in the base DVG as allowable values. When you assign a DVG to a Question, you must specify the subset number. The system allows only the active values in that subset as responses to that Question. The base DVG contains the complete set of responses that could logically be accepted as responses to a type of Question, and its subsets contain only those responses that are appropriate in certain circumstances.

For example, you could create a DVG called AE_SEV to describe the severity of an adverse event, with the values Mild, Moderate, Severe, and Life-Threatening. You could assign this DVG to a Question to be collected during the active part of the study. However, during the screening phase of the study, you could use a similar Question assigned to a subset of the DVG that did not include the value Life-Threatening.

You can also use subsets to store different layouts for DCI Forms-based layouts used in the Oracle Clinical Remote Data Capture PDF option (see [Example 7-9](#)).

For more information, see:

[Creating a DVG Subset](#)

- [Kinds of DVGs](#)

Kinds of DVGs

There are three kinds of DVGs. It is possible to assign both an alpha DVG and either an internal or thesaurus DVG to a single Question.

- **Internal** (named to distinguish them from thesaurus DVGs, whose values originate in a table external to Oracle Clinical): This is the most commonly used type of DVG. You define the list of values in Oracle Clinical, in the Discrete Value Groups window.
- **Alpha**: Alpha DVGs allow a data entry operator or batch data load to enter information about a response when it is not possible to enter an actual response. For example, an alpha DVG can contain values like Not Applicable, Unknown, or Not Done, which are more informative than a null response value. In addition, you can use alpha DVGs to enter alphanumeric values (such as TRACE) for Questions with data types of Date or Number, if it is not possible to enter a normal response of the correct data type. Alpha DVG values are stored in the **Exception Value Text**: For each alpha DVG value, you can choose whether or not to raise a discrepancy when an alpha DVG value is entered. See [Alpha DVGs](#).
- **Thesaurus**: Thesaurus DVG values are contained in external tables that you create in the Oracle Clinical database for this purpose, or that are part of a Thesaurus Management System instance in the same database. See [Thesaurus DVGs](#) for more information.

Alpha DVGs

Alpha DVGs are designed to allow you to capture information when it is not possible to collect the expected type of response. They are most commonly used for two purposes: entering information on missing data and entering alphabetical characters for numeric lab test results. See [Missing Data](#) and [Alphanumeric Lab Values](#).

Alpha DVG values are stored in the **Exception Value Text** field, not the **Value Text** field used to store real responses.

For each alpha DVG value, you can choose whether or not to raise a discrepancy when an alpha DVG value is entered. See [Create Mand Disc?](#) in [Adding Values to a DVG](#).

You cannot use alpha DVG values for the **Default Value** field in the Question form.

Missing Data

You can use alpha DVGs to allow a data entry operator to enter information about a response when it is not possible to enter an actual response value. For example, an alpha DVG can contain values like Not Applicable, Unknown, or Not Done, indicating that there is no data to enter, and the reason why.

- If the Question is of data type Date or Number, or if the Question has an internal or thesaurus DVG, without an alpha DVG the data entry operator would have to leave the response blank and this information would be lost.
- Even for Questions with a character data type, using an alpha DVG standardizes missing data information, making it easier to follow up.

 **Note:**

You can specify whether or not the system should create a discrepancy whenever a data entry operator enters a particular value in an alpha DVG. This is especially useful for missing data. For example, you could choose to create a discrepancy when the value Not Done is entered, but not when the value Not Applicable is entered. See [Create Mand Disc?](#)

Alphanumeric Lab Values

Alpha DVG values, which may include alphabetical characters, can be entered as responses to Questions with data types of Date or Number. This may be useful for lab test Questions where a numeric response is expected, but the lab returns a result such as TRACE, a standard industry term that does not fit into a numeric data type field. Using alpha DVGs, you can capture the actual lab result and perform statistical analysis on the data.

In this case, only an alpha DVG is associated with the lab test Question. The normal expected response is one of a range of numeric values, not a discrete value from an internal DVG.

For more information, see:

- [Notes on Using Alpha DVGs](#)

Notes on Using Alpha DVGs

Alpha DVGs impact other Oracle Clinical systems as follows:

- In Data Extract, you can include alpha DVG values for missing data in views, either in a separate column or in the same column as the data field (using FULL_VALUE_TEXT). For Questions based on a regular DVG, the sequence number of the alpha data value is available in data extract when the sequence number of the DVG value is specified.

- In Procedures, the special processing of the **Perform Detail Only If Not Null** box (in the Detail Variables window) treats responses as null for the purpose of exclusion from testing, even if they have an alpha DVG value.
- Adding/removing an alpha DVG to/from a Question can trigger the creation or obsolescence of mandatory discrepancies, or cause a different type of discrepancy to become obsolete or to be created. See [Applying New DVGs to Existing Data](#).

Thesaurus DVGs

A thesaurus DVG serves the same general purpose as an internal DVG, but its values are contained in a decode table external to Oracle Clinical. Thesaurus DVGs are useful when one or more Questions have a large but fixed number of acceptable responses. You can use thesaurus DVGs with look-up tables of any kind.

You must create or import the decode table or view, or, if you have Oracle Thesaurus Management System (TMS) installed in the same instance as Oracle Clinical, you can create a view of a TMS dictionary to use as the value source (see [Using Thesaurus DVGs with TMS](#)).

You must create a public synonym for each table and view, and grant access to each table or view to the database role RXCLIN_READ.

Note:

Oracle Clinical has no control over the decode table or view that you use as a value source for thesaurus DVGs. You can add, update, and delete values at any time. You must maintain control over the table and its data.

When you define a thesaurus DVG, you specify the name of the table (or view) and column whose values are acceptable responses to Questions associated with the DVG. You can also specify up to three additional columns in the same table or view whose value the system displays during data entry for each row in the Question's list of values. The system also creates variables for these values for use in Validation and Derivation Procedures (see [DVG Variables](#)).

Restricting Values by Study, Site, and Patient (Context-sensitive Thesaurus DVGs)

If you are using RDC Onsite for data entry, you can restrict the Thesaurus DVG values displayed based on the user's current study, site, and patient context. In the decode table or view, in addition to a column for DVG values you must define columns for the Clinical Study ID, Site ID, and Patient ID. You must also add a WHERE clause to fetch DVG values only in rows whose Clinical Study ID, Site ID, and Patient ID match the current user context passed by RDC.

RDC Onsite populates variables for Clinical Study ID, Site ID, and Patient ID. Three predefined functions in the `rdc_thes_dvg_client` package, `getClinicalStudyId`, `getSiteId`, and `getPatientId`, read these variables. You can call these functions from your WHERE clause.

You can use the conditions listed below in the WHERE clause for the thesaurus DVG view, depending on the level of restriction you want to enforce.

Add one or more of the conditions below as necessary to restrict the list of values. Join multiple conditions with the AND operator.

- To restrict by study, add:

```
table_alias.clinical_study_id = rdc_thes_dvg_client.getClinicalStudyId
```

- To restrict by site, add:

```
table_alias.site_id = rdc_thes_dvg_client.getSiteId
```

 **Note:**

When using a site restriction, you should also specify a study restriction because site IDs can be used across multiple studies.

- To restrict by patient, add:

```
table_alias.patient_position_id = rdc_thes_dvg_client.getPatientId
```

 **Note:**

When using the patient condition, you do not need to include the study and site conditions because patient IDs are unique across study and site.

When restricting by patient, adding the site ID condition could have a negative performance impact if there is an index on the driving table which includes the patient position ID but it does not include the site ID.

Restriction of Thesaurus DVG values based on the current study, site, and/or patient context is only supported in RDC. Only use questions that have a context-sensitive thesaurus DVG in DCMs for which data entry is performed in RDC.

The following means of entering and updating data do not support context-sensitive DVGs:

- Oracle Clinical Data Entry
- Mass Changes
- Batch Data Load
- DCAPI

The above components do not provide the study, site, or patient context required to generate the restricted Thesaurus DVG list of values. In these cases, no values are available where a context-sensitive Thesaurus DVG is used. Therefore, if any of the components listed above are used to enter or update a response for a question that uses context-sensitive thesaurus DVGs, a DVG discrepancy is always raised. However, if the entered response is correct for the study, site, and patient in context, Batch Validation obsoletes the discrepancy once it executes.

In RDC, if no rows match the user's current study, site, and patient, the system displays the message "No values are available. Please enter your response manually." The field then accepts a manually entered response. This response raises a univariate DVG discrepancy. If context-restricted values are available and you manually enter a response that is not included in this list of values, the response also raises a univariate DVG discrepancy.

Sorting Values in Thesaurus DVGs

In RDC Onsite, the list of values in a Thesaurus DVG is sorted by the Decode Term column (THES_DECODE_TERM_COL) by default, but you can specify a different column for sorting.

If the decode table/view has a column with the name SORT_COL and the column is defined in the Thesaurus Values form for the DVG in Oracle Clinical as Label Col 1 (THES_DECODE_LABEL_COL1), Label Col 2 (THES_DECODE_LABEL_COL2), or Label Col 3 (THES_DECODE_LABEL_COL3), the LOV is sorted by that column.

Thesaurus DVG Limitations

Thesaurus DVGs have the following limitations, compared to internal and alpha DVGs:

- You cannot define subsets of thesaurus DVGs.
- Thesaurus DVGs cannot be expanded through the Oracle Clinical user interface. To add values, add rows to the table in the database.
- You cannot use **Enter by Sequence** or **Resequence**.
- The only possible **Display Type** is LOV, regardless of the layout system you are using.
- The system disregards **Max Value Length**. Define the length in the Thesaurus DVG window, using the actual length of the table or view column.

For more information, see:

- [Creating a Thesaurus DVG](#)
- [Using Thesaurus DVGs with TMS](#)

Creating a Thesaurus DVG

To create a Thesaurus DVG, do the following:

1. In the Discrete Value Groups window, define the first fields as usual (see [Creating a DVG](#)).
2. Select **Thesaurus** from the list of values in the field: **Thesaurus** or **Alpha** or **Internal**. The **Thesaurus** button becomes active.

Accept the default values for other fields. Do not enter any values in the lower section of the window. You can update the fields **Status**, **Status Comment**, and **Retirement Reason** when appropriate.

3. Click the **Thesaurus** button. The Thesaurus Values window opens.
4. In the **Source Table/View** field, enter the name of the look-up table or view in the database. The system does not validate that the table or view exists, so you can define it here before creating it in the database.
5. In the **Term Column** field, enter the name of the column whose values you want to use as DVG values. In the **Length** field for this and every other column, enter the length of the actual table or view column.
6. The **Decode Table/View** field defaults to the value you entered for the **Source Table/View** field. Accept the default.

7. The **Term Column** field under the **Decode Table/View** defaults to the value you entered for the Source Table/View Term column. Accept the default.
8. The **Label 1...3 Col** fields are optional. Use them to specify another column in the Decode table whose values you want the system to display in the data entry list of values to help the data entry operator decide which value to choose.

The system also populates variables with these values for use in Validation and Derivation Procedures. The system uses the Short Label as the suffix for the thesaurus DVG variable it creates for the value of this column in a Validation and Derivation Procedure (see [DVG Variables](#)).

The system saves your definition regardless of whether an actual table or view exists that matches the definition you enter here. You can either create (or import) the table first and then define a DVG based on it, or define the DVG first and then create the table. However, if you activate the DVG, associate it with a Question in a study DCM and try to enter data for the Question before creating the actual table or view, the data entry operator receives an error message and cannot use the list of values for the Question.

Using Thesaurus DVGs with TMS

If you have the Oracle Thesaurus Management System or TMS installed in the same instance as Oracle Clinical, you can use the terms contained in a single dictionary level as the values in a thesaurus DVG. All levels of all TMS dictionaries are contained in the table TMS_DICT_CONTENTS (see the *TMS Technical Reference Manual* for further information). To limit the values to a single level of a single dictionary—normally either the verbatim term level or the classification level—you must create a view of the table to use as the source.

You can specify up to three columns for display in the list of values to provide additional information about the term. For example, for each term in the list of values, you could specify that the system display the values in its APPROVED_FLAG column, COMMENT_TEXT column, and the user-defined VALUE_1 column.

Example

Create a view of the classification level of WHO-Drug in the same instance as Oracle Clinical and TMS. Create a thesaurus DVG based on this view and specify the columns APPROVED_FLAG, COMMENT_TEXT, VALUE_1 for display, and assign it to the Question CONCOMMITANT_MEDICATION. During data entry for the Question CONCOMMITANT_MEDICATION, the operator looks for terms in the list of values that match the response on the CRF and have an approval status of **Yes** and appropriate comments and company-specific values.

- If it is clear to the operator which term is the correct one, given the response on the CRF, he or she enters the term as a response in Oracle Clinical. When the response (verbatim term) is processed by TMS, it is autoclassified and no further action is necessary.
- If it is not clear to the operator which term is correct, given the response on the CRF, he or she enters the term as it appears on the CRF. The system creates a univariate discrepancy of type Thesaurus DVG. TMS processes the response (verbatim term), and either autoclassifies it or creates an omission for manual classification. After the verbatim term is classified in TMS, the Oracle Clinical discrepancy is automatically closed during the next batch validation.

Creating a DVG

You can create a new DVG, copy an existing DVG from the Global Library, and modify DVGs.

From the **Glib** menu, select **Discrete Value Grps**, then choose **(Prov) Discrete Value Grps**.

To copy an existing DVG, from the **Special** menu, select **Copy DVG**. Rename and modify as necessary. See descriptions of each field below.

To create an entirely new DVG, you must:

1. Enter the required information the upper half of the Maintain Discrete Value Groups window or the Maintain Provisional Discrete Value Groups window (depending on your access privileges). The fields are described in this section.
2. For internal and alpha DVGs, enter values in the lower half of the window (see [Adding Values to a DVG](#)). For thesaurus DVGs, click the **Thesaurus** button and continue (see [Thesaurus DVGs](#)).
3. You must set the DVG to Active status before you can assign it to any Questions. You must have access to the Maintain (non-provisional) Discrete Value Groups window.

The fields in the upper portion of the Maintain Discrete Value Groups window include:

Example 7-1 Subset Number

The system generates the subset number, using 0 for the base DVG, 1 for the first subset, 2 for the second, and so on.

Example 7-2 DVG Type

Select one DVG type from a list of values. These values are configurable; the Oracle Clinical system administrator can maintain the list in the installation reference codelist DISCRETE VAL GRP TYPE CODE. Use this setting to categorize the DVG for querying purposes. This setting does not trigger any enforced behavior.

Example 7-3 Thesaurus or Internal or Alpha

Select the kind of DVG from the list of values. If you select **Thesaurus**, skip the rest of this section and follow directions under [Thesaurus DVGs](#). For information on different kinds of DVGs, see [Kinds of DVGs](#). See also [Alpha DVGs](#) and [Thesaurus DVGs](#).

Example 7-4 Status

You must have access to the Maintain (non-provisional) Discrete Value Groups window to change the status of a DVG. There are three possible statuses:

- **Provisional.** The default status for newly created DVGs. When a DVG is Provisional you can change any of its attributes and assign it to Provisional Questions.

When you are ready to set the Question and DVG to Active, you must set the DVG to Active before you can set the Question to Active.

You cannot assign a provisional DVG to an active Question.

- **Active.** Change the DVG's status to Active when you are satisfied with its attributes and values and want to assign it to Questions for use in Oracle Clinical. If the DVG's Subsettable, Expandable, and Resequencing attributes are selected, you can change related characteristics (see [Example 7-5](#), [Example 7-6](#), and [Example 7-8](#) below).

 **Note:**

When you change a provisional DVG to **Active**, the system automatically generates Subset 1 (one) at the same time. It is a copy of the base DVG, which is Subset 0. To see Subset 1, query for it.

- **Retired.** Retired DVGs can no longer be assigned to Questions. However, Questions that already use the Retired DVG can continue to use it.

 **Note:**

You can change a DVG's status from Active to Retired at any time.

You **cannot** change a DVG's status from Active to Provisional if the environment is replicated; that is, if there are more than one source locations set to Active in the installation reference codelist SOURCE LOCATION CODE; or if the DVG is used by any other object:

- active subsets of the DVG exist
- the DVG is assigned to a question in an active question group
- the DVG is assigned to an active question
- the DVG is assigned to a DCM Question
- the DVG is assigned to a DCM Qualifying Question

Example 7-5 Subsettable?

If selected, it will be possible to create subsets of this DVG even after the DVG is set to Active.

Example 7-6 Expandable?

If selected, it will be possible to add new values to this DVG even after the DVG is set to Active.

Example 7-7 Enter by Seq?

(Enter by Sequence Number?) If selected, the data entry operator can enter the sequence number only and the system will enter the full value.

 **Note:**

Do not select **Enter by Seq?** for DVGs with numeric values; the system cannot distinguish whether the number entered is intended as a numeric value or as a sequence number.

 **Note:**

If you have both an internal and an alpha DVG assigned to a single Question, the **Enter by Seq?** setting for the internal DVG always overrides that of the alpha DVG:

- If the internal DVG's **Enter by Seq?** box is deselected, then data entry by sequence number is not allowed for either the internal or the alpha DVG values, regardless of the alpha DVG's setting.
- If the internal DVG's **Enter by Seq?** box is selected, then data entry by sequence number is allowed for both the internal or the alpha DVG values, regardless of the alpha DVG's setting.

If entry by sequence number is allowed, and each of the two DVGs has a value with the same sequence number defined, the system uses the setting of the DVGSEQOVERALPHA value in the local reference codelist OCL_DE_CONFIG to determine which value to take. To avoid this situation, follow a convention of giving alpha DVG values high sequence numbers such as 90, 91, and 92.

Example 7-8 Resequence?

If selected, it will be possible to change the order of the values in this DVG even after it is set to Active.

Example 7-9 Layout Options

The following fields determine the layout of the DVG in data entry layouts based on DCI Forms: **Display Type**, **CB Layout**, **CB Label Source**, **Checked Flag Value**. You can give these fields different settings in different subsets of the same DVG.

If you are using the traditional, character-based data entry layout, you can leave these fields blank. Proceed to the next field, [Example 7-15](#).

Example 7-10 Display Type

Choose the type of DVG to display in the default generation. These are the choices:

- **CBG:** (Check Box Group) The list of values appears as a group of boxes. Each box is labeled with one of the values in the DVG. The boxes behave like radio buttons; the Investigator or data entry operator can select only one value. If you choose CBG, you must also choose a CB Layout value and a CB Label Source value, described below.
- **Flag:** Use the Flag display type for DVGs with only two values. When you use a **Flag** field, it is important to know that the database interprets an unvisited **Flag** value as NULL. If you pass over a **Flag** field when you are entering the data in the rendered form, the database assumes the Question is unanswered and marks it NULL instead of either of the discrete values. If you choose a Flag display type, you also choose a Checked Flag Value, described below.
- **LOV:** This option is the default value for this field. It displays the DVG as a list of values. In character-based data entry layouts, this is the only display type used.

Example 7-11 CB Layout

(Check Box Layout; available only for DVGs with a display type of CBG.) This setting determines how the boxes are aligned on the page. You can select either a **Vertical** or **Horizontal** grouping, with the labels to either the **Left** or **Right** of the boxes.

Example 7-12 CB Label Source

(Check Box Label Source; available only for DVGs with a display type of CBG.) Choose either the **Value** or **Long Value** of each value as the default prompt, or label, for each box in the group.

Example 7-13 Checked Flag Value

This parameter is active only for DVGs with a display type of Flag. The default is deselected. You can specify the default label value in this field.

Example 7-14 Settings for All Layout Types**Example 7-15 Upper Case?**

If selected, the system forces the value to uppercase, regardless of how it is entered.

Example 7-16 Max Value Length

The default value is 15 characters. If you need longer values, you can enter a higher number, up to 80 characters.

Example 7-17 Retirement Reason

When you change an Active DVG's status to Retired, the system prompts you for a reason. Select one from a list that includes: DCI/DCM Enhancement, DVG Obsolete, Improved Version, Inaccurate Logic, New Questions Defined, New Standard Established for DCI, New Subset Created, Not Replaced, Poor Design for DCI, Redundant, Relevant Questions Retired, Replaced, Replaced—Enhancements, Replaced—Errors.

Example 7-18 Status Comment

When you change the status of the DVG, you can enter a free form text comment.

Adding Values to a DVG

For internal and alpha DVGs, after you save the basic definition you must enter the values you want to include in the DVG.

For thesaurus DVGs, do not add values here. Thesaurus DVG values are contained in an external table. Click the **Thesaurus** button to continue defining the DVG. See [Thesaurus DVGs](#) for further information.

To assign values to an internal or alpha DVG, move the cursor to the Discrete Values section of the window. For each value, enter the following and save your work:

Seq#

(Sequence Number) Determines the order in which the values appear in the list of values during data entry. See "[Example 7-7](#)".

Value

The actual value (such as Yes, No; Male, Female; Severe, Moderate, Mild) that you are defining as an acceptable response to a Question associated with this DVG (or DVG subset). This value appears in the list of values for the Question during data entry.

Note:

If you use special characters (such as > < [] & -) anywhere in the DVG value, auto fill does not work during data entry for that value. That is, even if auto fill is turned on in a user's data entry configuration preferences, if that user enters ">" as the beginning of a DVG value of ">10" for example, the system does not automatically enter the rest of the value, as it should with auto fill turned on. The user must type the full value or select it from the list of values.

Long Value

A description that also appears in the list of values during data entry.

Active?

(For use in subsets.) All subsets are created containing all the values included in the base DVG at the time they are created. To exclude a value from the subset, clear this box.

Create Mand Disc?

(Create Mandatory Discrepancy?) For alpha DVG values only. If selected, the system creates a univariate discrepancy when this value is entered.

For internal and thesaurus DVGs, a univariate discrepancy is always created when a Question response is not included in the DVG's list of values. If you are using an alpha DVG as a way to collect information when an appropriate response is missing, you may want to create a discrepancy if any alpha DVG value is entered. If your alpha DVG is for a lab test Question, and the alpha DVG value (such as "trace") contains medically valid information, you may prefer not to raise a discrepancy.

Creating a DVG Subset

When you define a new DVG, the system gives it a subset number of zero (0). Subset 0 serves as a base DVG. When you activate the DVG, the system automatically creates Subset 1, which is identical to Subset 0. In Subsets 1 and higher, you can choose to make one or more values **Inactive**, effectively creating a subset of the values in the base DVG. If the DVG **Subsettable?** box is selected, you can create additional subsets at any time.

When you assign a DVG to a Question, you must assign a specific subset of the DVG. The system then allows only the active values in that subset as responses to that Question.

Subset 0 serves as the base DVG on which all others are based, and its functionality is different from that of all other subsets:

- When you add values to a DVG, you must add them first to Subset 0.
- You cannot assign Subset 0 to a Question.

When you create a subset, it contains all the values included in the base DVG (Subset 0) at the time of the subset's creation. If the base DVG (Subset 0)'s **Expandable?** box is selected, you can add values to it. If any other subset's **Expandable?** box is selected, you can add values that were added to the base DVG after the creation of that subset.

You can also use subsets to store different layouts for DCI Forms-based layouts used in the Oracle Clinical Remote Data Capture PDF option (see [Example 7-9](#)).

To create a DVG subset:

1. From the **Glib** menu, select **Discrete Value Grps**, then choose **Discrete Value Grps**. Information about the DVG is displayed in the upper half of the window, and its discrete values, in display sequence, in the lower half.
2. Query and select the base DVG (Subset 0).
3. From the **Special** menu, select **Create DVG Sub**. The system copies the entire base DVG and assigns it the next available subset number.
4. Re-query to display the new DVG subset.
5. Clear the **Active?** box of each value you do not want in the subset.

For an alpha DVG, you can also change the setting of the **Create Mand Discrep.?** box. See [Create Mand Disc?](#) in [Adding Values to a DVG](#).

6. Save.

For more information, see:

- [DVG Subset Example](#)
- [Alpha DVG Subset Examples](#)

DVG Subset Example

Create a DVG called AE_SEV (Severity of Adverse Event {non-cancer}) with the values MILD, MODERATE, SEVERE, and LIFE THREATENING. Assigning this DVG to a Question limits responses to one of those four values. The response INTERMITTENT creates a discrepancy.

Name: AE_SEV Domain: STANDARD Subset Number: 0			
Seq #	Value	Long Value	Active?
1	MILD	Mild	- -
2	MODERATE	Moderate	X -
3	SEVERE	Severe	X -
4	LIFE THREATENING	Life threatening	X -

However, a particular Question might not require all four values, and making inappropriate responses available to data entry could lead to discrepancies. For example, in the Screening phase of a study, the response LIFE THREATENING should not be allowed. For this case, you could create a subset with only three Active values:

Name: AE_SEV **Domain:** STANDARD **Subset Number:** 1

Seq #	Value	Long Value	Active?	
1	MILD	Mild	X	-
2	MODERATE	Moderate	X	-
3	SEVERE	Severe	X	-
4	LIFE THREATENING	Life threatening	-	-

You can assign different subsets to different Questions for use during different phases of the trial.

Alpha DVG Subset Examples

See [Alpha DVGs](#) for information about alpha DVGs. You create alpha DVGs the same way you create internal DVGs.

Date

A base alpha DVG for Date Questions might have three values: Not Applicable, Unknown, and Continuing (see table below). All three values are suitable for End dates. A subset for use in Start dates might include Not Applicable and Unknown but not Continuing.

Numeric

An alpha DVG for numeric fields might have five values: Unknown, Not Applicable, Not Done, and Trace. You could create a subset with only the values Unknown, Not Applicable, and Not Done for use with general Questions of data type Number. For lab Questions of data type Number you could use Subset 1, which includes all five values.

Maintaining a DVG

The Global Librarian or other person with access to the full-privilege window, Maintain Discrete Value Groups, is responsible for modifying and changing the status of DVGs. Anyone with access to Maintain Provisional Discrete Value Groups can modify and delete DVGs with a status of Provisional, and copy Active or Provisional DVGs.

For access to the Maintain Discrete Value Groups window, you must have one of the following roles: RXC_GL_FULL, RXC_GLA, RXC_PA, RXC_QADM, RXC_SDEF, RXC_TA, RXC_SUPER.

For more information, see:

- [Modifying a DVG](#)
- [Changing the Status of a DVG](#)
- [Deleting a DVG](#)

Modifying a DVG

You can modify any attribute of a Provisional DVG.

If you have access to the Maintain Discrete Value Groups window (non-provisional), you can modify attributes of an Active DVG that are defined as modifiable; see [Subsettable?](#), [Expandable?](#), and [Resequence?](#).

You can copy an Active DVG and use it as a basis for a new, Provisional and therefore fully modifiable, DVG.

To copy a DVG, select the DVG and select **Copy DVG** from the **Special** menu in the (Provisional or full-access) Maintain Discrete Value Groups window.

Changing the Status of a DVG

Managing DVGs includes setting the status of DVGs, rejecting provisional DVGs, and defining sets of DVGs. Only the Global Librarian or other person with access to the full-privilege Maintain Discrete Value Groups window can change the status of a DVG to Active or Retired. When you retire a DVG, the system prompts you to fill in the **Retirement Reason** field.

Deleting a DVG

You cannot delete an Active or Retired DVG from the Global Library. To delete a Provisional DVG, from the **Glib** menu, select **Discrete Value Grps**, and choose **(Prov) Discrete Value Grps**. Select the DVG and use the Delete Record and Commit functions.

You cannot delete an Active DVG.

Adding DVGs to Active Questions, Question Groups, and DCMs

If you have the necessary privileges, you can add a DVG to an existing Question in the Global Library; from the **Glib** menu, select **Questions**, then select **Questions** (see [Assigning DVGs to a Question](#)).

After you assign a DVG to a Question in the Global Library, you can assign it to instances of the same Question contained in Question Groups and DCMs, as follows:

- [Question Group Questions](#)
- [DCM Question Group Questions](#)

Question Group Questions

If the **DVG Modifiable** (or **Alpha DVG Modifiable**) box is selected in the definition of the Question in the Maintain Questions window, you can select the same box for the Question in the context of a Question. You can then add, delete or change the DVG (or alpha DVG) assigned to the same Question within a Question Group; from the **Glib** menu, select **Question Groups**, then choose **Question Group Questions**.

You can change from one subset to another of the same DVG regardless of the setting of the box in the parent Global Library Question.

DCM Question Group Questions

If the **DVG Modifiable** (or **Alpha DVG Modifiable**) box is selected in the definition of the Question in the Maintain Questions window, and in the Question Group Question window on which a DCM Question is based, you can select the same box for the Question in the context of a DCM Question Group. You can then add, delete or change the DVG (or alpha DVG)

assigned to the same Question within a DCM; from the **Definitions** menu, select **DCMs**, then choose **DCMs**, select **Question Groups**, and then **Questions**.

You can change from one subset to another of the same DVG regardless of the setting of the box in the parent Glib Question.

When you change the DVG or alpha DVG assignment to a DCM Question, the system automatically cascades the change to all subsets and layouts of the DCM.

When you change a DVG or alpha DVG *subset* assignment to a DCM Question, the system automatically cascades the change to all layouts of the DCM, but not to subsets of the DCM.

Applying New DVGs to Existing Data

When you add DVGs to existing DCM Questions currently in use in a study, the next batch validation checks existing responses to the Questions with newly assigned DVGs, and creates and obsoletes discrepancies as a result, as follows:

- If a response is different from any of the DVG values, the system creates a discrepancy.
- If a response that does correspond to a DVG value previously raised a discrepancy, that discrepancy becomes obsolete.
- If a response was discrepant before the DVG was applied, and is still discrepant afterward, the system closes the original discrepancy and opens a new univariate discrepancy of type DVG.

Migrating DVG Changes into Layouts

If you have updated one or more DVGs or DVG subsets used by Questions in a DCI DCM, you can update an existing DCI Layout to reflect the changes without generating another default layout. Run Update from the DVG Definition utility; from the DCMs window, select the **Graphic Layout** tab, then from the **Special** menu, select **Graphic Layout**, and choose **Update from DVG Definition**. The utility propagates changes to the following properties:

- Active Values
- Display Type Code
- Checked Flag Value

Note:

The utility does not propagate any changes related to CB (Check Box) Layout. To propagate changes to CB properties, you must generate a new default layout.

8

Global Library Activities

The Global Library is an organization-wide central repository for containing standardized data definitions. If you are using replication, you must choose one location to own the Global Library. All other locations replicate it. You can partition the Global Library into domains so to accommodate different definitions of the same data objects. This chapter describes the Oracle Clinical's utilities for managing Global Library definitions.

In this section:

- [Storing Objects in the Global Library](#)
- [Maintaining Domains](#)
- [Maintaining Standards Affiliations](#)
- [Maintaining Global Library Copy Groups](#)

Storing Objects in the Global Library

Global Librarian is an Oracle Clinical database role that you must have to save and activate Questions, Question Groups, DVGs, key templates, and extract macros. The creation of these definitions is described in other chapters. Designers then model these Global Library objects into study-level definitions. They create DCMs, DCIs, procedures, view definitions, view templates, and copy groups. The Global Librarian can then copy any of them into the Global Library and make them available for all studies.

For more information, see:

- [Maintaining Global Library DCMs](#)
- [Maintaining Global Library DCIs](#)
- [Maintaining Global Library Procedures](#)
- [Maintaining Global Library View Definitions](#)

Maintaining Global Library DCMs

You can create and maintain Data Collection Modules in the Global Library, but it's more practical to refine them at the study level and then copy them to the Global Library. The maintenance tasks are the same for either module. See [Defining DCMs](#) for all DCM maintenance topics.

In this section:

- [Copying Study-Level DCMs to the Global Library](#)

Copying Study-Level DCMs to the Global Library

To promote a DCM from the study level to the Global Library, you must copy it. The target DCM must be active at the study level before you can copy it. To copy a DCM, follow these instructions:

1. Navigate to **Glib, DCIs DCMs Procedures**, then **DCMs**.
2. Select **Special**, and then **Copy DCMs**. The system starts the Locate DCMs utility. A dialog box opens where you choose between **Library** and **Study**.
3. Choose the study. The **Study Query** dialog box opens.
4. Locate the study that contains the target DCM.
5. Execute a query and select the target DCM.
6. Click **OK**. A dialog box opens where you can choose to copy all of the DCM's associated subsets or just the base subset. (See [Configuring DCM Subsets](#).)
7. Choose which subsets to copy. The system creates the copy in the Global Library.
8. Click **Save**.

Maintaining Global Library DCIs

You can create and maintain Data Collection Instruments (DCIs) in the Global Library, but it's more practical to refine them at the study level and then promote them. The maintenance tasks are the same for either module. See [Defining DCIs](#) for complete all DCM maintenance topics.

In this section:

- [Copying DCIs into the Global Library](#)

Copying DCIs into the Global Library

To promote a DCI from the study level to the Global Library, you copy it. The DCI must be active at the study level before you can copy it. To copy a DCI, follow these instructions:

1. Navigate to **Glib, DCIs DCMs Procedures**, then **DCIs**.
2. Select **Special**, then choose **Copy Study DCI**. The system starts the Locate DCIs utility.
3. Choose the study containing the target DCI.
4. Locate the target DCI.
5. Click **OK**. A dialog box opens where you choose to copy all the DCI Modules.
6. Choose **Yes** or **No** to copying all of the DCIs DCI Modules. The system creates the copy in the Global Library.
7. Rename the DCI and save your work.

Maintaining Global Library Procedures

You cannot generate or test procedures in the Global Library. Therefore, define Procedures at the study level. See [Validation and Derivation Procedures](#) for more information.

In this section:

- [Copying Procedures to the Global Library](#)

Copying Procedures to the Global Library

To promote a procedure from the study level to the Global Library, you must copy it. The procedure must be active at the study level before you can copy it. To copy a procedure, follow these instructions:

1. Navigate to **Glib, DCIs DCMs Procedures**, then **Validation (Derivation) Procs**.
2. Select **Special**, and then **Copy Proc - Study**. The system starts the Locate Validation utility.
3. Choose the study containing the target procedure, if necessary.
4. Locate the target procedure and click **OK**. The system creates a copy of the procedure.
5. Save.

Maintaining Global Library View Definitions

You create view definitions in the Data Extract Module. See [Data Extract](#) for more information.

In this section:

- [Copying View Definitions to the Global Library](#)

Copying View Definitions to the Global Library

To promote a view definition from the study level to the Global Library, you must copy it:

1. Navigate to **Glib, Data Extract View Builder**, and then **View Definitions**. The Maintain Library View Definitions window opens.
2. Select **Special**, and then **Copy Vies - Study**. The system starts the Locate View Definition utility.
3. Change to the study containing the target view definition, if necessary, by clicking the **Change Study** button.
4. Locate the target definition and click **OK**. The system creates a copy of the view.
5. Rename the view definition and save your work.

Maintaining Domains

Global Library domains provide a way to create any number of unique name spaces within Oracle Clinical. When you define Global Library domains, object names are required to be unique only within the domain, rather than across the entire Global Library. Definitions can be moved from one domain to another as long as the new domain does not contain an object with the same name.

The flexibility that domains allow you is particularly useful in cases where you want separate namespaces to handle both legacy data and data from new studies.

The Global Library can store definitions that you copied into or modified in studies. Each library object belongs to a domain, and together, the library object name and domain form a unique key. Two library definitions with the same name cannot exist in one domain, but different domains can contain library definitions with the same name.

For more information, see:

- [Limiting Access to Definitions Through Domains](#)
- [Domains of Parent and Child Definitions](#)
- [Modifying the Domains of Library Definitions](#)
- [Creating Domains](#)
- [Maintaining Domains](#)
- [Using Domains within the Global Library](#)
- [Domain Selection for Data Objects in Studies](#)
- [Examples of Domain Use](#)

Limiting Access to Definitions Through Domains

Each user and each study has a list of domains from which definitions are accessible to either the user or the study. The user domain list determines what data is accessible to someone working in the Global Library. The study domain list limits the domains in which studies may access Global Library definitions during such operations as copying a DCM into a study. Definitions in a domain that is not on a user or study domain list are not accessible to that user or study, depending upon the task.

Default domain lists are created by the system administrator or Global Librarian for all users and all studies; with the appropriate privileges these lists can be modified for individual studies or users.

Domains of Parent and Child Definitions

Definitions that reference other definitions, such as DCMs, can be seen as parent objects, being made up of child objects, such as DCM Question Groups.

Parent objects in the Global Library are:

- Discrete value groups
- Questions
- Question Groups
- DCMs
- DCIs
- Derivation Procedures
- Validation Procedures
- Where clauses
- Queries

Parent definitions may reference child definitions belonging to different domains. If the domain of any child object is changed, the parent object continues to reference the same child object in its new domain.

When parent definitions are composed of child definitions belonging to different domains, the user domain list needs to be modified to allow access to these domains. (See [Overriding Default Domain Lists](#).) Restrictions on using definitions from different domains are:

- SAS names must be unique for any one Question Group, regardless of their Question's domain.
- A Question Group name must be unique within a DCM.
- Two Questions of the same name cannot appear in the same Question Group or DCM if they are in different domains.

Modifying the Domains of Library Definitions

You can modify the domain of a library object, provided an object with the same name does not exist in the new domain. When a parent object's domain is modified, the child definitions retain their domains.

When a child object in the Library moves to another domain, all instances of its use continue to reference it.

When a library object is copied into a study, the domain of the object at the parent level becomes the name of the study, since object names must be unique within a domain/study combination. The effective domain is therefore the study. The child definitions of the copied object continue to reference their domains as defined in the Global Library, although these domains may be changed if the domain of the child object is changed at the Library level.

Note:

You can modify an object's domain only in the parent form. For example, you can change the domain of a Question in the Maintain Questions form, but not in its instantiation as a child object in the Question Group Questions window of the Maintain Question Groups form.

Creating Domains

The system administrator or Global Librarian creates domains by adding values to the reference codelist OCL_DOMAINS. This codelist is an installation codelist, and only a user with appropriate privileges can modify it. To add or modify a domain in the OCL_DOMAINS codelist, from the **Admin** menu, select **Reference Codelist**, and then **Installation Codelists**. Query and modify the codelist OCL_DOMAINS. The domains are available immediately after you save your work.

Note:

No Oracle Clinical validation prevents the name of a Library domain from coinciding with that of a study; company rules need to be in place to enforce this restriction.

For more information, see:

- [Domains and Prioritization](#)
- [Domains and Object status](#)

Domains and Prioritization

The sequence of domains on the user- and study-level domain lists is significant. The first domain listed has priority over other domains. For example, while creating a Question in the Maintain Questions form, when you navigate from the **Question** field to the **Domain** field, the first domain in the user domain list automatically defaults into the field. This defaulted value can be modified, but will be defaulted into the **Domain** field when any new or copied object is created. If you are creating a number of Library definitions, it is helpful to have the domain in which they should be created appear first on your user domain list.

Similarly, when you are copying definitions by name, such as adding Questions to a Question Group by typing their names, Oracle Clinical copies the first object it finds with a matching name in the order of your domain list.

Domains and Object status

You can set a Retired Global Library object's status back to Active. In addition, Library domains allow for the reuse of the names of retired definitions. For example, a retired Question in one domain can be copied into another domain as an active object. Or, a retired Question can be moved to another domain, for example, OBSOLETE, thus freeing up the Question name in the original domain.

Maintaining Domains

This section describes how to define domain lists for all users or all studies, then how to override a default for a particular user or study.

For more information, see:

- [Creating Default Domain Lists](#)
- [Overriding Default Domain Lists](#)
- [Maintaining User-Level Domain Lists](#)
- [Maintaining Study-Level Domain Lists](#)

Creating Default Domain Lists

The Maintain Default Domain Lists window allows the study administrator or Global Librarian to select the default domain or domains, either for all users or in all studies. If user- or study-level domains are set for particular users or studies, they take precedence over these default lists.

Depending on your role permissions, you can change the default domain listing for a particular user or a particular study.

To define default domain lists for all users or all studies, from the **Admin** menu, select **Glib Admin**, and choose **Default Domains**. Toggle between **User** and **Study** in the **Domain List Sub Type** field to set the user or study default domain lists.

You can add domains to, remove domains from, or resequence the domains on the domain list by using these buttons:

- **All Domains** populates the domain list with all domains in the OCL_DOMAINS reference codelist.

- **Up** moves a selected domain in the list up in prioritization.
- **Down** moves a selected domain down in prioritization.

Any domain can be removed from the list by deleting its record from the Define Default Domain Searchlist window.

The default domain lists take effect immediately for all users and studies without defined domain lists, once you save and exit the form.

Overriding Default Domain Lists

If you have the proper privileges, you can override the default domain lists for a particular user or study by defining user- or study-level domain lists. The domain list determines the accessibility of data in various domains and also prioritizes activities involving domains.

You may need to override the default domain list because you want to query data from domains not on a default list, or to order domains differently from the default.

Maintaining User-Level Domain Lists

To set a user-level domain list, from the **Admin** menu, select **Glib Admin**, and choose **User Domain**. Alternatively you can invoke the **Special** menu and choose **Define Domain Searchlist** from the Global Library forms for maintaining definitions. Whenever you invoke the domain searchlist from the **Special** menu, the system prompts you to save unsaved work.

If you have the necessary privileges (access to the Oracle Clinical menu path Admin, GLIB Admin, Define User Domain Searchlist, if the Oracle-defined menu-role associations have not been modified), you can modify your domain list with these buttons:

Note:

If you do not have the necessary privileges, you can only resequence the domains in the list when you access the Define Domain Searchlist form from the Special menu in other forms such as Maintain Questions.

- **Default** returns the list to the default listing set by the system administrator or Global Librarian.
- **All Domains** populates the domain list with all domains.
- **Up** moves a selected domain in the list up in prioritization.
- **Down** moves a selected domain down in prioritization.
- **Delete Record** removes any domain from the list.

Maintaining Study-Level Domain Lists

To set the study domain list for a particular study, from the **Design** menu, select **Studies**, then choose **Clinical Studies**. Navigate to or query the study whose domain list you want to modify, and choose **Define Domain Searchlist** from the **Special** menu. Modify your domain list, using these buttons:

- **Default** returns the list to the listing set by the system administrator.

- **All Domains** populates the domain list with all domains.
- **Up** moves a selected domain in the list up in prioritization.
- **Down** moves a selected domain down in prioritization.
- **Delete Record** removes any domain from the list.

Using Domains within the Global Library

Working in the Global Library, you can view and access only definitions belonging to domains defined in your user domain list, which may be based on a default user domain list or on a customized user-level domain list. As well as limiting the domains where you can access data, the domain list sets up certain prioritization.



Note:

A Question name and/or SAS name must be unique for any one Question Group. Also, two Questions with the same name, but in different domains, cannot appear in the same Question Group.

For more information, see:

- [Querying a Library Object](#)
- [Creating a Library Object](#)
- [Modifying the Domain of a Library Object](#)
- [Copying an Object in the Global Library](#)
- [Adding and Locating Objects](#)

Querying a Library Object

Any queries you make in the Global Library are restricted to the domains on your user domain list; they bring up all objects that belong to domains in that list. When you query parent objects, only objects belonging to domains on your user domain list are returned. However, when you select a particular object, you can view all child objects of the parent, regardless of the domain of the child object. For example, if you have access only to objects in domain STANDARD and query on DCMs, all DCMs with domain STANDARD are returned. If you click the **DCM Question Groups** button, you see all the DCM Question Groups associated with that DCM, even if those Question Groups belong to domains not on your user domain list.

Creating a Library Object

When you create a Global Library object, the domain of the created object must be on your user domain list. The first domain listed on the user domain list automatically defaults into the **Domain** field, so if you are working primarily in one domain, it is helpful to have that domain appear first on your user domain list.

You can change the defaulted domain to another domain, provided the combination of object name and domain does not already exist. A list of values (LOV) is available to select a domain.

Modifying the Domain of a Library Object

The domain of any library object can be changed, regardless of the status of the object, provided the new combination of object name and domain is unique.

Copying an Object in the Global Library

When an object is copied, either from a different domain in the Global Library or from a study, the newly copied object is created in the first domain on the user domain list. After copying the new object, you can rename it to another domain, provided the object name/domain combination is unique.

Certain copy functions allow you to copy an object from within the Global Library or from a study. If you are copying from the Global Library, you are limited to objects belonging to domains on your user domain list. If you choose to copy from a study, you are under no such restriction.

Adding and Locating Objects

Adding and locating actions that can be invoked from the **Special** menu are bounded by the user domain list. Possible actions include **Add Question**, **Add Ques Grp** (which are copy functions on child objects), and **Locate DVG options**.

Domain Selection for Data Objects in Studies

The domain of all parent-level objects in a study is the study name, since all object names must be unique within the study. The default or study-level domain list limits access to library objects from study data definition forms. Only library objects belonging to the domains on this list are accessible to the study during copy functions.



Note:

A Question name or SAS name must be unique for any one Question Group. Also, two Questions of the same name cannot appear in the same Question Group if they are in different domains.

For more information, see:

- [Domain Rules When Creating or Copying Objects](#)
- [Adding and Locating Objects](#)

Domain Rules When Creating or Copying Objects

When you create an object in a study, the domain is the study name, and it cannot be modified.

When you copy an object from the Global Library to a study, you are limited to accessing the domains on the domain study list. Restrictions on copying from another study are determined through security and not by domain accessibility.

Objects copied from the Global Library have the study name as the domain at the parent level, and child objects retain their Global Library domain names. If the domain of a child object is changed in the Global Library, this change is reflected in any studies in which the object appears.

Adding and Locating Objects

Adding and locating actions that can be invoked from the **Special** menu are bounded by the study domain list. Possible actions include the **Add Question**, **Add Ques Grp** (which are simply copy functions on child objects), and **Locate DVG** options.

Examples of Domain Use

Here are some examples of the ways that you can use domains and manual procedures:

- [Support Conversion and Current Standards](#)
- [Support Project-Specific Standards Exceptions and Development](#)
- [Remove Obsolete Objects to Allow Reuse of Names](#)
- [Create a Special Domain for Testing](#)

Support Conversion and Current Standards

Create a new domain, CONVERSION, in addition to the domain STANDARD. Create library definitions for converted studies that cannot be modified to comply with new standards. Enforce the standard operating procedure that converted studies can only use objects containing objects from the CONVERSION domain, while new studies may reference only STANDARD definitions. You must manually enforce that unique parent objects are created in the CONVERSION domain if any component objects are from the CONVERSION domain.

Support Project-Specific Standards Exceptions and Development

In addition to the STANDARD domain, create a domain for each new project. Define in each new domain any objects specific to the project. If objects defined at the project level later become candidates for general use, they can be moved to the STANDARD domain. For any study, the project name domain would be the first on the study domain list, with STANDARD second.

Manually enforce the following constraints:

- A given project can copy objects only from its own project domain and from the STANDARD domain.
- Candidate objects—first defined at the project level so that they might later become standards—initially have unique names.
- No STANDARD objects are created with the names of candidate objects while the decision to elevate them from "candidate" to "standard" is pending.

Remove Obsolete Objects to Allow Reuse of Names

In addition to the STANDARD domain, create domains OBSOLETE, OBSOLETE1, OBSOLETE2, etc. When it is determined that a standard object must be retired from

use, and that also a new object of the same name must take its place, change the domain of the retired object to OBSOLETE. If, by some rare circumstance, the successor object must itself be retired and moved out of the way, rename its domain to OBSOLETE1.

You cannot move the successor object to OBSOLETE, since OBSOLETE already contains the original object with that name. Manually enforce the constraint that you cannot copy objects from an OBSOLETE domain to a new study or to be part of a new library object.

Create a Special Domain for Testing

In addition to the STANDARD domain, create the domain INSTALL_TEST to hold objects for testing purposes in a new database installation. Manually enforce the constraint that two domains should not reference each other, and that only STANDARD objects be used for production work.

Maintaining Standards Affiliations

A goal of Global Library management is to maximize conformity among data collecting components common to all clinical studies, or to groups of similar clinical studies, so that you can make direct comparisons. Standards affiliations standardize data collection in new clinical studies by helping classify, locate, and track the use of the Global Library objects.

For example, if a standard Question Group is affiliated with the affiliations type SYSTEM, you can suggest that all studies in the Oracle Clinical system reference that Question Group. Such a reference, constituting a system-wide standard, might be a requirement of uniform collection of data about safety, which would require that all studies use a standard safety Question Group called SAFETY_QUES. The Global Librarian sets these requirements.

Standards are defined by affiliations types, which are maintained through the form Maintain Standards Affiliations (from the **Glib** menu, select **Standards Affiliations**).

Affiliations is the means by which Oracle Clinical reaches these ends.

For more information, see:

- [Using Affiliations](#)
- [Maintaining Affiliations](#)

Using Affiliations

This section describes the workflow for affiliations in the Global Library and the procedure for locating library Question Groups, DCIs, DCMs, and both Derivation and Validation Procedure definitions with affiliations.

These are the tasks that you can accomplish by using standards affiliations:

1. You can locate library objects such as Active Questions, Question Groups and DCMs, Global Library DCIs, DCMs, and Validation and Derivation Procedures. For this task, first choose the affiliations types pertinent to the study. Then use these to locate the library objects for the new clinical study. Related reports help you identify library objects left out of the new clinical study and to monitor other uses of affiliations.
2. Affiliating library objects with affiliation types, so that the objects can be located later. This activity is performed by the Global Librarian.
3. The Global Librarian can create new affiliations types.

For more information, see:

- [Locating Library Objects](#)
- [Mapping Library Objects to Affiliation Types](#)

Locating Library Objects

You can use the following method to locate library DCMs, DCIs, and Validation and Derivation Procedures from the corresponding study definition forms.

To locate Question Groups with affiliations, select **Locate QG by SA** from the **Special** menu to open the **Query Library Objects by SAS** dialog box.

1. Move the insertion point to the **Standards Affil. Type** field. Enter the affiliation type you want, or select a value from the list of values. If you select one of the listed values, the **Standards Affiliation Name** field is filled automatically.
2. Select the **Required?** field if you want only Question Groups that are REQUIRED; or, to locate all affiliated Question Groups, leave that field blank.
3. Select the **OK** button or **Save** to initiate the query. The Question Groups form reappears, displaying only those Question Groups that have been associated with the affiliations you entered.

Mapping Library Objects to Affiliation Types

You can map library Question Groups, DCMs, DCIs, and procedures to affiliation types.

For example, if you want to associate a Question Group with the indication of angina *and* a particular compound, enter the same set value for the two records. If the affiliation is for the indication of angina *or* a particular compound, use different set numbers to indicate that they are independent affiliations. Select the **Required?** field if this is a required affiliation, or leave it blank if the affiliation is simply recommended.

To map library objects to affiliation types, follow these instructions:

1. Navigate to one of these Global Library maintenance windows, or their Provisional equivalents:
 - From the **Glib** menu, select **Question Groups**, then choose **Question Groups**.
 - From the **Glib** menu, select **DCIs DCMs Procedures**, then choose **DCIs**.
 - From the **Glib** menu, select **DCIs DCMs Procedures**, then choose **DCMs**.
 - From the **Glib** menu, select **DCIs DCMs Procedures**, then choose **Validation Procedures**.
 - From the **Glib** menu, select **DCIs DCMs Procedures**, then choose **Derivation Procedures**.
2. Locate or create the object you want to map to an affiliation.
3. Select the **Standards Affiliations** button. The **Standards Affiliations Set** dialog box opens.
4. Create a new affiliation by entering values in the **Standards Affiliations Type** and **Standards Affiliations Name** fields. Use the **Set** field to group two SA records in the same affiliation.

5. Click **Save** to save the affiliation, then click **Back** to return to the Question Groups window.

Maintaining Affiliations

Maintaining affiliations consists of determining whether affiliations are active and whether Question Groups affiliated with specific standards are required or recommended. New standards affiliations areas are created in the Maintain Standards Affiliations form; from the **Glib** menu, select **Standards Affiliations**. The Maintain Standards Affiliations form appears, with the cursor in the **Standards Affiliations Type** field.

1. Enter a type or select a value from the list of values. (You cannot create a new standards affiliations type from this form; the types are part of a reference codelist and populated at the installation level.)
2. The cursor goes to the **Affiliation Name** field, and existing values are displayed. Enter a new affiliations name by moving the cursor to the first blank field and typing in the name.
3. Click **Save** to save the new entries. The entries are immediately available for association.

Maintaining Global Library Copy Groups

Copy groups simplify copying multiple data definitions between studies or between studies and the Global Library. They allow you to perform one copy operation for a group of related definitions instead of one copy operation for each definition, without violating any data model rules in the process. You can perform copy group operations between studies, or to or from the Global Library.

This section describes Global Library copy group activities. For study-level activities, see [Copy Groups](#)

In the Global Library, you classify copy groups by domains and standards affiliations. For example, a Standard Safety copy group might consist of a standard Adverse Event DCM, a standard Blood Chemistry DCM, and several other similar DCMs, together with all the standard DCIs, Procedures, and view definitions associated with those DCMs. Copy groups have a status of **Provisional**, **Active**, or **Retired**.

For more information, see:

- [Copy Group Restrictions](#)
- [Creating a Global Library Copy Group](#)
- [Performing Copy Group Operations as Batch Jobs](#)
- [Performing Ad Hoc Copies](#)

Copy Group Restrictions

These Oracle Clinical rules control which definitions you can add to a copy group:

- Definitions allowed in a copy group are DCMs, DCIs, Procedures and view definitions.
- All definitions to be grouped must be related to the included DCMs. Conversely, DCMs referenced by another object in the group cannot be removed.
- For DCMs, you specify whether to copy just the base subset or all subsets.

- Definitions can come from multiple domains, but the object name and object type must be unique within the group. Specifically, a single object may be included only once in a given copy group, and you cannot include definitions from different domains with the same name.
- Provisional and active definitions can be added to a provisional copy group, but all definitions must be active before a copy group can be made active.
- Definitions are included in copy groups by reference. Changes to the underlying definitions are reflected in the results of subsequent copy operations.

For more information, see:

- [Validation Rules](#)
- [DCMs](#)
- [DCIs](#)
- [Procedures](#)
- [View Definitions](#)
- [Copy Rules](#)
- [DCM Copy Rules](#)
- [DCI Copy Rules](#)
- [Procedures Copy Rules](#)
- [View Definitions Copy Rules](#)

Validation Rules

Oracle Clinical does not modify any object names or attributes to prevent failure of a copy group action. Error messages are recorded in a log file. The system follows these rules to validate individual definitions:

DCMs

If copying all subsets is specified, all subsets in the source DCM must be active or the copy procedure fails.

If a DCM with the same name exists in the target domain, Oracle Clinical allows the copy to proceed only if that DCM is functionally equivalent to the DCM being copied. To be functionally equivalent, the two DCMs must meet these criteria:

- same Qualifying Question, if any
- same named DCM Question Groups with the same repeat value
- same name, short name, and subset names
- same Question/occurrence numbers in DCM Questions for each group with the same settings for `Collected_in_study_flag`, `Derived_flag`, and `DVG_ID`

DCIs

The DCI name and short name must not be present in the target domain.

Procedures

The Procedures name must not be present in the target domain.

View Definitions

Name must not be present in the target domain. If the source domain is a study, then there must not be a view template of the same name in the target domain.

Copy Rules

These rules are in effect during a copy activity:

- All active definitions are copied; definitions that have been retired are not copied.
- Definitions are copied in this order: DCMs, DCIs, procedures, view definitions.

The rules for copying specific definitions are:

DCM Copy Rules

All of the DCMs must be in **Active** status. Depending on the setting of the **Base Subset Only?** field, the system either just copies Subset 1 Layout 1 of a DCM, or all subsets and layouts. The system copies all child definitions.

DCI Copy Rules

The system copies all DCI modules. The system treats clinical planned events as they currently exist.

Procedures Copy Rules

The system copies active versions. The system ensures that the copied procedure references the equivalent DCM in the target.

View Definitions Copy Rules

The system copies all view definitions and all of their components, with these limitations:

- If the view definition references a library view template, the system only copies the view definition; the template remains a reference to the library template.
- If the source of the view definition is a study, the system also copies the view template.
- The mapping of the view definition to the DCM in the copy group ensures that the copied view definition references the equivalent DCM and DCM Questions in the target.

Creating a Global Library Copy Group

You create a copy group by assigning the group a name, a domain, and a description; selecting one or more DCMs; and, finally, picking the related definitions and adding them to the group.

You can use these groups immediately to copy definitions to the library or study, and then dispose them. To create a permanent library or study object, make the copy group active before exiting the form.

The following steps describe creating a copy group in the Global Library. You can also create a study-level copy group from the **Definition** menu.

1. From the **Glib** menu, select **Copy Groups**, then choose **Copy Groups**. The Maintain Library Copy Groups window opens.
2. Enter values for these mandatory fields: **Copy Group Name**, **Domain**, and **Description**.
3. Save the copy group name.
4. Click the **Copy Group Details** button.
5. Click the **DCMs** button. Since all other copy group elements must relate to DCMs, this is the only active button at this stage. A list of candidate DCMs opens.
6. Select one or more DCMs.
7. Save the DCM choices.
8. Click each of the other buttons to include any other types of definitions you want to include in the copy group. As you complete adding each type of object, click **OK** to return to the Details form.
9. Perform one of these operations:
 - To make a reusable copy group, click **Save**. You can then immediately perform a copy operation, or you can back out of the window.
 - Click **Exit** to store the copy group or choose **Copy to Study** from the **Special** menu to immediately use it.

Performing Copy Group Operations as Batch Jobs

Copy groups are utilized by submitting a batch job. When you submit the batch job, the expectation is that you are copying a copy group from another study or the Global Library to your current location.

You have these options:

- [Copying from the Global Library to a Study](#)
- [Copying from Study to Study](#)
- [Copying from Your Current Study to the Global Library](#)

Copying from the Global Library to a Study

To perform a copy group operation from the Global Library to your current study, navigate to **Definition**, then **Group Copy from Library to Study**. Then choose a domain and a Global Library copy group.

Copying from Study to Study

To copy a copy group from another study into your current study, navigate to **Definition**, then **Group Copy from Study to Study**. Then choose the other study and the target copy group.

Copying from Your Current Study to the Global Library

To perform a copy group operation from a study to the Global Library, navigate to **Glib, Copy Groups**, and then **Group Copy from Study to Library**. Then choose the target copy group, its study, and its domain.

Performing Ad Hoc Copies

Oracle Clinical lets you create a copy group to perform an ad hoc copy of multiple database definitions. This allows you to easily copy multiple definitions without the overhead of maintaining the new copy group in the database. If you don't intend to activate the copy group: a provisional copy group is sufficient.

9

DCMs and DCIs

Data Collection Modules (DCMs), Data Collection Instruments (DCIs), and DCI Books are data collection containers. A DCM contains Question Groups, a DCI contains DCMs, and a DCI Book contains DCIs. Each can be reused in the next larger container either in the same study or a different study.

For more information, see:

- [Defining DCMs](#)
- [Creating Character-Based Layouts](#)
- [Defining DCIs](#)

Defining DCMs

A Data Collection Module (DCM) contains Question Groups, which you define in the Global Library. In either case, all Questions contained in a DCM must be completed in a single visit.

You define a new DCM for a particular study. See [Creating a New DCM](#) for more information. You can then copy it either to the Global Library or to a different study. See [Copying a DCM](#) for more information. A DCM may correspond to a whole CRF or to a section of a CRF.

For example, a DCM might contain a DCM Question Group of physical examination and demography questions for a patient's first visit. Another kind of DCM might contain a set of related laboratory measurements from a later clinical visit. The lab data could be electronically gathered and destined for batch loading into Oracle Clinical using the lab DCM.

For more information, see:

- [Generating Layouts from DCMs](#)
- [Creating a New DCM](#)
- [Copying a DCM](#)
- [Adding Question Groups to a DCM](#)
- [Configuring Question Groups in a DCM](#)
- [Adding or Modifying DCM Question](#)
- [Configuring DCM Subsets](#)
- [Configuring Qualifying Questions](#)
- [Using Repeating Question Groups](#)
- [Defining Conditional Branching](#)
- [Using Normalized Lab Data](#)
- [Defining Date Formats](#)
- [Scheduling DCMs](#)

Generating Layouts from DCMs

You can generate a character-based layout from a DCM that becomes the form used for data entry in Oracle Clinical. Alternatively, you can associate the DCM with a Data Collection Instrument (DCI) and generate a graphic layout that becomes the HTML form used for data entry in RDC Onsite and the PDF form used to generate the Patient Data Report.

The order in which you add Question Groups determines the default layout of the data entry form. Although you can edit the default layout, you can reduce layout work if you are aware of how the DCM's definition affects the layout, especially since regenerating the layout overrides your edits. In the layout:

- DCM Questions become fields
- DCM Question Groups become blocks

For more information, see:

- [Using Variations of the Same DCM](#)
- [Using Provisional and Active DCMs](#)

Using Variations of the Same DCM

Oracle Clinical does not permit you to create multiple identical study DCMs in one clinical study because identical DCMs would be indistinguishable during data analysis. However, it is likely that your study design calls for you to collect certain kinds of data repeatedly throughout the study. You handle this requirement with base DCMs.

You can define a *base DCM* that includes all related Question Groups and then create *DCM subsets* that collect only a portion of the base DCM Questions.

You can also differentiate occurrences of the same DCM by defining Qualifying Questions; for example, Questions about eyes might begin with a Qualifying Question asking if it is the right or left eye.

Using Provisional and Active DCMs

You can modify DCMs with a status of Provisional without constraints. You can use provisional DCMs in test mode to enter test data and identify required modifications to the DCM definition. When you are ready to use the DCM to collect production data, set its status to Active. If you have defined the DCM in the Global Library, copy it to your study.

Creating a New DCM

To create a new DCM:

1. From the **Definition** menu, select **DCMs**, and choose **DCMs** for the Data Collection Modules window of the Maintain Study DCMs form.

If you have not already selected a study during this Oracle Clinical session, the system prompts you to choose a study.

2. Insert a new record and enter a name for the new DCM. The name can be up to 16 characters, but limiting it to eight can make it easier to identify after it is added to a DCI.
3. The **Name** field is the subset name. This value becomes the default subset name when you add it to a DCI. The default subset starting number is one. It increments each time you make a new subset.
4. Give the DCM a short name of up to four characters.

 **Tip:**

Avoid using "CON" as a DCM short name. In a Data Extract SAS view, the SAS view creation file con.sas fails because "CON" is a reserved device name.

5. Choose a **Type** from the list of values. The values are set in the DCM DCI QG TYPE CODE installation reference codelist.
6. Add a description. It can be specific to the subset.

 **Tip:**

Avoid including the colon (:) character in the description field. They considerably reduce performance when you edit the graphic layout. A layout that would take a few seconds to open can take several minutes if there is a colon in the description.

7. Click **Save**.
8. Add Question Groups; see [Adding Question Groups to a DCM](#).

Copying a DCM

The most common procedure for creating a DCM is to copy an existing DCM and modifying it to suit the needs of the study.

To copy a DCM:

1. From the **Definition** menu, select **DCMs**, then choose **DCMs**.
2. From the **Special** menu, select **Copy DCM**. A dialog box appears.
3. Select one of the two buttons in the dialog box: **Library** or **Study**. If you select **Study** the system prompts you to select the study from which you want to copy a DCM.

The Data Collection Modules window opens in Query mode. You can query DCMs by entering the DCM name, qualifying value, or another attribute value and executing the query. You can perform multiple queries.

 **Note:**

The Study Domain Search List controls what you see when you search the Library from the study level.

4. Select the DCM you want to copy and click **OK**.

The system creates a copy of the selected DCM. If a DCM of this name already exists in the target study, the system copies the DCM with the name `COPYOF DCM`. The system prompts you to change the name before saving because you cannot save a study DCM with a name beginning with `COPYOF`. Furthermore, you may not copy another DCM until you have removed the `COPYOF` name and the short name beginning `C-` from the one you just copied.

Adding Question Groups to a DCM

After saving a DCM to your study, populate it with DCM Question Groups. From the Maintain Study DCMs window, select your DCM and click the **DCM Question Groups** button. You cannot create a new Question Group from any of the forms in the Definition subsystem. You can create new Questions and Question Groups only in the Global Library.

You can only add a Question Group that has a status of Active. When you add a Question Group to a DCM, it becomes local to that DCM. Some of its library attributes are copied into the local copy, some are not, and some new ones are available.

1. From the **Definition** menu, select **DCMs**, then **DCMs**. The Data Collection Modules window opens.
2. Click the **DCM Question Groups** button. The DCM Question Groups window opens.
3. Either select a Question Group from the list of values in the DCM Question Group Name field or, from the **Special** menu, select **Add Ques Group**. You can then query for Question Groups by various attributes.
4. Select the DCM Question Group to copy.
5. Save. The system returns you to the DCM Question Groups window and displays a message that the copy either succeeded or failed.

You can modify Question Groups for use in this DCM; see [Configuring Question Groups in a DCM](#)

Configuring Question Groups in a DCM

In the context of a DCM, Question Groups are called DCM Question Groups. Some of their Global Library attributes cannot be changed and others can. They also have additional attributes that apply only in the context of the current DCM. DCM Question Group attributes include:

- **DCM Question Group Name** You can give this local copy a different name.
- **Library Question Group Name** The system copies this name from the library version. You cannot change it.
- **Question Group Domain** The system copies this name from the library version. You cannot change it.
- **Short Name** An abbreviated name used for naming data extract views.
- **Collect in Subset** If this field is checked, responses to the Questions in the DCM Question Group of this DCM subset are collected. You can change the Collect in Subset? field only for Provisional and Active DCMs. You can exclude a

Question group from this DCM subset without removing it from its record by deselecting this box.

 **Tip:**

Avoid unchecking this field once data has been collected for the DCM. If you must uncheck the field, you must immediately regenerate the form.

Otherwise, in HTML data entry (RDC Onsite), the form opens and the field is still displayed. However, the field is no longer updateable. Discrepancies, Investigator comments and audit history for the field are still displayed. Discrepancies and Investigator comments appear to be updateable, but they are not.

Once the form is regenerated, the field does not appear, nor do associated discrepancies, Investigator comments and audit history. However, note that the data remains in the database and cannot be deleted.

- **Display Seq#:** Sets the default cursor tab sequence among all of the Question Groups you add to the DCM. As a data entry operator enters data and tabs, this number determines the default tab sequence in which the operator's insertion point moves through the form. You can change this value later when you lay out the DCM, but this default sequence remains with the DCM if you use it in another study.
- **Repeating Group?** Select this box if this Question Group needs to be collected multiple times in this DCM; for example, if you are loading normalized lab data using this Question Group; see [Using Repeating Question Groups](#) and [Using Normalized Lab Data](#).
- **Max Reps Expected** Graphic layouts, which are used in RDC Onsite, always display the number of repeats specified here. Character-based layouts, which are used in Oracle Clinical, allow dynamically adding scrollable displays of repeating Question Groups up to this value. The default value is one. You can change the Max Reps Expected field only for Provisional and Active DCMs. For non-repeating Question Groups, this field is display-only.
- **Reps to Display** (Applies to character-based layouts only.) This setting determines the size of the scroll area in terms of the number of repeats displayed at one time; if set to 2, the user can see two repeats in the layout without scrolling. For non-repeating Question Groups, this field is display-only.
- **Enforce Max Reps?** (Applies to character-based layouts only.) This box prevents more than the set value of **Max Reps Expected** from being collected, except for users with privileged update. For a repeating Question Group, you can change this field only for Provisional and Active DCMs. For a non-repeating Question Group, this field is display-only, with a value of unchecked.
- **Has Border?** Selecting this causes the default layout generator to add a rectangular border around the Question Group. You can change this field for both Provisional and Active DCMs.
- **Prompt Position** This value determines the location of the response field's label, or prompt text, in the layout. Select **Left** or **Above**.
- **Protect Repeating Defaults?** Select this box to protect the displayed default values serving as prompts from being changed in the Layout Editor or during data entry; see [Using Default Repeat Values](#). You can change this field only for Provisional and

Active DCMs. For a non-repeating Question Group, this field is display-only, with a value of unchecked.

- **Save Repeat Defaults?** Select this box to create a set of rows with repeating defaults, even if the repeats are blank after data collection.

If checked, this DCM definition overrides the automatic deletion of any empty rows so that these rows are saved. This override allows the use of empty repeating default Question Groups as targets of derivation.

You cannot check this if there are no repeating defaults defined for any of the Questions in the Question Group.

- **Help Text** Add extra information, up to 200 characters, about the Question Group. The text you enter here is available for display during data entry. You can change this field only for Provisional and Active DCMs.
- **Normalized?** Check this flag if you are planning to batch load normalized lab data and want the input formats of the Batch Data Loader to accommodate normalized lab data modeled as lab test name/lab test result value pairs in the data files by automatically generating, for both inserts and updates, two response records for each incoming data record; see [Using Normalized Lab Data](#) .

 **Note:**

The **Normalized?** flag is available only for repeating Question Groups.

You may need to Save before checking **Normalized?** to enable the lists of values for the First and Second Repeat fields.

Enter values for the following fields only if you checked the **Normalized?** flag. You must first define the Questions in the Question Group.

- **First Repeat Descriptor Question** From the list of values, which displays all Questions in the Question Group that are collected in the DCM Subset, select the first Question in the normalized pair (the lab test name Question).
- **Occ Sn** (Question occurrence sequence number) The system enters this value when you select the first repeat descriptor Question. You cannot change it.
- **First Repeat Descriptor Label** The system enters the Default Prompt defined for the Question you selected as the First Repeat Descriptor Question. You can change it. This value is stored in `dcm_question_groups.repeat_descr1_label` and is available to use as a variable (`\DESCRIPTOR1\`) in Standard Text in a discrepancy comment wherever discrepancies are displayed.
- **Second Repeat Descriptor Question** From the list of values, which displays all Questions in the Question Group that are collected in the DCM Subset, select the second Question in the normalized pair (the lab test result Question).
- **Occ Sn** (Question occurrence sequence number) The system enters this value when you select the second repeat descriptor Question. You cannot change it.
- **Second Repeat Descriptor Label** The system enters the Default Prompt defined for the Question you selected as the Second Repeat Descriptor Question. You can change it.

Adding or Modifying DCM Question

After creating a new study DCM or having copied a DCM from another study, you can add more Questions to the Question Groups or modify the existing Questions. To add or modify a DCM Question:

1. From the **Definition** menu, select **DCMs**, then choose **DCMs**. The system displays the DCM Questions window.

 **Note:**

You can only copy or add new Questions into a Question Group if the Question Group is flagged as expandable. (Expandability is an attribute set at the Global Library level.) Oracle Clinical prevents you from copying Questions into a non-expandable Question Group.

2. Click the **DCM Question Groups** button. The DCM Question Groups window opens.
3. Choose a DCM Question Group name and click the **DCM Questions** button.
4. Select **Add Question** from the **Special** menu. The system displays the Locate Active Questions form.
5. Enter, or select from the list of values, the name of the Question you want to copy.
6. Click the **Commit** button. The system returns you to the DCM Questions window and displays a message that the copy succeeded or failed.

Alternatively, you can add Questions to a Question Group:

1. From the **Definition** menu, select **DCMs**, then choose **DCMs**.
2. Enter or select from the list of values, into the first blank record, the name of the object to copy.
3. Save.

You cannot create a new Question in the Definition subsystem. You can create new Questions and Question Groups only in the Global Library.

Configuring DCM Subsets

Using DCM subsets, you can have multiple occurrences of a DCM in a single DCI, but collect some DCM Questions in one subset and other DCM Questions in another subset.

All subsets contain the same Questions and Question Groups, and when you add a Question to one subset, the system automatically adds it to all the other subsets. The difference is that the Question's **Collected?** box can be selected in one or more subsets but not in others. A Question appears in data entry and in a layout only if it is marked as collected.

You can also associate different DVGs, or lists of allowable values, with the same Question in different DCM subsets.

Each subset can have one or more layouts.

- [To Create a New DCM Subset](#)

To Create a New DCM Subset

To create a new subset of a DCM, from the Data Collection Modules window, do the following:

1. Put your cursor on an existing DCM that you want to create a subset for. If the DCM has more than one layout, select the one you want to use for the new subset.
2. In the **Special** menu, choose **Create DCM Subset**. The system creates a new subset with the subset name (labeled "Name") given a default value of "Copy *original_name*."
3. Give the new subset a better name and, in the **Description** field, explain its purpose.
4. Click the **Question Groups** button. If there is a whole Question Group that you do not want to collect in the subset, deselect its **Collect in Subset?** box.

If there are individual Questions you do not want to collect in the DCM, but you do want to collect other Questions in the same Question Group, select the Question Group in the Question Groups window and click the **Questions** button. Then deselect the **Collect in Subset?** box of the Questions you do not want to collect.

5. (Optional) In the Questions window you can select a DVG, or list of allowable values, for a Question that is different in this subset than in other subsets of the same DCM.
6. Save your changes.

Configuring Qualifying Questions

If you need to collect responses to the same set of Questions more than once in the same DCI, you can group the Questions into a DCM and define a Qualifying Question for the DCM, the response to which identifies the set of responses. You then create the number of DCMs you need for the DCI.

For example, you may want to record a patient's blood pressure and other vitals in both the standing and the sitting position. You create a DCM called Vitals with a Qualifying Question named Vitals Position, for example, associated with a DVG that includes two values: Standing and Sitting.

You can define a Question especially for this purpose, or use an existing Question. If you create a new Question for the purpose, you may want to give it the same name as the DCM and, in the **Description**, note that it is used as the Qualifying Question for the DCM. Qualifying Questions must be associated with a DVG, or static list of allowable values.

Do one of the following to create two of the same DCMs within the DCI:

- In the DCM window, copy the DCM (see [Copying a DCM](#)) and give it a different name and a different qualifying value in the DCMs Qualifying Question tab; for example, create DCM VITALS_SIT and DCM VITALS_STAND and give VITALS_SIT the default value Sitting and VITALS_STAND the default value Standing.
- In the DCI window, click the **DCI Modules** button. Here you can enter the same DCM multiple times, each with a different qualifying value. For example, enter the

DCM called VITALS twice, once with the qualifying value Sitting and once with the qualifying value Standing.

The Qualifying Question prompt (defined as part of the Question) appears in the DCM header during data entry. If you define a qualifying value (in this case, either Standing or Sitting) for each occurrence of the DCM, that value appears in the DCM header after the prompt. If you do not define a default value, the field is blank in the DCM Header and the data entry operator must enter the appropriate qualifying value.

- [To Associate a Qualifying Question with a DCM](#)

To Associate a Qualifying Question with a DCM

You associate a Qualifying Question with a DCM in the Qualifying Questions tab of the Maintain DCMs window:

1. Put your cursor in the **Qualifying Question Name** field for the DCM for which you want to define a Qualifying Question.
2. From the list of values, which displays only DVG Questions, choose the Question you want to use.
3. Put your cursor in the **Qualifying DVG Name** field. The field enters the DVG that is defined for the Question. You can choose a different DVG from the list of values if you want to.
4. (Optional) Enter one of the DVG values in the **Qualifying Value** field. This value appears as the default value for the Qualifying Question in the DCM Header.

Using Repeating Question Groups

Sometimes you need to collect the same group of Questions multiple times at the same visit. Depending on your requirements, you can handle this different ways in Oracle Clinical:

- **DCM Qualifying Questions:** If there is a key difference in each collection—for example, you need to collect one set of Questions about the right eye and the same set of Questions again for the left eye; see [Configuring Qualifying Questions](#).
- **Repeating Question Groups:** If you need to collect the same set of Questions repeatedly within the same DCM, use a repeating Question Group. For example, if you need to collect vitals at predefined time Intervals after a dose, make the Vitals Question Group a repeating Question Group within in the DCM.

For more information, see:

- [Using Default Repeat Values](#)

Using Default Repeat Values

To simplify Question definition and to make Question prompts available in data extract views, you can define a repeating Question Group where the first question is used to display a label or prompt for the remaining questions in the row (or repeat). You specify the number of repeats expected for the Question Group in a particular DCM, and then specify the label to be displayed for each repeat (Occurrence Sequence Number, or Occ SN) of the first Question as a "default repeating value" and protect the value of that Question from update.

 **Note:**

Because you are using the same Question definition for each repeat of the response value Question, all responses must be collected as the same data type. If you want to use the repeating Question Group for both text and numeric Questions, you can collect all Questions as text and then use a derivation procedure to derive each numeric response value entered as text (based on the occurrence SN) into a Question with a data type of NUMBER.

Prompts defined as default responses are available in data extract views, unlike *boilerplate text* (text the layout generator derives from the Default Prompt values of Questions in a DCM) or text objects added in the DCM Layout Editor.

Example 1

As a variation on the interval example above, specify the expected intervals with default values for a question 'Interval'.

Example 2

Use default repeating values in batch loading normalized lab data; see [Using Normalized Lab Data](#) .

Example 3

If you want to collect a finding about various organs, such as the brain, the liver, and the heart, you can define an ORGAN Question to have default values of BRAIN, LIVER, and HEART for Repeats 1, 2, and 3 of the Question. Then define the next Question as FINDING. In this case, the values for the Question ORGAN would not be user-collected, but would simply serve as prompts for the Question FINDING. The data entry form would look like this:

Organ	Finding
BRAIN	<i>field for user-entered response</i>
LIVER	<i>field for user-entered response</i>
HEART	<i>field for user-entered response</i>

resulting from the following Repeating Default Value definition:

Occurrence SN	Organ Question Value
1	BRAIN
2	LIVER
3	HEART

To create default repeat values:

1. From the **Definition** menu, select **DCMs**, then choose **DCMs**.
2. Click the **DCM Question Groups** button and select a repeating Question Group.
3. Select the Question to which to assign default repeat values.
4. Click the **Default Repeat Values** button.
5. Enter the number of the repeat and the value to display for that occurrence of the Question.

 **Tip:**

If you are using default repeating values as question prompts, select **Protect Repeating Defaults** and **Save Repeating Defaults** for the Question Group and in the **Status Comment** field of the DCM enter this string: `ADDREPDEF`.

This ensures that, if a user enters data in a CRF that then has a status of Saved Complete or Pass 1 Complete, and additional repeats are later added (Max Repeats is increased) to the form and additional repeating default values are specified for the new repeats—and a new form version generated and the document migrated to it—when the user opens the document again, the new repeating default values (prompts) are displayed in RDC Onsite. The user must open and save the document to save the repeating default values.

Defining Conditional Branching

Branching occurs when a data entry interface directs a user to a different set of questions on a CRF, depending on the response entered for a question. There are several types of branching, including Indicator branching, conditional branching, and flexible study design; see [Defining Response-Dependent Branching](#) for a comparison of these types.

In conditional branching, you specify a *source* Question and a response value or range of values that, when entered as patient data, triggers the activation of a *target* Question in the same DCM. You can specify multiple responses or ranges of responses for the same source Question, and different target Questions for each response or range of responses.

For example, define a Question "Are you pregnant?" with a DVG containing three values: Yes, No, and I Don't Know. For the response Yes, define a target of the first Question in a Question Group about the pregnancy. For I Don't Know, define a Question "Is lab test scheduled?" as the target, with the Question "Date scheduled" nested. For No, either define a target of "Verification that lab test confirms" or do not define a target at all, in which case the cursor goes to the next sequential Question during data entry.

You can define conditional branching on numeric, time, and date Questions. You can specify ranges using greater than (>) or less than (<) operations in combination with each other or with the equal to (=) operation. For example, use a numeric range for a Question **Year of Birth** to bypass age-specific Questions that are inappropriate for a particular patient.

To define a conditional branch:

1. From the **Definition** menu, select **DCMs**, then choose **DCMs**. The system displays the Maintain DCMs window. You can also define conditional branching directly in the Global Library: From the **Glib** menu, select **DCMs**, then choose **DCMs**.
2. Query for or create the DCM.
3. Click the **DCM Question Groups** button. The DCM Question Group window opens.
4. Select the DCM Question Group containing the Question that initiates the branching.

 **Note:**

If you are using graphic layouts (for RDC Onsite), do not define conditional branching within a repeating Question Group.

If you are using character-based layouts (for Oracle Clinical) you can define conditional branching within a repeating Question Group. If the target Question is before the source Question in the Question Group, entering the trigger response value during data entry moves the cursor to the target Question in the next repeat of the Question Group.

5. Click the **DCM Questions** button.
6. Select the initiating, or *source* DCM Question. You may want to use a Question with a DVG; see [Creating and Using DVGs](#) , especially if the source Question is a character Question.
7. Click the **Conditional Branch** button. The system displays the Conditional Branches for (Question Group, Question) form.
8. For each response that should result in a branch, specify the target Question for that branch:
 - a. In the Value column, enter the response value that will serve as the trigger for the conditional branch you are defining.
 - If the Question has a DVG, select a value from the list.
 - If the Question is of data type CHAR but does not have a DVG, only NULL or NOT NULL are allowed.

 **Note:**

If the source Question is of data type CHAR, and you want a particular response (such as Y, WHITE, or SEVERE) to trigger a branch, create a Discrete Value Group (DVG) containing all possible responses to the source Question; see [Creating and Using DVGs](#) . You may also want to use a DVG for some numerical Questions.

If you want to trigger the branch if any response is entered, do not assign a DVG to the Question and enter NOT NULL here.

If you want to trigger the branch if a response is not entered, enter NULL here. In this case, tabbing into the field and out again triggers the branching.

- If the Question is of data type NUMBER any numerical value is allowed.
 - If the Question is of data type DATE the value must be in a date format.
- b. In the target Question Group Name column, select the Question Group that you want to branch to if the specified response value is entered. This can be the same Question Group that contains the source Question.
 - c. In the target Question Name column, select the Question in the Question Group that you want to collect next, based on the response to the source Question.

- d. In the Occ Sn (Occurrence Sequence) column, enter 0 (zero) if the Question is included in the Question Group only once. If the Question occurs more than once in the Question Group, enter the number for the occurrence of the Question that you want to use, starting with 1.
- e. Select a Branch Oper (ation) value:
 - Select **EQ** (Equals, =) if the branch should be triggered if the patient response value entered is the same as the specified response value.
 - Select **GT** (Greater Than, >) the branch should be triggered if the patient response value entered is greater than the specified response value.
 - Select **LT** (Less Than, <) the branch should be triggered if the patient response value entered is less than the specified response value.

You can create combined conditions *greater than or equal to* (>=) or *less than or equal to* (<=) by defining multiple branches with the same source and target Questions, one with the operation GT or LT and the other with EQ.

9. Repeat the last task for each source Question value that you want to serve as a trigger for a conditional branch.

10. Save.

The system automatically sets the **Has Conditional Branch?** value for the DCM Question upon saving.

If you are using graphic layouts, the conditional branch definition is incomplete until you define conditional blocks in the DCM graphic layout editor; see [Creating and Maintaining Conditional Blocks](#) , and the following:

- [Conditional Branching and Character-Based Layouts](#)
- [Conditional Branching and Graphic Layouts](#)

Conditional Branching and Character-Based Layouts

For character-based layouts, which Oracle Clinical uses, conditional branching has no visual impact and is immediately in effect in the data entry form if there is a layout in use. The conditional Questions are always visible and displayed the same way as other Questions. The branching logic makes the cursor jump to the target Question when the user enters the response trigger value, but the user can navigate back and enter or update intervening fields.

Conditional Branching and Graphic Layouts

To complete the definition for graphic layouts, which RDC Onsite uses for data entry, you have to do more work in the layout editor. The default layout identifies all source and associated target Questions you have defined. For each source Question, you create target blocks for each target Question. Then, you can move other Questions—except the target Question—in and out of the block(s). These Questions do not necessarily have to be part of the same Question Group as the target Question. In addition, you can reposition Questions such that the target Question is not necessarily the Question that the cursor automatically goes to first when the trigger response is entered; see [Creating and Maintaining Conditional Blocks](#) .

 **Note:**

You control whether disabled blocks are displayed as hidden or greyed in the DCI Forms Local Study Settings window at the database or study level, in the Represent Disabled Block setting; see [Configuring Study-Level DCI Form Settings](#).

If you have existing conditional or indicator branching definitions used in Oracle Clinical data entry, you can make use of them in your graphic layouts by adding the graphic conditional blocks to your layouts.

Tracking Conditional Block Status at the DCM Level

The display-only field **CB Graphic Blocks Defined?** in the DCM Conditional Branches window and the DCM Questions window indicates whether the target conditional blocks for the current source Question have been defined in the graphic layout:

- The checkbox is checked once target conditional blocks are defined in the Layout Editor.
- The checkbox becomes unchecked and the target conditional blocks are deleted when:
 - The DCM layout is regenerated.
 - The conditional blocks for the current source Question are explicitly deleted by the user in the Layout Editor.
 - Certain types of updates are made to the conditional branch definition. See "How Updating Conditional Branching Definitions Affects DCM/DCI Graphic Layout Flags" for more information.

How Updating Conditional Branching Definitions Affects DCM/DCI Graphic Layout Flags

After the set of target conditional blocks for a source Question are defined in the Layout Editor, updates made to the base conditional branch definitions can affect the DCM and DCI graphic layout settings. The following table provides details:

Table 9-1 Impact of Updating Conditional Branching Definitions

Update	Indicator/ CB Graphic Block(s) Defined?	Avail- able?	Needs Edit / Verifi- cation?	Post- Edit Up- dates?	Immediate Effect in RDC Onsite Data Entry	User Action Needed in Layout Editor
a. Update Value or operator for an existing branch. Block already exists.	No effect until you open the layout editor. Then the blocks are deleted with a confirmation message. When you return to DCM Questions, the check box is unchecked.	No effect	No effect	No effect	New branching behavior is observed	None
b. Add a new branch or update target for an existing branch where target is already used in an existing branch for this source Question. Block already exists.					Any of these changes may result in conditional branch violations.	
c. Delete a branch. Block/target Question is referenced by another branch for this source Question and should not be deleted in the LE.						
d. Add a new branch. (Target is NOT already referenced in a defined block. New block is needed.)	No immediate effect; becomes unchecked when you open the Layout Editor and blocks are deleted.	Uncheck ed	Checked	Checked	Branch has no target block. Branch goes to first enabled Question in the form following the Question with the conditional branch.	When you open the DCM Layout Editor, all conditional blocks are automatically deleted. The user must recreate the remaining conditional blocks.
e. Update Target in an existing branch. (Target is NOT already referenced in a defined block. New block is needed.)						
f. Delete a branch. (Block is not referenced by another branch, so should be deleted.)						
g. Turn a Question into a branching source Question (by defining conditional branches).	No effect (initialized to unchecked)	No effect	No effect	No effect	No effect. No blocks, no hiding, no branching.	Create conditional blocks.
h. Check as mandatory a Question which appears in a conditional block.	No effect	Uncheck ed No effect	Checked No effect	Checked No effect	A mandatory discrepancy will be raised. If the block is disabled, the user is unable to resolve the discrepancy.	If mandatory Question is in a conditional block and does NOT have a default value, it must be moved.

Table 9-1 (Cont.) Impact of Updating Conditional Branching Definitions

Update	Indicator/ CB Graphic Block(s) Defined?	Avail- able?	Needs Edit / Verifi- cation?	Post- Edit Up- dates?	Immediate Effect in RDC Onsite Data Entry	User Action Needed in Layout Editor
i. Uncheck Collect in subset for a source Question or a Question in a target block. (Note this action is disallowed for a Question that is definitionally identified as a target on the Conditional Branches form.)	No effect	Uncheck ed	Checked	Checked	Label and response fields are still displayed. Response can no longer be entered. However, a pre-existing response can be updated.	Edit the layout and save. Create new form version.*
j. Uncheck Displayed. (Can only do this for a non-enterable Question.)	No effect	Uncheck ed	Checked	Checked	Question continues to be displayed (still not enterable), Branching is as before.	Same as for item j, above.

*Upon edit, the Question is automatically removed from the layout. If this is a source Question with conditional blocks, the Layout editor should also automatically delete conditional blocks for this Question. If this is a target Question, the system removes the Question from the layout.

Using Normalized Lab Data

To use normalized lab data, you must define a pair of Questions in a repeating Question Group to collect the lab test name and lab test result, respectively, for multiple lab tests. Make the number of occurrences of the Question Group equal to or greater than the number of lab tests required.

When you batch load lab data in a normalized format, the system automatically creates two records from each normalized lab data record. The lab test name is then stored as a Question response and you can use it in data extract and in validation Procedures to identify the test name for each test result.

For example, from records with a normalized input file format as follows:

```
,,2,,VISIT1,0,20110101,,HEMA,HEMA,HEMA,LAB,HCT,0,1,43
,,2,,VISIT1,0,20110101,,HEMA,HEMA,HEMA,LAB,HGB,0,2,17
```

where the lab test names are HCT and HGB and the respective lab test values for the patient are 43 and 17: a DCM question, LTEST, has been defined to collect the lab test Question name and another DCM question, LVALUE, has been defined to collect the lab test result, the Batch Data Load process converts the preceding two input records to the following four records:

```
,,2,,VISIT1,0,20110101,,HEMA,HEMA,HEMA,LAB,LTEST,0,1,HCT
,,2,,VISIT1,0,20110101,,HEMA,HEMA,HEMA,LAB,LVALUE,0,1,43
,,2,,VISIT1,0,20110101,,HEMA,HEMA,HEMA,LAB,LTEST,0,2,HGB
,,2,,VISIT1,0,20110101,,HEMA,HEMA,HEMA,LAB,LVALUE,0,2,17
```

The system stores "HCT" as the response value for REPEAT_SN 1 of Question LTEST and "HGB" for REPEAT_SN 2 of LTEST.

To use normalized lab data, define the following Questions and include them in a repeating Question Group in a DCM:

- **Lab Test Name** You must define a Question to collect the name of the lab test. It must have a data type of CHAR, and it must not be derived.
- **Lab Test Result** You must define a Question to collect the lab test result. It must have a Question type of CHAR and a data type of CHAR, even if the lab data is expected in numerical format. This accommodates responses to numeric Questions with text such as "TRACE" and the use of special characters such as less than (<) and greater than (>), and allows you to use the same Question for all lab tests.

The lab test result Question must not be derived, but you may want to define an additional Question that is derived and numeric to hold the value of numeric tests, and a derivation Procedure to populate its value; see [Validation and Derivation Procedures](#).

 **Note:**

Since these same two Questions are used for all lab tests that are loaded in a normalized format, it is difficult to understand discrepancies based on lab test Questions. However, you can use the repeat sequence number as a variable (\REPEAT_SN) in a Standard Text definition to identify which lab test result is discrepant wherever discrepancies are displayed. For information on Standard Text, see the *Oracle Clinical Administrator's Guide*.

Defining Date Formats

You can specify the order in which the system displays a Question's date components (day, month, and year). Set the date format at the DCM level.

During Log-In, by selecting **Preferences** from the **Special** menu, the data entry operator can set the value of a user preference, data entry display format, to determine the format of the DCI and DCM dates' values:

- European format is DD-MM-YYYY
- Swedish format is YYYY-MM-DD
- U.S. format is MM-DD-YYYY
- Standard format is DD-MON-YYYY
- Dynamic format is based on the log-in header

If you set the date format at the DCM level to US, EUROPEAN, SWEDISH, or STANDARD, the display format for Date-type Questions in data entry is as described above, regardless of the values of the data entry display format user preference set in the Log-In form. However, if you set the date format for the DCM to DYNAMIC, the display format for Date-type Questions in data entry defaults to the value of the data entry display format used in the log-in process.

Because the DYNAMIC date format allows more flexibility for the data entry operator, it is especially useful in distributed studies with different country locations where forms might be identical except for their date format. In these situations, the date displays in the appropriate country date format at a location other than its origin.

To set the country date format:

1. From the **Definition** menu, select **DCMs**, then choose **DCMs**. The system displays the Maintain DCMs form.
2. Query or create the DCM.
3. Select the **Date Format** field.
4. Use the List function to select the appropriate value.
5. Save.

Scheduling DCMs

You can schedule DCMs by associating them with Clinical Planned Events (CPEs), which are ordinarily clinical visits by patients. You can define a DCM once and collect it at multiple visits.

If you are using RDC, do not schedule DCMs directly to CPEs. Instead, define a DCI Book that defines associates all study visits with DCMs. You can also use DCI Books for Oracle Clinical data entry; see [Defining DCI Books](#)

In this section:

- [Associating DCMs to CPEs](#)
- [Schedule DCMs Form](#)
- [Copying the CPE Schedule](#)
- [Copying the DCM Schedule](#)
- [Resequencing DCMs](#)

Associating DCMs to CPEs

The study's protocol determines the number of CPEs and the Intervals between them. You associate each DCM to a specific clinical visit. Mapping a DCM to a CPE lets Oracle Clinical track patient data over time. Mapping a DCM to several CPEs lets Oracle Clinical track specific study Questions under different circumstances, usually after the passage of time or under various experimental situations.

You create a study's Intervals and CPEs in the Design subsystem. Each DCM corresponds to medical data at an instant in a patient's history, and that instant is defined by associating the DCM with the clinical visit during which the data is collected. The results of this mapping task are presented in a matrix format, with CPEs along the top row as an event sequence, and DCMs or DCM subsets on the left as a data collection sequence. Each intersection of row and column defines a possible association between DCM and CPE.

Values for these points are as follows:

- **E** – You expect a site to collect the DCM's data at this CPE.
- **O** – Collection of the DCM is optional at this CPE.
- **(Blank)** – DCM is not scheduled to be collected for this CPE.

When you have completely defined the associations, commit the schedule.

Schedule DCMs Form

You schedule DCMs from the Schedule DCMs form. From the **Definition** menu, select **DCMs**, then choose **Schedule DCMs**. The form appears with a pop-up window for you to select a clinical study. The system arranges the DCMs in the collection sequence. From the **Special** menu of the Schedule DCMs form, you can copy DCMs, copy CPEs, resequence the DCMs, or change the study.

Navigating to any point on the grid and clicking the **DCMs** button displays a window with detailed information on the current DCM. Similarly, pressing the **Clinical Planned Events** button displays detailed information on the current CPE.

Copying the CPE Schedule

You may find you want to use the DCM schedule of one CPE for another CPE—that is, you want to copy the DCM values from one CPE column to another. To copy CPE values:

1. From the **Definition** menu, select **DCMs**, then choose **Schedule DCMs**. Select the DCM schedule with the CPE you want to copy.
2. From the **Special** menu, select **Copy CPE**. A pop-up window appears, Copy Schedule for CPE.
3. Enter a CPE in the pop-up window to copy the schedule from, or use the List function to view and select valid CPEs.
4. Click:
 - a. **OK** to copy the schedule from one CPE.
 - b. **Cancel** to return to the Schedule DCMs form without copying.

Copying the DCM Schedule

You may find that you want to use the CPE schedule of one DCM for another DCM—that is, you want to copy the CPE values from one DCM row to another. To do this, follow this procedure in the Schedule DCMs form:

1. Navigate to Copy DCM or CPE Definition, DCMs, Schedule DCMs, Special Menu, Copy DCM.
2. Select **Copy DCM** from the Special menu. A pop-up window appears, Copy Schedule for DCM.
3. Enter a DCM in the pop-up window to copy the CPE schedule from, or use the List function to browse the valid DCMs.
4. Click **OK** to copy the DCM schedule, or cancel and return to the Schedule DCMs form without copying.

Resequencing DCMs

The DCM sequence is the order in which DCMs are displayed in reports (except the DCM Detail Tracking Matrix Report, which is ordered by ascending DCM ID) and on the DCM schedule. When you first enter the Schedule DCMs form, you may receive a message that you must resequence the DCMs. This is because the DCMs in the study are not currently sequenced consecutively—probably because a DCM was added or deleted from the study.

The system requires that you alter this sequencing, making the DCM consecutively numbered from 1 to n before proceeding.

To resequence the DCMs:

1. Select **Resequence DCM** from the **Special** menu. Oracle Clinical displays the DCM Sequencing form.
2. Change the sequence number (Seq #).
3. Select **Commit**. The system moves the DCM to a new position in the sequence.

Creating Character-Based Layouts

This section describes designing character-based data entry layouts used for data entry in Oracle Clinical. If you are using RDC Onsite, see [Creating Graphic Layouts](#)

Two tools help you create data entry form layouts:

- **Default DCM Layout Generator** Creates an initial layout of data entry fields on a DCM by reading its DCM Question, DCM Question Group, and DCM definition information. This tool takes as input the preferences specified in the DCM definitions—including prompt position, prompt text, and Question sequencing—which makes it capable of generating several different styles of DCM layouts.

Note:

The Default Layout Generator assigns a display order to each field from left to right and top to bottom according to the DCM Question's Disp Seq# (Display Sequence Number) value. If the Disp Seq# values are changed for the DCM Questions, the existing layout is not affected and the 'Regen Req'd?' flag for the DCM is not checked. The display order of the fields in the character layout is changed only when the layout is defaulted again.

- **DCM Layout Editor.** Use the DCM Layout Editor to customize the appearance of default layouts, refining the DCM's interface to be more intuitive for data entry.

The basic steps to create data entry screens are:

1. Generate a default layout for the specified DCM.
2. View the default layout with the DCM Layout Editor.
Optionally, return to the DCM and change default settings and regenerate the default layout.
3. Modify the default layout with the DCM Layout Editor.
4. Test the layout by using it in the test environment.
5. Move the layout from the test environment into production.

For more information, see:

- [Creating Default DCM Layouts](#)
- [Creating Additional Layouts](#)
- [Editing DCM Layouts with the DCM Layout Editor](#)

- [Sequencing Data Entry](#)
- [Using Data Entry Forms in Test and Production Modes](#)

Creating Default DCM Layouts

To create a new DCM character-based layout:

1. Put your cursor on an existing DCM that you want to create a layout for. If the DCM has more than one subset, select the one for which you want the new layout.
2. In the Special menu, choose **Default Character Layout**. A message appears in the status bar stating that a layout has been successfully created, and the system sets the Data Entry Generation Time value. You can see the timestamp in the Character Layout tab.

Creating Additional Layouts

You can create multiple layouts for the same DCM; for example, layouts in German and French. If a DCM has multiple subsets, you can create multiple layouts for each subset.

To create an additional layout for a DCM, from the Data Collection Modules window, do the following:

1. Put your cursor on an existing DCM that you want to create a layout for. If the DCM has more than one subset, select the one for which you want the new layout.
2. In the Special menu, choose **Create DCM Layout**. The system creates a new layout with the Layout number incremented by one.
3. In the **Description** field, explain the purpose of the new layout.
4. If you want to prevent a Question from being displayed in the new layout, click the **Questions Group** button, select the Question's Question Group, click the **Questions** button, and deselect **Displayed?** for the Question.
5. Edit the new layout (see [Editing DCM Layouts with the DCM Layout Editor](#)).

Editing DCM Layouts with the DCM Layout Editor

The DCM Layout Editor allows you to modify data entry forms to more closely resemble CRFs, to improve the efficiency of data entry, or to add on-screen information. The DCM Layout Editor can display all the Questions in a particular Question Group, including all repeating instances of fields. The ability to create more than one layout for a single DCM, combined with the ability to tailor each one, means that you can create different layouts for different uses.

An example of this flexibility is a case where two laboratories collect data in different Question sequences and with different labels. You need create only one DCM in Oracle Clinical, with two layouts tailored to each laboratory's usage. The use of different DCM layouts does not affect the underlying structure of the DCM, nor the organization of the resulting data in the database.

To open the DCM Layout Editor:

1. From the **Definition** menu, select DCMs, then choose DCMs.
2. Select the DCM to edit
3. From the Special menu, select **Edit Layout**.

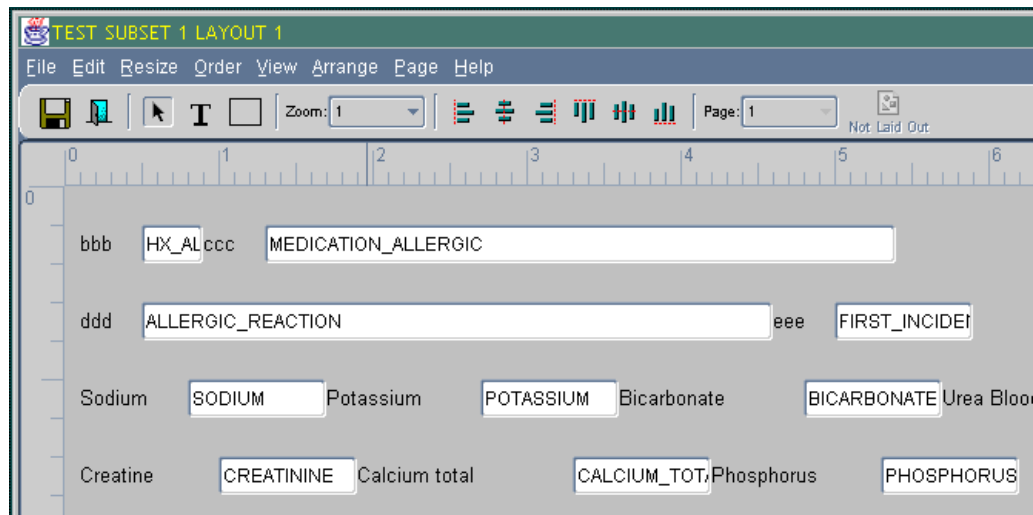
The DCM window opens the DCM Layout editor in a new window. All windows in your session remain open. If you return to your session, you see a message stating that a Layout Editing session is in progress. (Check that you have saved your work in the DCM Layout Editor, if necessary, before closing this message box.)

For more information, see:

- [Parts of the DCM Layout Editor](#)
- [Positioning Objects](#)
- [Selecting Objects](#)
- [Editing or Adding Text](#)
- [Adding Boxes](#)
- [Moving Objects](#)
- [Arranging Objects](#)
- [Deleting Objects](#)
- [Resizing Objects](#)
- [Controlling Spacing and Sizing of Objects](#)
- [Changing Page Numbers of Objects](#)

Parts of the DCM Layout Editor

The DCM Layout Editor window has a menu, a toolbar, a work area, and—at the bottom of the window—a task bar. A sample detail of the DCM Layout Editor's window follows:



Objects are any selectable parts of the layout: response fields, field labels, text, and graphic shapes. In the sample illustration above:

- ddd is the label text

- ALLERGIC_REACTION is a Question name, and the surrounding white box is the representation of the field's shape and position in the corresponding data entry form.

Positioning Objects

The DCM Layout Editor allows positioning objects within a *character grid* in one-character increments. The administrator of the Web server controls the character grid size by changing registry settings (described in Appendix A of the *Oracle Clinical Administrator's Guide*). You can display an overlay of the character grid on your workspace by dropping down the View menu and selecting the Show Grid box. Rulers along the top and left sides of the work area display the coordinates of the cursor position. (The numbers on the ruler represent tens of characters; the number 1 represents ten characters.) You can also see the exact position of a selected object, or the mouse pointer, in the left-side corner of the taskbar.

Selecting Objects

Selected objects display *pull handles* (small black squares at the corners and midpoints).

Selecting Individual Objects

To select one response field, field label, or text object: with the selection tool, click inside the borders of the object. For a box object, either position the point of the selector over the border of the box and left-click, or position the cursor inside the box and click-drag in a diagonal direction.

Selecting Contiguous Objects

To select all the objects in an area of the layout: Click-drag the selector in a diagonal direction across the area.

Selecting Non-contiguous Objects

You can Control-click to select or deselect objects regardless of their geographic location in the layout. If you are Control-clicking a box, note that you must position the pointer of the selector tool directly on its border.

Editing or Adding Text

The DCM Layout Generator creates text objects of the Questions' Default Prompt values, referred to sometimes as boilerplate text. You can change the text in these objects as well as in new text objects you add to the layout. To edit the text in a text object, double-click the object. The system changes to text edit mode. To stop editing a text object, click outside the text object.

You can add extra text objects to the layout. Text objects cannot exceed one character in height. (Stack multiple text objects to create multiple lines.) To create new text objects:

1. Click the Text button in the tool bar.
2. Position the mouse pointer in the layout at the position where the text starts. You can see the character grid location of the pointer on the rulers and in X,Y character coordinates on the taskbar.
3. Click. The system places a text field in the layout, snapped to the coordinates of the closest positions in the character grid, and changes to text editing mode.
4. Enter the text.

Your organization sets the DCM layout text font characteristics with registry entries on the Web server computer. For text objects, the registry entry is OC_DE_TEXTFONT. For field text, the registry entry is OC_DE_FIELDFONT. Note that these registries control layout screens and data entry screens.

Adding Boxes

You can add ruled rectangles, or boxes, to enhance the usability of the data entry form. You can, for instance draw a rectangle around a Question Group to indicate to the data entry operator that the fields are related.

1. Click the Rectangle button in the toolbar.
2. Position the mouse pointer in the layout at the point of a desired corner. You can see the character grid location of the pointer on the rulers and in X,Y character coordinates on the taskbar.
3. Click and drag the pointer diagonally to the desired position of the opposite corner. The system places a text field in the layout, snapped to the coordinates of the closest positions in the character grid.

Moving Objects

You can move objects with either the mouse or with your keyboard's arrow keys. All movements increment to the closest position on the character grid.

Moving an Object with the Mouse

You can move an individual object with the mouse. For a rectangle object, place the tip of the mouse pointer on the perimeter of the rectangle. (Try to avoid the midpoints and corners, which are the locations of the grab handles.) Mouse-click and hold while you drag the rectangle to a new position. For any other type of object, place the tip of the mouse pointer within the object's perimeter, then mouse-click and hold while you drag the object to another position.

Moving Objects with the Keyboard Arrows

You can move one or more selected objects with the keyboard arrows. This approach is effective for maintaining the geographic relationship among a local set of objects (for instance, response fields and their labels), while moving them.

Arranging Objects

The DCM Layout Editor has a set of tools for aligning selected objects, and a tool for controlling the distance between repeating Questions.

Aligning Selected Objects

The DCM Layout Editor has six tools for systematically aligning groups of selected objects by the objects' axes or borders. You invoke the tools either from the Arrange menu or from toolbar buttons. Position the mouse pointer over a symbol to read a description of how it aligns selected objects.

Setting Distance Between Repeating Questions

You can adjust the spacing between repeating Questions. Select a set of repeating Questions, then from the Arrange menu, select Lines Per Repeat. The Lines Per Repeat dialog box opens. The lines per repeat increment corresponds to the character

grid. Enter a number and click OK. The system adjusts the distance between repeating Questions a corresponding number of character cell increments.

 **Note:**

The Lines Per Repeat spacing tool only works for Question Groups with a prompt position value of 'L.' Its prompt position must be set to the Left radio button in the DCM Question Groups dialog box. (See [Creating and Maintaining Question Groups](#) for more information.)

Deleting Objects

To delete an object, select it and press the Delete keyboard key, or use the keyboard combination Control-X. Alternatively, from the Edit menu, select Delete. If the object is a non-expendable part of the DCM, deleting without pasting generates an error message.

Resizing Objects

You can change the height and width of rectangles and text objects, and the lengths of text fields. You can also systematically resize a group of selected items in one operation.

Resizing Individual Rectangles or Text Objects

Select the object to expose its pull handles (small black boxes at the midpoints and corners of the object). Place the point of the mouse pointer over a pull handle. Click and hold while you drag the pull handle. You must pull the handle at least one character length for a change to take place.

Resizing Individual Response Fields

You can only lengthen the width. Select the field to expose its pull handles. Place the point of the mouse over one of the drag handles at either end of the field. Click and drag the handle at least one character length.

Resizing Widths of Multiple Selections

Use the tools in the Resize menu to change the widths of all selected objects to a uniform size, either:

- the shortest selected width
- the average of all of the selected widths (to the nearest character)
- the longest selected width

Controlling Spacing and Sizing of Objects

Layout controls are independent of the default font setting. These controls enable you to specify the spacing of the objects in your data entry layouts and the length of field. The values are in pixels.

You control the appearance of the data entry panel (the lower panel in the data entry window), with parameters `de_gridwidth` and `de_gridheight`. You must set these values.

You adjust the width of list of values columns in the data entry form with the parameter `de_lgridwidth`. This parameter is optional. If you don't supply a value, the default value is 11

pixels You adjust the size character width of List of Values entries in data entry screens with parameter `de_lgridwidth`.

To adjust these values, edit file `formsweb.cfg`, which is located at `oracle_home\forms60\server`. Follow these instructions:

1. Open `formsweb.cfg` in a text editor.
2. In the `[opa45]` section, find the line that begins: `form=opamenu.fmx`.
3. Add the following text string to the end of this line:

```
de_gridwidth=number de_gridheight=number
```

Where *number* is the parameter value (in pixels) you choose.

4. Save and close `formsweb.cfg`.
5. Test your settings by opening a data entry form.

For default DCM data entry layouts, Oracle recommends you use `de_gridheight=18` `de_gridwidth=7`. If you re-arranged objects in earlier layouts, Oracle recommends you use `de_gridheight=27` and `de_gridwidth=10`. DVG LOVs are objects in the bottom panel of the data entry form. To adjust the width of DVG list of valueslist of values columns, add to the `otherparams` line: `de_lgridwidth=number` Where *number* is the parameter value you choose. If you do not add this line, the system sets the `de_lgridwidth` default value to 10.

Here is a sample text string for new default layouts:

```
form=opamenu.fmx term=opahome\opa_fmrpcweb.res de_gridheight=18  
de_gridwidth=7
```

Here is a sample text string for revised default layouts:

```
form=opamenu.fmx term=opahome\opa_fmrpcweb.res de_gridheight=27  
de_gridwidth=10
```

Changing Page Numbers of Objects

Select the object(s) you want to move to another page. Select Cut to move to the clipboard. Use the Page list to move to the page where you want to place the object(s). Select Paste, then click where you want to place it. The editor pastes objects at the clicked mouse location if they fit there. Otherwise they remain in the clipboard, and an error message appears.

The following are the function of each menu entry:

- File
 - Save - Update the database with the current version of your edits.
 - Validate - Test that the layout conforms to layout rules that prevent, for instance, overlapping fields.
 - Revert - Select Revert to return to the previous saved version without exiting the Layout Editor.
 - Exit - This choice performs a number of tests to ensure that the current layout conforms to DCM and layout rules.
 - Exit without saving - Choose this entry to bypass conformance checks. (The system still prompts you if you have unsaved work.)

- Edit
 - Cut - Copy the selection to the clipboard and remove it from the layout. (Control-X)
 - Copy - Copy the selection to the clipboard. (Control-C)
 - Paste - Place the contents of the clipboard on the current page. (Control-V)
 - Undo - Return the layout to its state prior to the last action. (Control-U)
 - Redo - Return the layout to its state prior to the last Undo.
- Resize
 - Resize to smallest width - Adjust the width of selected objects to the width of the narrowest selected object in one operation.
 - Resize to average width - Adjust the width of selected objects to the average width of the selected objects in one operation.
 - Resize to largest width - Adjust the width of selected objects to the width of the longest selected object in one operation.
- Order
 - Order by clicking
 - Order by entering Sequence Number
 - Display
 - Order Current Page by Row/Column
- View
 - Show Grid box
 - Zoom (radio buttons for various magnifications)
 - Center on selection box
- Arrange
 - Lines Per Repeat
 - Align Left
 - Align Middle Horizontally
 - Align Right
 - Align Top
 - Align Middle Vertically
 - Align Bottom
- Page
 - Next
 - Previous
 - Insert
 - Delete - You must first delete or move elsewhere all objects on the page; otherwise the editor displays an error message and prevents you from continuing.
- Help
 - Help - Opens context-sensitive HTML-based extended online help.

Sequencing Data Entry

A data entry form has a *cursor visitation order*: the order in which the response fields become active as a data entry operator completes them. The default Layout Generator assigns visitation order values to each field from left to right, top to bottom according to the Questions' Disp Seq# values in the DCM Questions form. Until you use the DCM Layout Editor's Order utility, fields retain their default visitation order values, even if you move them in the Layout Editor. If a default or modified layout does not have the most efficient or intuitive visitation order, you can resequence the values.

To display a layout's cursor visitation order, from the Order menu, select Display. The DCM Layout Editor replaces each response field's Question name with its cursor visitation order value. To simplify reordering, the values are multiples of 10. For instance, the DCM's first active field has a value of 10, and the next field in the visitation order has a value of 20. (You cannot adjust the visitation order from this menu path. See the following sections for changing the visitation order.) The system replaces the layout toolbar with the ordering mode toolbar.

There are three visitation ordering tools, each described in the following sections.



Note:

You cannot insert the visitation order value of a Question into the visitation order sequence of a repeating Question Group unless it is a member of the repeating Question Group.

For more information, see:

- [Ordering by Row and Column](#)
- [Ordering by Clicking](#)
- [Ordering by Entering Sequence Numbers](#)
- [Validating a Layout](#)
- [Saving a Layout](#)
- [Closing the Editor](#)
- [Laid Out Versus Not Laid Out Objects](#)
- [Using Undo or Revert](#)

Ordering by Row and Column

When the system generates the default layout it assigns a visitation order to the fields from left to right, from top to bottom. You can assign this same visitation order scheme to a rearranged layout. The row/column visitation order might suffice for data entry, or you can use it as a starting point for further manipulation with one of the other ordering tools. Navigate to Order, then Order Current Page by Row/Column. With this tool you remain in layout mode. (To view the results, from the Order menu, select Display. To return to the layout after viewing the visitation order, click the Exit button in the toolbar, which appears in the editor's toolbar as a doorway.)

Ordering by Clicking

Use the Order by Clicking tool to set the visitation order by clicking each successive field in the layout: click once in the field you want to be first, then click once in each successive field in the new order. As you click each field, the system automatically adjusts the order values of the remaining fields. Note that once you have clicked as many times as there are fields, continued clicking assigns the last sequence number with each click.

When you finish ordering, click the Door-close button in the toolbar to return to layout mode. Or, if you are unsatisfied with the results of your pass through the fields, click the Door-close button. You can then reopen the Order by Clicking tool (or Order by Entering Sequence Numbers, described next), and start again.

Ordering by Entering Sequence Numbers

You can select a field and change its visitation order by entering a value. From the Order menu, select Order by Entering Sequence Number. You select a field and enter a number, and the system adjusts the values of all other fields accordingly. You can either enter the values manually or with the plus and minus incrementing buttons, described in the next sections.

Reordering Manually

To insert a field's visitation between two consecutively ordered fields, assign the inserted field the greater value. (You can also assign a value between the two values.) The system renumbers the fields to the new sequence. As an example, a rearranged layout might have a visitation order like:

Field	Visitation Order Number
WEIGHT	10
HEIGHT	40
BIRTHDATE	20

To sequence the visitation order of these fields top to bottom, change the value of **HEIGHT** from 40 to 20 (or, for instance, to 15), and select another field. The system reorders the fields to:

Field	Visitation Order Number
WEIGHT	10
HEIGHT	20
BIRTHDATE	30

Reordering with the Plus and Minus Buttons

The Order by Entering Sequence Numbers tool also includes **Plus** and **Minus** buttons in the toolbar that you can use to change individual sequence values. You select a field, then click either button to change its value by one value. In the example in the previous section, to change **HEIGHT** from 40 to 20, you would select 40, then click the **Minus** button twice. The results are the same as manually entering 20.

Validating a Layout

The DCM Layout Editor includes several constraints to prevent lost objects or other layout errors. When you save a layout the system tests if it is valid. You can invoke these tests on a layout before you save it. From the **File** menu, select **Validate**. You can check that your layout meets the following constraints:

- There must be at least one Question requiring data on each page.
- At least one Question must follow an indicator Question in the cursor visitation order.
- Objects cannot overlap.
- For Question Groups:
 - An indicator response field must be the first response field visited by a cursor in a Question Group.
 - A response field cannot be physically on the same row as a response field in a repeating Question Group, unless the two response fields belong to the same Question Group.
 - The cursor visitation order of a Question cannot place a response field between two response fields of a repeating Question Group unless all three response fields are part of the same Question Group.
 - All of the response fields of a repeating Question Group must be located on the same page.
 - If the **Prompt Position Type** field is "L" (left), the distance between response fields must be less than the number of lines specified by the lines per repeat for that repeating Question Group. For instance, if lines per repeat is 3, response fields cannot be more than 2 rows apart. If lines per repeat is 1, response fields must be on the same row. Note that the lines per repeat is one less than the LINES_PER_ROW specified in the database.
 - For repeating Question Groups where **Prompt Position Type** field is "A" (Above), lines per repeat must be one in number.

When you test validation or save, the system displays a log of any errors.

Saving a Layout

You can save your work even if a layout does not pass validation. In that case you receive a message that the layout is invalid. You can save your work, but you cannot move the DCM to production until you reopen the DCM and fix the errors.

Closing the Editor

You can exit by menu path or the **Door-close** button in the toolbar.

The system prevents you from modifying a DCM while the Layout Editor is open, and displays a warning if you attempt to access the Maintain DCMs form.

Laid Out Versus Not Laid Out Objects

If you make changes to a DCM after you have generated a default layout, the system automatically collects any new objects into a Not Laid Out area and activates the **Not**

Laid Out button in the toolbar. In fact if you have not generated a default layout and you open the DCM Layout Editor, all the DCM's objects are in the Not Laid Out area. In order for a layout to pass validation, you must move all of these objects from the Not Laid Out area to the Laid Out area:

1. Click the **Not Laid Out** button to open the Not Laid Out area.
2. Select one or more objects. (See [Selecting Objects](#).)
3. Cut (Ctrl+X) the objects. This action removes the selected objects from the current area and makes a copy of them in the Windows clipboard.
4. Click the **Laid Out** button in the toolbar to return to the layout area.
5. Paste (Ctrl+V) the objects.

Using Undo or Revert

To undo the last action, from the **Edit** menu, select **Undo** last action (where the system determines the type of action).

To return the state of your layout to the previous saved version without exiting the Layout Editor, select **Revert**.

Using Data Entry Forms in Test and Production Modes

You can test DCMs and layouts by doing data entry in test mode; see [Test Mode](#) for more information.

In this section:

- [Character-Based Layouts in Test and Production Modes](#)
- [Graphic Layouts in Test and Production Modes](#)

Character-Based Layouts in Test and Production Modes

If you are working with character layouts, you explicitly move the layout to production upon finalizing your DCM layout. The requirements for test and production data entry are:

- **Test mode** requires that DCMs have a valid layout. The status of the DCM can be either provisional or active. It does not matter if the DCM layout has been moved to production or not.
- **Production mode** also requires a valid layout. It is possible to move the layout for a provisional DCM to Production, but Oracle recommends moving layouts to production for active DCMs only.

To move a character-based DCM layout to Production:

1. From the **Definition** menu, select **DCMs**, then choose **DCMs**.
2. Select the Character Layout tab, then the DCM layout you want to move to Production.
3. From the **Special** menu, select **Move to Prod**. The system updates the **Production Generation Time** field. The DCM is then ready for data entry.

Graphic Layouts in Test and Production Modes

If you are using graphic layouts in RDC, you must embed your DCM layouts into a DCI Form; see [Creating Graphic Layouts](#). The requirements for test and production data entry are:

- **Test mode** requires a valid DCI Form. The status of the DCI can be either provisional or active. If both are available, the system uses the provisional version and its provisional, valid DCI Form version. If the DCI is active and both provisional and active DCI Form versions are available, the system uses the provisional version.
- **Production mode** requires an active DCI and an active DCI Form version.

Defining DCIs

A DCI includes one or more DCMs and corresponds to a CRF.

 **Note:**

A DCI's size directly affects how long it takes for end users to open and save CRFs in RDC Onsite. Avoid designing Oracle Clinical DCIs with more than 250 fields. Do not exceed 500 fields per DCI.

For more information, see:

- [Creating a DCI](#)
- [Adding DCMs to DCIs](#)
- [Defining DCM Pages](#)
- [Copying a DCI](#)
- [Marking a DCI as Critical for Source Data Verification](#)
- [Deleting a DCI](#)
- [Defining DCI Access at the Study Level](#)
- [Defining Custom Reviews Required at the DCI Level](#)

Creating a DCI

You can create DCIs directly from a DCM definition, by copying a DCI, or "manually" in the DCI window:

- When you make a DCM active, you have the option of creating a DCI that has the DCM subset name. You can choose the **Create** option to create the DCI as Provisional, or the **Create and Activate** option to create and activate the DCI.
- From either an Active or a Provisional DCM, choose **Create a Default DCI** from the **Special** menu to create a provisional DCM that has the DCM subset name.

 **Note:**

If a DCI has already been manually created for a Provisional DCM (and no other DCMs), the system allows you to activate the existing DCI or to create a new one, depending on whether the existing DCI's name or short name is the same as the DCM subset name.

If either the DCI's name or short name is the same as the DCM subset name, the message says, "Make corresponding DCI active also?" If you choose **Yes**, the system makes the DCI Active.

If they are different, the message says, "Create Default DCI." If you choose **Create**, the system creates a new provisional DCI. If you choose **Create and Activate**, the system creates a new active DCI.

- Copy a DCI from the Global Library or from a study; see [Copying a DCI](#).
- If the DCI has many DCMs, create a DCI in the Maintain Study DCIs window:
 1. Navigate to **Definition, DCIs**, then **DCIs**. The Maintain Study DCIs window opens.
 2. Refer to the online help for information on each field.

The **Collect Time?** setting determines if the DCI Time field appears in the Log-In form.

Adding DCMs to DCIs

In the Maintain Study DCIs window, click the **DCI Modules** button to reach the Data Collection Modules window. Copy either a Library DCM or a study DCM into the DCI using the Special menu.

Any changes you make apply only to the DCM in the context of this DCI, where it is also known as a *DCI Module*. Some DCI Module attributes, such as qualifying values and Clinical Planned Events, are available only in the context of a DCI.

Enter values as follows:

- **Disp Seq #** Controls the order in which the DCMs appear in the Received DCMs window.
- **Planned Event** If provided, forces the received DCM to be logged in for that clinical planned event and cannot be changed during login. You could use this feature to structure a DCI for a CRF containing multiple predefined visits.

If you leave the **Planned Event** field empty, at login you can create received DCMs for as many unique CPEs as there are on the CRFs. You might do this when you have a CRF that collects multiple CPEs, but allows the name of the event to be entered by the Investigator (instead of being preprinted).

- **Qualifying Value** If provided for a qualified DCM, forces the RDCM to be logged in for that particular qualifying value and cannot be changed during login. You could use this feature to structure a DCI for the particular variant of a CRF that collects information about the particular qualifying value. For instance, with two identical CRFs, one preprinted for information about the right eye and one for the left eye, you could create two DCMs, one with RIGHT as the qualifying value, and one with LEFT.

If you leave the **Qualifying Value** field empty, you can create received DCMs at each visit for as many unique qualifying values as are defined for the Qualifying Question. You

might do this when you have a CRF that collects a group of Questions about, for example, multiple culture sites, but that allows the name of the culture site to be entered by the Investigator (instead of being predefined on the CRF). At login, you could create a received DCM for each culture site named on the CRF.

- **Use DCI Date?** Determines whether the received DCM created at login automatically inherits the date and clinical planned event (and time if used) entered for the DCI. If so, the date and clinical planned event (and time) cannot be changed at the received DCM level.

 **Note:**

If you have chosen to collect the time, also check this box to collect the DCI Date.

- **Collect Time?** If checked, the **Time** field is enterable at login.
- **Optional?** Invokes a special behavior during login. You can only specify that a DCM is optional in a DCI if there is a single DCM in the DCI. At login, you can enter as many received DCMs for an optional DCM as are allowed by the study definition.
- **Start Page and End Page** Display only; see [Defining DCM Pages](#) .
- **DCM Available** The system checks this value if the DCM Graphic Layout is available. This field is read-only.
- **Show Qual Value?** Select this setting if the DCM has a Qualifying Question and you want to display the Question and text. If selected, the system includes the Qualifying Question in the DCM Header. If there is no Qualifying Question for the DCM, you cannot select this field. If there is a Qualifying Question, but no value for the qualifying value on the DCI Modules window, the system selects this field and prevents you from changing it.
- **Show Lab?** Select this setting if the DCM has a lab Question and you want to collect the name of the lab for the DCM. If selected, the system includes the lab Question in the DCM Header. You cannot select this field if there is no lab Question.
- **Visit Display** This field controls the display of **Visit Name** and **Subevent Number**. Here is a description of the available values:
 - **NAME ONLY** The system displays only the **Visit Name** on the DCI Form.
 - **NAME+SUB#** The system displays the **Visit Name** and **Subevent Number** as one field with a period separator (for instance, CHECKUP.1).
 - **NAME/SUB#** The system displays the **Visit Name** and **Subevent Number** as separate fields.
 - **HIDDEN** The system hides the **Visit Name** and **Subevent Number** in the DCM Header. If there is no defined CPE, or the **Use DCI Date** setting is deselected, you cannot select value HIDDEN for the **Visit Display**.
- **Show Blank?** If this box is selected, the system generates a **Blank** box in the DCM Header.

- **Show Data Comment?** If selected, the system includes the **Data Comment** in the DCM Header. You can only enter data through the Oracle Clinical Data Entry subsystem. Select this field if you want the data comment to show in the PDR.
- **Show Comment?** If selected, the system includes the **Comment** in the DCM Header. You can only enter data through the Oracle Clinical Data Entry subsystem. Select this field if you want the comment to show in the PDR.

Defining DCM Pages

Use this window only if you are using the Page Tracking feature for a paper-based trial. See [Using CRF Page Tracking](#).

If a DCI spans multiple pages, specify which questions go on which page by defining the final Question on each page in the Data Collection Module Pages window.

From the **DCI Modules** page, select the DCM and click the **DCI Modules Pages** button.

Copying a DCI

If there is a DCI in the Global Library or in another study that suits your needs, you can copy it into the current study—rather than creating it from scratch—by selecting one of two copy options. To copy a Global Library DCI:

1. Navigate to **Definition**, then **DCIs**, and **DCIs**.
2. Select either **Copy Lib DCI** or **Copy Study** from the **Special** menu. Either the Locate Library DCIs or the Locate Study DCIs form appears, according to your selection.
3. Query for the DCI you want to copy.
4. Click either:
 - **OK** button to perform the copy
 - **Cancel** to return to the Maintain Study DCIs form without making a copy

Follow this procedure to copy an existing study DCI:

1. Navigate to **Definition**, then **DCIs**, and **DCIs**.
2. Select **Copy Study** from the **Special** menu. The Locate Study DCIs form appears.
3. Enter the study name to copy from.
4. Click the **OK** button to perform the copy, or **Cancel** to return to the Maintain Study DCIs form without making a copy.

Marking a DCI as Critical for Source Data Verification

Critical CRFs must be verified against source data for all patients. You can designate a CRF as critical by marking the DCI on which it is based as critical, either in the global library or in a study.

- To mark a DCI as critical at the study level, go to **Definition**, **DCIs** and then select **DCIs**. Mark the **SDV?** check box for the critical DCI.
- To define the DCI critical at the Global Library level, select **Glib**, then **DCIs DCMs Procedures**, and then select **DCIs**. Mark the **SDV?** check box for the critical DCI.

See [Setting Up Partial Source Data Verification](#) for more information.

Deleting a DCI

Under some circumstances you can delete a DCI by selecting it and then selecting Delete Record from the Data menu.

The system gives a warning if you delete a DCI that is associated with a Rule in an enhanced DCI Book for the study. If you proceed with the deletion you must delete the Rule or delete the DCI from the Rule before you can validate the DCI Book.

Defining DCI Access at the Study Level

You can define access to particular DCIs at the study level by overriding the default user role access settings. Here are two example situations:

- A local laboratory's personnel require RDC access so they can enter their lab results directly. However, lab personnel must only have access to the specific laboratory CRF and no others. An administrator can create a user role for the local laboratory personnel and you can restrict access for that role to the DCIs relevant to the lab's CRFs.
- The Patient Data Report includes only data in DCIs to which the user who runs the report has access. If the role of the user who runs the report does not have access to a particular DCI, the report does not include any data for the patient for that DCI.

The default user role settings, defined at the installation level, are either completely restricted or completely unrestricted. At the study level, you can override these settings and for any user role, specify either a list of DCIs that this role can see (an Inclusion list), or cannot see (an Exclusion list).

Note:

If a user does not have access to a study based on the defined study-level or site-level access, the study DCI access cannot provide the user with access to the DCI or to the study.

To specify DCI access at the study level:

1. Create the default user roles. (See the *Oracle Clinical Remote Data Capture Administrator's Guide* for more information.)
2. In Oracle Clinical, navigate to **Definition**, then **DCIs**, and then **DCI Access**. (Users with Maintain privileges can see both the DCI Access and Query DCI Access navigation options. Users with Browse privileges can see the Query DCI Access option.)
3. Choose a study and click OK. The DCI Access window opens with the current study displayed in the title bar. (You can click the **Change Study** button at the bottom of the DCI Access window to select another study. You can also select **Special**, then **Select Study**.)
4. In the **User Role** section, select the first empty row and enter a valid user role name. You can also open the list of user role values, and choose a user role.

5. In the **DCI List Type** section, open the list of values in the same row as the User Role row you selected in the previous step. You have two choices:
 - Select **EXCLUSION** to allow the user to see everything except the listed DCIs.
 - Select **INCLUSION** to allow the user to see only the listed DCIs.

The administrator sets default value for the DCI List Type field.

6. In the **DCI Name** section, enter a DCI name, or select a DCI from the list of values. The list of values includes all DCIs defined in the study.

You can also leave the DCI Name field empty. An empty DCI list is interpreted differently:

- **EXCLUSION**: The user has access to all DCIs. This approach may be taken when the default DCI access for the role is Restricted, but for a specific study the user role has access to all DCIs.
- **INCLUSION**: The user has access to no DCIs for this study. This approach may be taken when the default DCI access for the role is Unrestricted, but for a specific study the user role has access to no DCIs.

Note that for individual users, you can accomplish the same access with study security.

7. In the **Access** section, select the type of access to allow for this DCI. Your options vary depending on whether you are including or excluding access to the DCI.

If the DCI List Type is set to INCLUSION you can select:

- **Default study/site privileges**: Indicates that the DCI is accessed according to the user's study or site privileges.
- **Browse**: Indicates that the user role can open and view the DCI only in browse mode.

If the DCI List Type is set to EXCLUSION, you can select:

- **None**: The user role cannot access the DCI.
- **Browse**: The user role can open and view the DCI only in browse mode.

8. Continue to define the DCIs that this user role can access (include and exclude).
9. Save.

Defining Custom Reviews Required at the DCI Level

1. Expand the **Definition** node, then expand **DCIs**, and then select **DCIs**.
2. Select or switch to the study to which the DCI belongs.
3. Select the **Review Requirements** tab.

Tip:

The checkbox **Allow CRF status change to RN when review is required?** at the top of the tab is read-only and indicates the inherited installation or study setting, if one is defined. You can specify a different setting for individual custom review types in step 7.

4. Query in the DCI Name field for the DCI for which you want to set the custom review requirements.

5. Click in the **Custom Reviews Required** field for that DCI to open the Reviews Required dialog box.
6. Select (highlight) all the review types that should be required for this DCI.

 **Tip:**

Clicking a single review type drops the current selection and selects only that review type. Use Shift+click, Ctrl+click, or the controls below the list to select or deselect multiple review types.

7. Make sure that **Allow Override RN Status** is checked for only those selected (required) review types for which you want to allow reviewers to set the review status to Review not Required (RN).
8. Click **Save**.

For information about custom review types, see chapter "Configuring Custom Review Types" in the *Oracle Clinical Remote Data Capture Administrator's Guide*.

10

Defining DCI Books

In this chapter:

- [Traditional and Enhanced DCI Books](#)
- [Defining an Enhanced DCI Book](#)
- [Calculating DCI and Interval Expectedness for Patients](#)
- [Data Entry and Expectedness](#)
- [Assigning Patients to a DCI Book](#)
- [Defining a Traditional DCI Book](#)
- [Using CRF Page Tracking](#)

Traditional and Enhanced DCI Books

You can organize DCIs, which generally correspond to CRFs, into a casebook and assign them to visits (Clinical Planned Events, or CPEs) by creating a DCI Book. DCI Books are required if you are using Oracle Remote Data Capture (RDC) and can be used for Oracle Clinical data entry too.

Before you can define a DCI Book you must define Intervals and Clinical Planned Events (CPEs) as well as Questions, Question Groups, DCMs, and DCIs; see [Treatments and Schedules Questions and Question Groups](#) and [DCMs and DCIs](#)

Oracle Clinical provides two types of DCI Books for use in different types of clinical trials:

- [DCI Books](#)
- [Enhanced DCI Books](#)

DCI Books

The traditional DCI Book is suitable for paper-based clinical trials for which you want to use the Oracle Clinical Page Tracking feature. This feature helps ensure that all paper CRFs are collected and their data entered. If your study uses the Page Tracking feature, you must use the original DCI Book type. See [Defining a Traditional DCI Book](#) and [Using CRF Page Tracking](#) for more information.

Enhanced DCI Books

Enhanced DCI Books offer:

- [Support for Flexible Studies](#)
- [Additional Features](#) You can use enhanced DCI Books to take advantage of these features even if your study does not have a flexible design, as long as you do not use Page Tracking.

Page Tracking is not supported in Enhanced DCI Books.

- [Support for Flexible Studies](#)
- [Additional Features](#)

Support for Flexible Studies

The enhanced DCI Book is suitable for flexible trial designs such as studies with multiple treatment arms and cohorts (groups of patients) with different assessments. In these trials, patients receive different assessments based on data collected during the trial. For example, a flexible study design for an oncological indication might require different disease states to have different schedules and assessments, with patients with more rapidly progressing disease being evaluated and dosed more frequently and patients with less rapidly progressing disease states being evaluated less frequently.

Open-ended cycling trials are also supported. You must define an adequate number of CPEs and Intervals and add them to the Enhanced DCI Book. You can add more during the course of the trial if necessary.

When you use an Enhanced DCI Book, the following flexible study features are supported:

- **Data-Triggered Rules:** You specify data points that determine which CRFs, CPEs, and Intervals are required, or *expected*, for each patient.
- **Data Expectedness Calculation for Each Patient:** As soon as data is saved for a patient at one of these predefined points (or the system completes a batch job that may impact data expectedness for a patient), the system recalculates which CRFs and Intervals are expected for the patient, based on the defined rules.
- **Display of CRFs in RDC Onsite:** For each patient, the Casebooks page displays only the CRFs that are expected, and as the user enters data and clicks Next, the next CRF expected during the CPE for that particular patient opens. Any CRFs that have been entered but are no longer expected are still displayed.

Note:

In other manual Oracle Clinical and RDC data entry interfaces all CRFs included in the DCI Book are available for data entry, not just those expected for the patient.

- **Validation:** You can define an Enhanced DCI Book in any order and then run a validation utility that checks rule definitions against DCI, CPE, and Interval definitions and identifies any invalid conditions so that you can fix them; see [Validating an Enhanced DCI Book](#).

Additional Features

Enhanced DCI Books include other features that are helpful whether your study is defined as flexible or not:

- **Navigation:** You can create, maintain and view the DCI Book pages for a selected CPE without navigating through all the pages for the book; see [Navigator: Building an Enhanced DCI Book](#).
- **Copying:** You can copy all the DCI Book pages for a single CPE and insert them into another CPE.

- **Deleting pages:** You can delete all the DCI Book pages for a single CPE at once.
- **Renumbering pages:** You can run a utility to automatically renumber any range of Start Page numbers in CPE and DCI order, using prefixes and/or suffixes you specify. You can insert or delete a DCI in the middle of the study, or assign DCIs to CPEs in any order, and then run the utility to fix the Start Page numbers.; see [Numbering and Renumbering Pages](#).
- **Validation:** The validation utility checks Start Page numbers and gives a warning if they are nonsequential. You can choose to continue with nonsequential page numbers or you can use the renumbering utility.

Defining an Enhanced DCI Book

Oracle recommends defining a single DCI Book in flexible studies; see [Using a Single DCI Book](#).

To open the Enhanced DCI Book window, select **Definition**, then **DCIs**, and then **Enhanced DCI Books**.

The Maintain Enhanced DCI Books window includes the following **Special** menu items:

- **Copy DCI Book:** You can copy only Active DCI Books. The system creates the new Book with a status of Provisional. All rules, Intervals, CPEs, and DCI Book pages are included in the copy, and the new book has the name COPYOF *original_Book's_name*. You must change the name before you save the new Book. The new Book is in the same study as the original Book; you cannot copy a Book from one study to another.
- **Validate DCI Book;** see [Validating an Enhanced DCI Book](#)
- **View Validation Results;** see [Viewing Validation Warnings and Errors](#)
- **Calc Expectedness in Test;** see [When and How to Run Expectedness Calculation Manually](#)
- **Calc Expectedness in Prod;** see [When and How to Run Expectedness Calculation Manually](#)
- **Re-sequence DCI Book** reorders display numbers that are not in ascending CPE CPE Number order or have gaps. This option is available only when display numbers are out of sequence. This happens only in rare circumstances (either the Book was not created in the Enhanced DCI Book window or CPE numbers were changed after the Book was created) but if it does, the Book becomes unusable; no data can be entered against it. In addition, RDC Onsite displays (Invalid) after the book name in the Casebook column and does not include the book in the Assigned Book list of values in Patient Search.

If the Book's display numbers are not sequential, you receive a warning when you try to open your DCI Book in the **Navigator** window. You cannot make any changes to the book until you run the **Re-sequence DCI Book** job from the Special menu in the main Enhanced DCI Book window. The job does not affect Start Page numbers.

For more information, see:

- [Defining Book Properties](#)
- [Navigator: Building an Enhanced DCI Book](#)
- [DCI Book Pages: Assigning DCIs to CPEs](#)
- [Copying Pages for a CPE](#)
- [Deleting Pages from a CPE](#)

- [Numbering and Renumbering Pages](#)
- [Defining Interval Rules](#)
- [Defining DCI Rules](#)
- [Validating an Enhanced DCI Book](#)
- [Defining DCI Book Constraints](#)

Defining Book Properties

Name	Status	Description	Default?	Validation Status	St
RULESBOOK	P	RULESBOOK	<input checked="" type="checkbox"/>	NOT APPLICABLE	
	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>		<input type="checkbox"/>		

Exit Save Change Study Navigator Constraints Interval Rules DCI Rules

To define an Enhanced DCI Book, enter values in the following fields:

Note:

Only one person at a time can work on an Enhanced DCI Book. The Book is locked as soon as a user initiates any action that could modify the Book, and it remains locked until the user saves or cancels. You receive an error if you try to work on a Book that has been locked for another user.

Name - Enter a name for the DCI Book; up to 30 characters.

Status - Possible values for this field are P (Provisional), A (Active), and R (Retired):

- **Provisional (P):** You can modify the DCI Book only if its status is Provisional. You can set the status of an Active DCI Book back to Provisional. You can enter test data using an Active DCI Book or a Provisional DCI Book with a validation status of Success or Warning.

- **Active (A):** You can use only Active DCI Books for entering production data. When you set the status to Active and save, the system runs the validation job behind the scenes and calculates the DCI Book's validation status. If it is Success, Warning, or Not Applicable (for nonflexible studies) the activation succeeds.
- **Retired (R):** Set the status to R for DCI Book definitions you no longer want to use but cannot delete because they are associated with data. You are prompted for a retirement reason.

Description - (Optional) Enter a description for the DCI Book; free-form text, 200 characters maximum.

Default? - Check if this is the default DCI Book for use in the current study. Only one DCI Book can be the default one for a study.

Oracle Clinical and RDC use this setting to determine which Book to display by default.

Validation Status - (Display Only) The possible validation statuses are:

- **Not Applicable.** The system applies this status automatically to DCI Books used in nonflexible studies. You can set DCI Books with this validation status to Active.
- **Pending.** The system applies this status automatically to DCI Books used for flexible studies that have never been validated or that have changed since the last validation job was run. You can set DCI Books with this validation status to Active. The system then runs the validation job and the activation succeeds if there are no errors.
- **Success.** The validation job applies this status to DCI Books when the job completes with no warnings or errors. You can set DCI Books with this validation status to Active.
- **Warning.** The validation job applies this status to DCI Books when the job completes with one or more warnings but no errors. You can set DCI Books with this validation status to Active.
- **Error.** The validation job applies this status to DCI Books when the job completes with one or more errors. The job may have generated warnings as well. You cannot set DCI Books with this validation status to Active.

For information about the validation utility, see [Validating an Enhanced DCI Book](#).

Status Comment - This is a free text field, up to 200 characters.

Retirement Reason - When you change the status of the DCI Book to R you are prompted for a comment. This field displays the most recent comment.

Navigator: Building an Enhanced DCI Book

The DCI Book Navigator window provides summary information about every Clinical Planned Event (CPE) defined for the study, so that you can track your DCI Book definition progress. To reach it, click the Navigator button in the main Enhanced DCI Books window.

Use the **Navigator** window to:

- Check for Intervals and CPEs with no DCIs assigned yet
- Select a CPE to display its individual pages (DCIs) in the DCI Book Page Details window
- Select a CPE to add, delete, or modify individual pages in the DCI Book Page Details window
- Select a CPE to copy individual pages in the DCI Book Page Details window to another CPE in the book

Use the **Special** menu to:

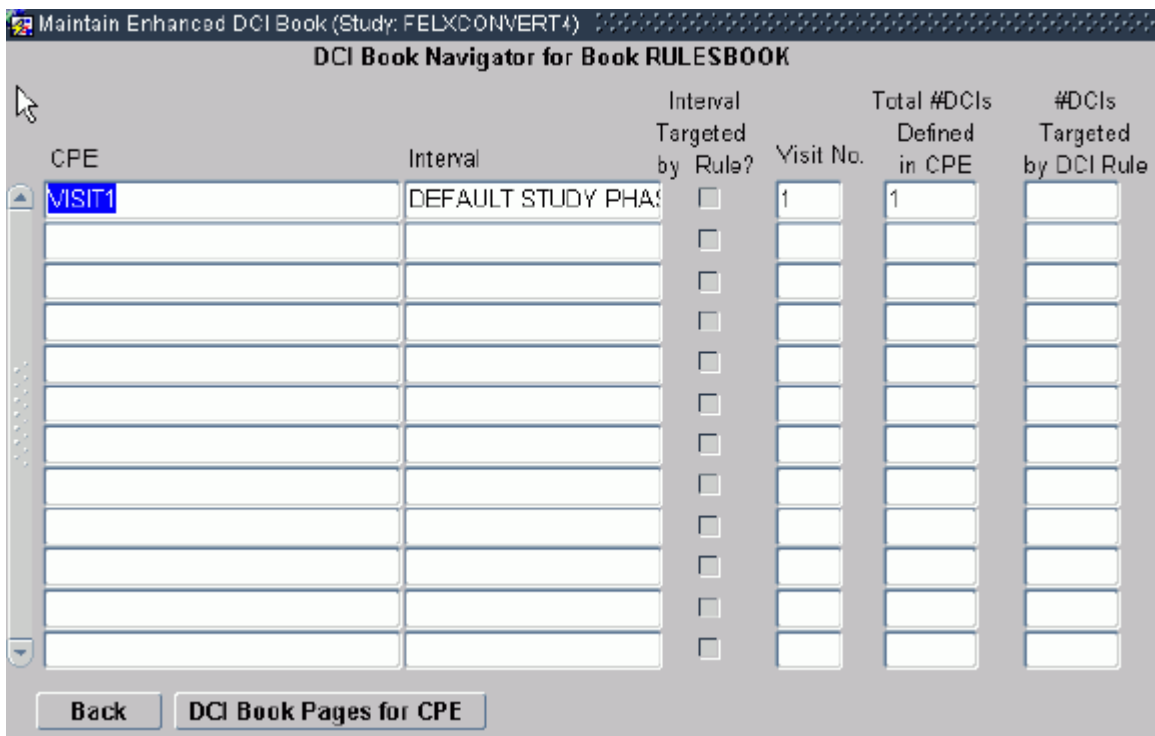
- **Copy Book Pages for CPE**; see [Copying Pages for a CPE](#)
- **Delete Book Pages for CPE**; see [Deleting Pages from a CPE](#)
- **Renumber Book Pages**; see [Renumbering Start Pages](#)

You can work in any order. For example, you can define DCI Book Pages for the Termination CPE before the Screening CPE. In a flexible study, you can define DCI Book Pages before you define rules, or you can define rules first, or alternate between the two tasks. You can run the validation utility (available from the Special menu of the Maintain Enhanced DCI Books window) or the renumbering utility for a Provisional Book at any time.

You can make changes only if the Book's status is Provisional.

 **Note:**

All CPEs are always displayed in Navigator window. However, if none of the CPEs in an Interval have any DCIs defined, the system treats the Interval as if it does not exist; validation fails if such an Interval is the target of a rule.



CPE	Interval	Interval Targeted by Rule?	Visit No.	Total #DCIs Defined in CPE	#DCIs Targeted by DCI Rule
VISIT1	DEFAULT STUDY PHAS	<input type="checkbox"/>	1	1	
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			

The DCI Book Navigator window displays the following information. Records are displayed in ascending visit number order.

CPE - The system displays all the CPEs defined in the Maintain Clinical Planned Events window for the current study, including CPEs defined as Optional, in CPE Number order. If any CPEs have been added since the DCI Book was made Active, they are also displayed.

Interval - The system displays all Intervals—Phases, Periods, or Subperiods—defined in the Maintain Study Schedule window with one or more CPEs assigned directly to them in the current study. Intervals with no CPEs directly assigned to them are not displayed. For example, if a Phase contains several Periods that contain CPEs, but no CPEs are directly assigned to the Phase, the Phase is not displayed but the Periods are. If any Intervals with CPEs have been added since the DCI Book was made Active, they are also displayed.

Interval Targeted by Rule? - If checked, the Interval is the target of a DCI Book rule. Intervals that are the target of a rule are *conditional*; they are expected for a patient only if the patient's data meets criteria defined in a rule; see [Defining Interval Rules](#).

Visit No. - The system displays the Visit Number defined in the Clinical Planned Events window.

Total # DCIs Defined in CPE - The system displays the number of DCIs currently assigned to the CPE. To see the DCIs, select the CPE and click the button DCI Book Pages for CPE.

DCIs Targeted by DCI Rule - The system displays the number of DCIs currently assigned to the CPE that are the target of a DCI Book rule. DCIs that are the target of a rule are *conditional*; they are expected for a patient only if the patient's data meets criteria defined in a rule; see [Defining DCI Rules](#).

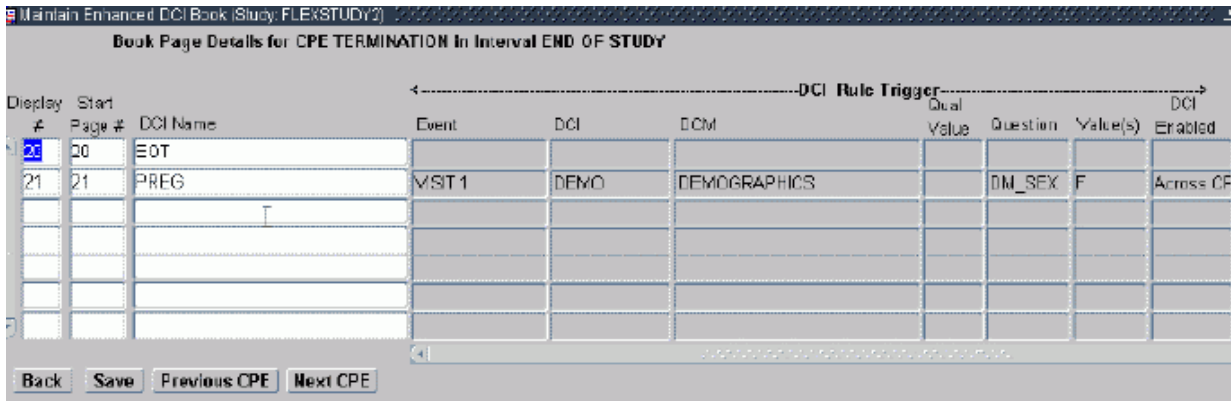
DCI Book Pages: Assigning DCIs to CPEs

To make it possible to collect a DCI, or Book Page, at a CPE, you must do one of the following:

- Assign the DCI to a CPE by adding it here.
 - If the DCI is not the target of a rule, it is *unconditional* and expected for all patients for whom the Interval in which the DCI is collected is expected.
 - If the DCI is the target of a rule, it is *conditional* and expected for a patient only when his or her data fulfill the condition of the rule.
- Make the DCI available as needed on an *unplanned* basis in the DCI Constraints window; this approach may be appropriate for adverse events and concomitant medications, for example. See [Defining DCI Book Constraints](#) .

You can view, insert, and delete DCIs in a CPE. DCIs should appear in the order in which they should be collected. Add them in this order or change their Display Numbers to change their order within the CPE. DCIs that are the target of a rule must come after the trigger DCI.

You can make changes only if the DCI Book's status is Provisional. You can insert DCIs even if there is already production data entered against the Book. If the insertion and renumbering of pages will change the page number of a page against which data has already been entered, you receive a warning. The page number displayed for the collected data changes.



For more information, see:

- [DCI Book Page Information](#)
- [Rule Trigger Information](#)

DCI Book Page Information

The window has the following active fields:

Display # - Oracle RDC Onsite uses the display number to display CRFs in its user interface. Display numbers are generated according to CPE Visit Number; within a CPE you can change the display order of DCIs. DCIs should be displayed in the order in which they should be collected.

Start Page # - This is the page number that appears in the data entry user interface and in the Patient Data Report. In non-Page Tracking studies, every DCI is considered to be one page.

You can specify an alphanumeric prefix and/or suffix for Start Page numbers. You may want to use meaningful abbreviations for the Interval and CPE, for example, rather than have simple numeric page numbers, if you do not want page number gaps in the Patient Data Report; see [Sample Numbering Schemes](#). The maximum length of the field is 15 characters.

The system automatically increments the first number (reading from left to right) in the Start Page number by one integer when you insert or explicitly renumber or copy pages. If a Start Page number has a whole number followed by a decimal point followed by a number (for example, 12.1) the system does not increment the decimal portion, which it treats as a static suffix. Special characters other than a period/full stop (.) are not allowed.

If you do not specify prefixes or suffixes for any pages in the Book, the start page number is the same as the display number throughout the Book.

See [Numbering and Renumbering Pages](#) for more information.

DCI Name - The name of the DCI—equivalent to a CRF in RDC—assigned to the CPE. To insert a DCI, select it from the list of values, which contains all Provisional and Active DCIs defined for the study. A particular DCI can be included only once in a CPE.

Rule Trigger Information

If the DCI is the target of a DCI rule, the system displays information about the rule trigger; see [Defining a DCI Rule Trigger](#) for more information:

Event - The CPE or CPEs that contain(s) the trigger DCI.

DCI - The trigger DCI.

DCM - The trigger DCM.

Qual Value - The DCM Qualifying Value, if any.

Question - The trigger Question.

Value(s) - The Question response values that trigger the rule.

DCI Enabled indicates the scope of the rule's effect:

- **Within CPE** The rule triggers the target DCI only when the trigger DCI is collected at the same CPE as the trigger DCI.
- **Across CPEs** The rule triggers the target DCI in every CPE where it is collected from the CPE at which the trigger DCI is collected onwards.

Copying Pages for a CPE

You can copy all the Book pages (DCIs) for a selected CPE into a target CPE in the same Book. The system automatically adjusts the Display numbers in the Book, but you may need to use the Renumber Book Pages job to fix Start Page numbers; see [Numbering and Renumbering Pages](#). You can only copy pages into a CPE that has no pages defined.

To copy Book pages:

1. In the **DCI Book Navigator** window, select the CPE whose pages you want to copy and select **Copy Book Pages** for CPE from the **Special** menu. A pop-up appears.
2. From the **Target CPE** drop-down, select the CPE into which you want to paste the pages. The name of the Interval that contains the target CPE is displayed in the **Target Interval** field.
3. In the **Page Number Seed** field, enter the seed you want to use, if any, for the pages in the target CPE. This seed, or lack thereof, overrides any existing seed used in the source CPE. If you want to use the same seed as in the source CPE, you must specify it. This is an optional field. See [How Oracle Clinical Page Numbering Works](#).

If you leave the Page Number Seed field empty, the system supplies whole numbers as page numbers, beginning with the number one (1) greater than the highest Start Page number without either a prefix or suffix already defined in the Book.

 **Note:**

Copying pages does not change any rules defined for the Book. If any rules become invalid as a result of the copy operation, the validation job will report it.

Deleting Pages from a CPE

The Delete Pages feature deletes all the Book pages (DCIs) from a selected CPE. You can delete pages only from a Provisional DCI Book. If either test or production data has been entered against any of the pages you delete, the data remains in the system. No further data can be entered against the deleted pages.

If you delete a DCI Book page where the DCI is a trigger DCI or the only target of a rule, the system deletes the rule along with the DCI Book Page. If the DCI is one of many target DCIs in a rule, the system deletes it from the rule's list of targets.

If you wish, you can run the Renumber Book Pages job to remove any gaps in Start Page numbers that result from the deletion.

To delete Book pages: In the **DCI Book Navigator** window, select the CPE whose pages you want to delete and select **Delete Book Pages for CPE** from the **Special** menu.

Numbering and Renumbering Pages

In Enhanced DCI Books you can enter DCI Book pages in any order and call a utility to renumber them at any time. The system calls the same utility when you copy CPE pages and paste them into a new CPE.

In addition, the system uses the utility to check Start Page numbers at the following times:

- **Book Validation.** When you run the Validate DCI Book job, the system calls the numbering utility to check if Start Page numbers are sequential and without gaps and returns a warning if they are not sequential. You can choose to have nonsequential Start Page numbers and gaps.
- **Book Activation.** When you activate a Book, if the Book does not already have a validation status of Success, the system calls the validation job, which calls the numbering utility. The activation succeeds even if Start Page numbers are not sequential, but sets a validation status of Warning. If you want sequential numbers, you can set the status back to Provisional and run the numbering utility manually (select Renumber Book Pages from the Special menu in the DCI Book Navigator).

Note:

The information in this section applies only to Start Page numbers. For information on Display numbers, see Display# in [DCI Book Page Information](#).

 **Note:**

When changes are made to several DCI Books that impact their rules and the Books are all activated simultaneously without prior validation, if one of the Books fails validation, then the Expectedness calculation for any of the modified books can fail with the following error:

```
1212500: An error occurred while performing expectedness 1212600: ORA:20000
```

To avoid this problem, run Validate DCI Book for each book, so that Validation Status is SUCCESS or WARNING, and then activate the Book(s).

For more information, see:

- [How Oracle Clinical Page Numbering Works](#)
- [Renumbering Start Pages](#)
- [Sample Numbering Schemes](#)

How Oracle Clinical Page Numbering Works

Start Page number default functionality is the same in Enhanced DCI Books as in traditional DCI Books. The numbering utility is available only in Enhanced DCI Books and only in Books with a status of Provisional.

Page Number Seeds

DCI Book Start Page numbers can include a prefix and/or a suffix; that is, a character or string of characters before and/or after a number. The combination of the prefix (if any) and suffix (if any) and the number is called the *seed*.

By default, when you add a page to the Book the system uses the same seed as the previous page and increments the first number encountered when reading from left to right. If there is a number followed by a decimal point and another number (for example, 12.1), the system treats the decimal portion as a static suffix and does not increment it.

If you do not include a whole number in the seed, the system treats the letters you enter as a prefix and appends the number 1 to it, then increments by one for subsequent pages with same prefix. If you want a suffix or both a prefix and a suffix, you must include a whole number in the seed in the correct position.

Examples:

- **1** — There is no prefix or suffix. Next page by default: **2**.
- **A1** — "A" is the prefix and there is no suffix. Next page by default: **A2**.
- **A1.1** — "A" is the prefix and ".1" is the suffix. Next page by default: **A2.1**.
- **1.1A** — There is no prefix and ".1A" is the suffix. Next page by default: **2.1A**.

See [Sample Numbering Schemes](#) and the following note for issues to take into consideration when creating Start Page numbers.

 **Note:**

When you generate the Patient Data Report in RDC, if any DCI/CRF does not fit on a single PDF page, the system automatically appends.x to the DCI's Start Page number.

For example, if a DCI's Start Page number is 600, and the DCI spans three PDF pages, the system numbers the PDR pages 600.1, 600.2, and 600.3. Or, if a DCI's Start Page number is A600.4, the system numbers the PDR pages A600.4.1, A600.4.2, A600.4.3.

 **Note:**

Start Page numbers cannot contain a special character other than a period/full stop (.).

Renumbering Start Pages

The **Renumber Book Pages** utility renumbers pages within a range of Display numbers you specify. Within that range, it groups all pages with the same prefix/suffix combination and renumbers within each group sequentially by Display number, beginning with the lowest number found in the group. (Display numbers are determined by CPE number order.)

You can invoke the **Renumber Book Pages** job from the **Special** menu of the **DCI Book Navigator** window. A pop-up window opens. Enter fields as follows:

1. **Start Page Display #** Click in the field to bring up the list of values, then select the Display number at which you want to start renumber pages. If you want to renumber the entire DCI Book, select the first item in the list of values.

The list of values displays the current Display number, Start Page number, CPE number and CPE name for each page. You can search for the Display number.

2. **End Page Display #** Click in the field to bring up the list of values, then select the Display number at which you want to stop renumbering pages. If you want to renumber the entire DCI Book, select the last number in the list of values.

The list of values displays the current Display number, Start Page number, CPE number and CPE name for each page. You can search for the Display number.

3. Click **Renumber** to invoke the utility or click **Cancel**.

Sample Numbering Schemes

You can design your own numbering scheme within the restrictions of [How Oracle Clinical Page Numbering Works](#). Two possible number schemes are described below.

Sequential Integers

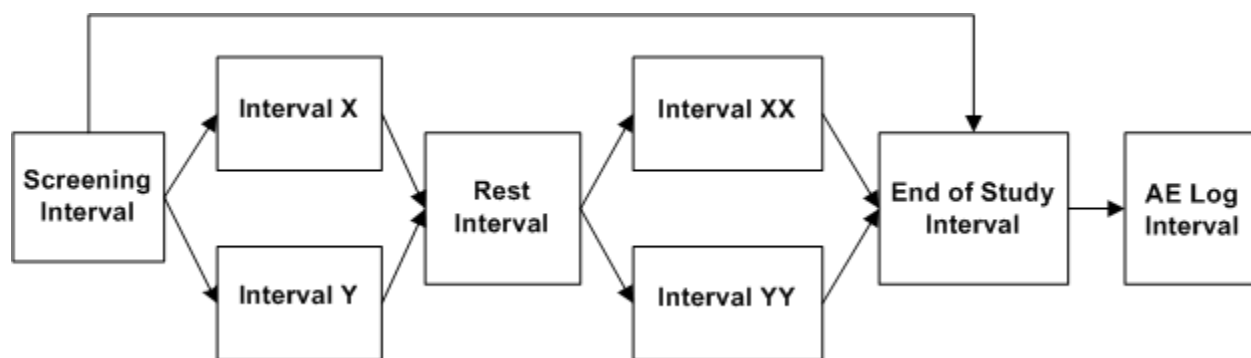
If you use Start Page numbers that are only whole numbers, with no alphanumeric prefix or suffix, by default each DCI's Start Page number is the same as its Display number. Using simple sequential numbers can have drawbacks:

- If you need to add a DCI in an ongoing study due to a protocol amendment, for example, you may not want the new pages to be inserted sequentially because page numbers associated with data already entered for subsequent pages will change. If you want to avoid this, use a suffix such as a, b, c for the newly inserted pages so that subsequent DCI Book pages keep the same numbers. You must insert these suffixes manually; the Renumber utility does not increment suffixes.
- If you use a sequential integer number scheme in a flexible study, where all patients normally do not complete all the DCIs defined in the DCI Book, there will be gaps in each patient's page numbers; see next section.

Interval and CPE Prefixes

In a flexible study, when you know that all pages are not expected for any patient—for example in a study with multiple treatment arms such as the one shown in [Figure 10-1](#)—you may prefer to use prefixes to give context to the page number.

Figure 10-1 Multiple Paths



In the study design shown in [Figure 10-1](#), some patients will complete Intervals X and XX and other patients will complete Intervals Y and YY, but no patient will complete all those Intervals. If you use sequential integer page numbering and if in the study schedule Interval X is defined before Interval Y and Interval XX is defined before Interval YY, you would have pages numbers such as:

- Pages 1-50, Screening Interval
- Pages 51-100, Interval X
- Pages 101-130, Interval Y
- Pages 131-146, Rest Interval
- Pages 147-200, Interval XX
- Pages 201-250, Interval YY
- Pages 251-252, End of Study Interval
- Pages 253-260, AE Log Interval

Pages 101-130 and 201-250 would not be expected for patients who complete Intervals X and XX, and pages 51-100 and 147-200 would not be expected for patients who complete Intervals Y and YY. Therefore each patient's Patient Data Report would have big gaps in page numbers.

You may prefer to use page numbers that represent the Intervals and CPEs by adding character prefixes to the page numbers. Do not use numbers to represent the Interval or CPE

name (or at least not in the prefix) because the system will increment the Interval or CPE number instead of the page number (it increments the first number in left-to-right order). You may want to name your CPEs in the study schedule with this in mind.

For example:

- Give all CPEs a name that starts with an abbreviation of the Interval name—such as SC for Screening, X for Interval X, RST for Rest, END for End, and AE for AE Log—followed by an abbreviation for the CPE; for the purposes of this example say the first Visit in each Interval is A, the second one is B, and so on. (A meaningful abbreviation would be better in case you need to add a visit later.)
- Use these abbreviations as the prefix in the Start Page seed, so for the first CPE in the Screening Interval the Start Page seed prefix would be SC_A. You must enter this Start Page prefix manually in Book Page Details once per CPE.
- Give the first DCI in each CPE the number one. The system automatically increments the number for each successive DCI in the CPE.

The first DCI Start Pages for a patient following Treatment Arm X might then be SC_A1, SC_A2, SC_A3, SC_B1, SC_B2, SC_C1, SC_C2, SC_C3, X_A1, X_A2, X_A3, X_A4, X_A5, X_B1, X_B2, X_B3, X_C1, X_C2, X_C3, RST_A1, XX_A1....

Defining Interval Rules

Interval rules can be of two types, or actions:

- **Enable** rules make the target Interval (or Intervals), all the CPEs within the Interval, and all unconditional DCIs collected at those CPEs, expected for a patient if the patient's data meets the criteria defined in the trigger.
- **Bypass To** rules can have only one Interval as a target. If a patient's data meets the criteria defined in the trigger, the target Interval becomes expected for the patient and all intervening Intervals, CPEs and DCIs are not expected.

An Interval can be the target of multiple rules. This is necessary, for example, when you want to bypass to the Interval in some cases but also use it for patients who do not trigger the bypass. See [Defining Multiple Paths Through a Trial](#) for an example of this; the End of Study Interval is the target of a **Bypass To** rule from the Screening Interval and it is also the target of an **Enable** rule from a DCI that is in the last CPE of both treatment paths.

The Book cannot be validated if there are duplicate rules. An Interval rule is considered a duplicate when the trigger DCI, trigger DCM (Qualifying Value), trigger Question (including null), target action and target Interval all have the same value as in another rule. The trigger Question values and "CPE(s) at which Trigger DCI is Defined" are not evaluated.

Bypass To Interval rules take precedence over all other types of rules if there is a conflict.

You are not required to define Interval rules in a flexible study. If you have no Interval rules defined in the DCI Book, all Intervals are expected for all patients who are assigned to that Book. Some DCIs in CPEs in the Interval may be conditional if you define DCI rules.

You can define rules only for studies defined as flexible in Clinical Study States (under Conduct, then Security).

 **Note:**

To see all the rule fields at the same time, expand the window by dragging the edge to the right. This does not work the first time you open the window during your session, but if you close the window and reopen it you can then expand it.

For more information, see:

- [Defining an Interval Rule Trigger](#)
- [Defining an Interval Rule Target](#)

Defining an Interval Rule Trigger

The trigger portion of an Interval rule specifies the data condition required to trigger the rule, making the target Interval or Intervals expected for the patient whose data satisfies the condition. Interval rules with an Action of **Bypass To** also make all Intervals, CPEs and DCIs between the trigger DCI and the target Interval *non-expected* for the patient; see [Defining an Interval Rule Target](#).

DCI	DCM / (Qual Value)	Question	CPE(s) at which Trigger DCI is Defined
DEMO	[Any Data]		VISIT 1
RAND	RANDOMIZATIO	TREATMENT_G	VISIT 3
RAND	RANDOMIZATIO	TREATMENT_G	VISIT 3
CONT	CONTINUE	CONTINUE	VISIT B3,VISIT A3

To define an Interval rule trigger, enter values in the following fields:

DCI - From the list of values, select the DCI that contains the DCM or Question whose value you want to serve as the trigger for the rule.

The list of values includes the long and short names of all Provisional and Active DCIs with DCMs defined for the current study, whether they are currently defined in the DCI Book or

not. If they are not yet included in the DCI Book, you must add them in the DCI Book Pages window before you can activate the Book.

DCM/Qual Value - From the list of values, select the DCM that contains the Question whose value you want to serve as the trigger for the rule, or select [Any Data] if you want to trigger the rule when data is entered for any Question in the DCI. See "Using the Any Data Trigger" in [Rule Triggers](#) for further information.

The list of values includes all DCMs defined for the selected DCI in the current study that meet the following conditions:

- The DCM must include at least one non-repeating Question Group.
- The DCM cannot be defined as Optional.
- The DCM must contain at least one DCM Question that has an Internal or Thesaurus DVG attached.
- If a DCM is defined as having a Qualifying Value, this value must be defined in the DCI Module for this DCM.
- **Use DCI Date?** must be selected in the DCI Modules window for the DCM.

If a DCM has a qualifying value, it is listed multiple times; once with each qualifying value specified in a DCI Module for the DCI.

Question - From the list of values, select the Question whose value you want to serve as the trigger for the rule. If you selected [Any Data] in the DCM/Qual Value field, you cannot select a Question.

The list of values includes all Questions defined for the selected DCM and Qualifying Value (if applicable) in the current study that meet the following conditions:

- The Question must be of type CHAR (not NUMBER or DATE). See "Using Derivation Procedures" in [Rule Triggers](#) if you need to use number Questions.
- The Question must have an Internal or Thesaurus discrete value group (DVG) assigned.
- The Question must be defined as Collected in Subset.
- The Question must be included in a non-repeating Question Group.
- The Question may be enterable or derived but cannot be complex; see "Using Derivation Procedures" in [Rule Triggers](#) for further information.

Value(s) - From the list of values, select the value or values you want to trigger the rule. The list of values includes all the values defined in the Question's Discrete Value Group (DVG).

CPE(s) at which Trigger DCI is Defined - This read-only field lists all the CPEs currently defined in this DCI Book as including the selected DCI. The rule you define applies to all occurrences of the DCI in the DCI Book.

Defining an Interval Rule Target

The target portion of the rule specifies the Interval or Intervals to be enabled or bypassed to if the rule trigger condition is met.

←-----Target----->			
Action	Interval (s)	Rule Description	
Bypass to	...	END OF STUDY	...
Enable	...	END OF STUDY	...
Enable	...	INTERVAL A,INTERVAL AA	...
Enable	...	INTERVAL B,INTERVAL BB	...

To define an Interval rule target, enter values in the following fields:

Action - From the list of values, select one of the following:

- **Enable:** If you select **Enable**, when the data condition defined in the trigger section is met for a patient, the Intervals listed in the **Interval(s)** field, all the CPEs within the Intervals, and all unconditional DCIs collected at those CPEs, become expected for that patient.
- **Bypass To:** If you select **Bypass To**, when the data condition defined in the trigger section is met for a patient, the Interval listed in the **Interval(s)** field becomes expected for the patient, and all intervening Intervals—and any remaining CPEs after, but not including, the CPE where the trigger DCI was collected, in the trigger Interval—become definitely not expected for the patient. Any remaining DCIs in the trigger CPE are still expected.

Interval(s) - From the list of values, select the name of the Interval or Intervals you want to serve as the target of the rule, or select [Next Interval].

- **Interval(s) by name:** If you have selected an Action of **Enable**, you can select any number of target Intervals. If you have selected an Action of **Bypass To**, you can select only one target Interval.
- **[Next Interval]:** [Next Interval] is the next Interval in which all CPEs have a higher Visit No than the CPE containing the Trigger DCI. This enables you to enable multiple Intervals with a single rule, which may be especially helpful with cycling studies. See **Next Interval** in [Rule Targets](#) for further information. This option is not available for rules with the **Bypass To** action.

The list of values includes [Next Interval] and all Intervals defined for the current study that meet the following criteria:

- The Interval must have CPEs assigned to them directly; for example, a Phase is not included if there are no CPEs assigned directly to the Phase, even if the Phase contains Periods or Sub-Periods with CPEs assigned directly.
- The Interval's CPEs must all occur after the last CPE containing the trigger DCI, if the trigger DCI is already included in the DCI Book.

- The first Interval in a study cannot be the target of a rule and so is not included in the list of values.
- The predefined DEFAULT STUDY PHASE is not available as a target Interval.

The list of values displays Intervals in CPE order. If there are no Intervals that meet these criteria, no Intervals are displayed in the list of values.

Rule Description - Optional, free text field, up to 200 characters.

Defining DCI Rules

By defining a DCI rule, you make one or more DCIs conditional, so that they are expected for a patient only if the patient's own data meets the criteria specified in the rule trigger. You are not required to define DCI rules in a flexible study. If you have no DCI rules, all DCIs are expected for all patients assigned to the Book, although entire Intervals may be conditional if you define Interval rules.

To define rules for a DCI Book, the current study must be defined as flexible in Clinical Study States (under Conduct, then Security).

You can define rules either before or after defining related DCI Book Pages.

Note:

To see all the rule fields at the same time, expand the window by dragging the edge to the right. This does not work the first time you open the window during your session, but if you close the window and reopen it you can then expand it.

For more information, see:

- [Defining a DCI Rule Trigger](#)
- [Defining a DCI Rule Target](#)

Defining a DCI Rule Trigger

The trigger portion of a DCI rule specifies the data condition required to trigger the rule, making the target DCI or DCIs expected for the patient whose data satisfies the condition.

Note:

The DCI that contains the trigger must itself be expected in order to make the target DCI expected. If a DCI that is used as a rule trigger is no longer expected, it can no longer trigger expectedness of another DCI even if a triggering data value was entered before the trigger DCI became unexpected.

DCI	DCM / (Qual Value)	Question	Value(s)	CPE(s) Trigger Defined
DEMO	DEMOGRAPHIC	DM_SEX	F	VISIT 1

To define an DCI rule trigger, enter values in the following fields:

DCI - From the list of values, select the DCI that contains the DCM or Question whose value you want to serve as the trigger for the rule. You can query on this field.

The list of values includes the long and short names of all Provisional and Active DCIs with DCMs defined for the current study, whether they are currently defined in the DCI Book or not. If they are not yet included in the DCI Book, you must add them in the DCI Book Pages window before you can activate the Book.

DCM/Qual Value - From the list of values, select the DCM that contains the Question whose value you want to serve as the trigger for the rule. You can query on this field.

The list of values includes all DCMs defined for the selected DCI in the current study that meet the following conditions:

- The DCM must include at least one non-repeating Question Group.
- The DCM cannot be defined as Optional.
- The DCM must contain at least one DCM Question that has an Internal or Thesaurus DVG attached.
- If a DCM is defined as having a Qualifying Value, this value must be defined in the DCI Module for this DCM.
- **Use DCI Date?** must be selected in the DCI Modules window for the DCM.

If a DCM has a qualifying value, it is listed multiple times; once with each qualifying value specified in a DCI Module for the DCI.

Question - From the list of values, select the Question whose value you want to serve as the trigger for the rule.

The list of values includes all Questions defined for the selected DCM and Qualifying Value (if applicable) in the current study that meet the following conditions:

- The Question must be of type CHAR (not NUMBER or DATE). See "Using Derivation Procedures" in [Rule Triggers](#) if you need to use number Questions.
- The Question must have an Internal or Thesaurus DVG assigned
- The Question must be defined as Collected in Subset
- The Question must be included in a non-repeating Question Group.
- The Question may be enterable or derived but cannot be complex; see "Using Derivation Procedures" in [Rule Triggers](#) for further information.

Value(s) - From the list of values, select the value or values you want to trigger the rule. The list of values includes all the values defined in the Question's Discrete Value Group (DVG). If there are many values, you may want to use the Find, Select All, or Clear All functions to speed the selection process.

CPE(s) at which Trigger DCI is Defined - For saved rules, this read-only field lists all the CPEs currently defined in this DCI Book as including the selected DCI. The rule applies to all occurrences of the DCI.



Note:

This field is blank if the trigger DCI is not yet used in the DCI Book.

Defining a DCI Rule Target

The target portion of a DCI rule specifies the DCI or DCIs enabled for the patient whose data satisfies the trigger condition. If a patient's data does not satisfy the trigger condition, the target DCI(s) are not expected for the patient.

Enable DCI		Target		Rule Description
Within CPE	Across CPEs	DCI	CPE(s) at which Target DCI(s) are Defined	
<input type="radio"/>	<input checked="" type="radio"/>	PREG	...	TERMINATION(1)
<input type="radio"/>	<input type="radio"/>		...	
<input type="radio"/>	<input type="radio"/>		...	
<input type="radio"/>	<input type="radio"/>		...	
<input type="radio"/>	<input type="radio"/>		...	
<input type="radio"/>	<input type="radio"/>		...	
<input type="radio"/>	<input type="radio"/>		...	
<input type="radio"/>	<input type="radio"/>		...	

To define a DCI rule target, enter values in the following fields:

Enable DCI Within CPE - Select this option if you want to enable the target DCI only within the same CPE as the trigger DCI, even if it occurs in other CPEs. Note that:

- The trigger DCI can be included in multiple CPEs.
- The trigger DCI must have a display number less than the target DCI(s) within every CPE in which they both exist.
- The target DCI can only exist in the same CPE as its trigger DCI. However, a trigger DCI can appear in a CPE that does not include its target.
- Each target DCI does not need to be included in every CPE that includes the trigger DCI, but each target DCI must be included in at least one CPE that includes the trigger DCI.

 **Note:**

This is enforced only when you validate the Book, not when you are defining the rule.

Enable DCI Across CPEs - Select this option if you want to enable the target DCI in every CPE where it is defined (including in the same CPE as the trigger DCI, if it is included there).

 **Note:**

The trigger DCI must be included in only one CPE.

DCI - From the list of values, select the DCI or DCIs that you want to enable if the trigger condition is met. The list of values displays the short and long name and the status of each DCI defined for the current study that satisfies the following criteria:

- DCIs must not already be the target of another rule.
- DCIs cannot be the same as the trigger DCI.

CPE(s) at which Target DCI is Defined - This read-only field lists the CPEs that include the target DCI. If you defined multiple target DCIs, the number of target DCIs in each CPE is displayed in parentheses () following the CPE name. If you selected Across CPEs, the rule enables all occurrences of the DCI.

 **Note:**

This field is blank if none of the target DCIs is included yet in the DCI Book.

Rule Description - This field is optional. You can enter up to 200 characters.

Validating an Enhanced DCI Book

The Maintain Enhanced DCI Book window gives you the flexibility to define rules and Book pages in any order by allowing you to temporarily put your Book into an invalid state.

However, the book must be valid—have a validation status of either Success or Warning—before it can be activated and used. The validation job checks for conflicts between Interval and DCI rules and DCI Book pages, then outputs a list of problems, if any, and sets the validation status for the DCI Book.

When you set a Book's status to Active and save, the system runs the validation job behind the scenes. If the Book has errors, the activation fails. If the Book has warnings but no errors, the activation succeeds.

You can also run the job at any time during the Book definition process to help you find and fix problems.

To run the job, select **Validate DCI Book** from the **Special** menu in the **Maintain Enhanced DCI Books** window. The option is not available if the Book is Active or if the Book's validation status is Success, Warning, or Error. (If the validation status is Warning or Error, no changes have been made to the Book since the validation job was run and you can view the results of the last validation; see [Viewing Validation Warnings and Errors](#).)

See also [Defining Book Properties](#), Validation status.

In this section:

- [Validation Checks](#)
- [Viewing Validation Warnings and Errors](#)

Validation Checks

The validation utility tests for conflicts and generates an error or a warning as follows.

Error Conditions for Interval Rules

The validation tests for the following conditions and raises an error if the requirements are not met, causing a validation status of Error and preventing the Book from being activated:

- Each rule's trigger DCI must be defined at one or more book pages.
- An Interval that is the target of an Interval rule must have at least one Book page defined for at least one of its CPEs.
- The CPEs in the Interval at which the trigger DCI is collected must all have a lower Visit Number than the first CPE in the target Interval. This does not apply to [Next Interval] rules where the target Interval is implicitly the first Interval whose first CPE follows the CPE containing the trigger DCI.
- Duplicate Interval rules are not allowed. The system compares the trigger DCI, DCM, and Question and the target action and Interval. If these are all the same in two or more rules, the rules are considered duplicates. The system does not consider the Question value or CPEs.

Note:

This rule is enforced during rule definition, not validation.

- A **Bypass To** Interval rule can have only one Interval as a target, and that target cannot be Next Interval.

 **Note:**

This rule is enforced during rule definition, not validation.

Error Conditions for DCI Rules

The validation tests for the following conditions and raises an error if the requirements are not met, causing a validation status of Error and preventing the Book from being activated:

- Each rule's trigger DCI must be defined at one or more book pages.
- Each DCI rule must have at least one Book page defined for each of its target DCIs.
- For an "Enable Within CPE" DCI rule, the trigger DCI must have a display number lower than all the target DCIs.
- For an "Enable Within CPE" DCI rule, each target DCI must be included in at least one CPE that includes the trigger DCI (but each target DCI does not need to be included in every CPE that includes the trigger DCI).
- For an "Enable Within CPE" DCI rule, the target DCI(s) cannot exist in a CPE without the trigger DCI.
- For an "Enable Across CPEs" DCI rule, the trigger DCI must be included in only one CPE.
- For an "Enable Across CPEs" DCI rule, each target DCI must be included only in CPEs that have a higher Display number in the DCI Book than the CPE that includes the trigger DCI.
- A DCI can be the target of only one rule.

 **Note:**

This rule is enforced during rule definition, not validation.

Warning Conditions

The following conditions constitute warnings. If the Book has one or more warnings but no errors, the validation job generates a validation status of Warning and the Book can be activated. However, you may want to fix these problems.

- Start Page numbers are consistent with ascending CPE number order and there are no gaps among Start Page numbers with the same seed. Any Start Page numbers that are out of order or have gaps are reported.

If you want to fix out-of-order pages or gaps, you can run the Renumber Book Pages job on a set of pages or on the whole Book; go to the **Special** menu in the **DCI Book Navigator** window.

 **Note:**

If the current study is not defined as flexible, the DCI Book has no rules and therefore the Start Page number checks are the only ones that can generate a warning. The validation status of Books in nonflexible studies is always Not Applicable. To see if there are any Start page number-related warnings, select **View Validation Results** from the **Special** menu.

- For an "Enable Within CPE" DCI rule, a trigger DCI exists at a CPE without any of the target DCIs.
- For an "Enable Across CPEs" DCI rule, a target DCI is not included in any CPEs other than the one that includes the trigger DCI.

Viewing Validation Warnings and Errors

The validation utility tests all rules and Book pages even after finding an error and produces a complete list of all warnings and errors, including information on the specific rules and pages involved.

To see this list, which appears as a text box on screen, select **View Validation Results** from the **Special** menu in the **Maintain Enhanced DCI Book** window. You can copy the text and paste it into a file to print it. The results persist until the validation job is run again for the Book.

Defining DCI Book Constraints

Access the DCI Book Constraints window by clicking the **DCI Book Constraints** button on the DCI Books window. (From the **Definition** menu, select **DCIs**, then choose **Enhanced DCI Books** or **DCI Books**. The functionality is the same for both types of Book.)

Maintain Enhanced DCI Book (Study: FLEXSTUDY2)

Maintain DCI Book Constraints for Book STUDYBOOK1

Unplanned Use Allowed if not listed below?

DCI Name	Planned in Book	Unplanned Use Allowed	Preferred Version
AE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
COMP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
CONT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
DEMO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
EOT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
PREG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
RAND	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
DASM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
MHIST	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT
	<input type="checkbox"/>	<input type="checkbox"/>	

The DCI Book Constraints window includes all DCIs assigned to the DCI Book, each marked "Planned in Book."

Use the Constraints window to do the following:

- You can make one or more DCIs already assigned to the Book available for unplanned use as well by selecting **Unplanned Use Allowed** for the DCI.
- You can specify that all Study DCIs that are not listed in the Constraints table can be used to create RDCIs for unplanned events by selecting **Unplanned Use Allowed if Not Listed Below**.
- If you have not allowed unplanned use of all unlisted Study DCIs, you can allow unplanned use for selected Study DCIs by adding them and selecting **Unplanned Use Allowed**. Click in the field to open the list of values.

If the DCI Book is Active, this list of values includes only Active DCIs that do not have any Retired DCMS. If the DCI Book is Provisional, then this list of values includes both Active and Provisional DCIs that do not have any Retired DCMS.

 **Note:**

The settings for Unplanned Use apply only to RDC data entry, and more information is available in the RDC documentation. In Oracle Clinical data entry, this concept of unplanned pages does not apply as it is not possible to enter a DCI for any visit other than those for which it is defined as a DCI page in a DCI Book. However, in Oracle Clinical you can use a subevent_number greater than zero to enter unplanned events.

- For each DCI, in the **Preferred Version** field specify which DCI Form Version to use for data entry in RDC Onsite or select CURRENT so that whatever version is current at runtime is used. This value is also used in migration.

See [Maintaining Form Layout Versions](#) and [Migrating Data to New Oracle Clinical DCI Form Versions](#) for more details.

Calculating DCI and Interval Expectedness for Patients

In order to accurately reflect which DCIs and Intervals are expected for each patient in real time in RDC Onsite data entry, Oracle Clinical automatically recalculates expectedness in almost every situation where it might change for a patient.

There are just a few situations in which you should run the job manually.

For more information, see:

- [Automatic Expectedness Calculation for Patients](#)
- [When and How to Run Expectedness Calculation Manually](#)

Automatic Expectedness Calculation for Patients

In this section:

- [Automatic Incremental Expectedness Calculation](#)
- [Automatic Full Expectedness Calculation](#)

Automatic Incremental Expectedness Calculation

The incremental mode of the expectedness calculation job is sufficient for updating expectedness due to changes in patient data. It begins at the patient's earliest CPE with data changes that might impact expectedness—that is, data entered for a Question or DCM that is defined as a rule trigger.

The system automatically runs incremental expectedness processing during Manual Data Entry and Batch Data Load and Batch Data Delete.

Manual Data Entry - The system automatically runs incremental expectedness calculation during data entry in Oracle Clinical and RDC, updating the expectedness of DCIs and Intervals for the patient if required. This occurs when the user saves data.

In RDC it occurs at every data entry transaction and, if the patient requires recalculation, as soon as a user opens a CRF for a patient.

Batch Data Load and Batch Data Delete - For Batch Data Load and Batch Data Delete, the system performs the expectedness calculation immediately after completing the main transactions to minimize locking time. Each patient with modified data is processed by a separate transaction.

Other Types of Data Changes - The system also recalculates expectedness as follows:

- **Mass Changes:** Every Received DCI is a single transaction which includes the immediate expectedness calculation, if needed.

- **Patient Transfers:** All changes for a single Patient transfer are performed as part of the transfer transaction, including the processing of multiple patients in an ongoing transfer.
- **Derivation Procedures:** Oracle Clinical processes the changes for each Received DCI as a transaction. The expectedness calculation is performed as part of these transactions if the Derivation Procedure derives a trigger Question. This applies whether the Derivation Procedures are run in Batch Validation, Execute Single Procedures, or as Online procedures from RDC.

Automatic Full Expectedness Calculation

Full expectedness calculation performs a complete recalculation of expectedness, beginning at the patient's first CPE and discarding all previous results. The system uses it when a patient is assigned or reassigned to a DCI_Book.

Full mode is required and automatically run as follows:

- Immediately after a user assigns or reassigns patients to a DCI Book using the user interface
- At the end of batch validation in production mode, if batch validation detects patient book assignment changes made using a Validation or Derivation procedure. In addition, any Full processing that may have been left incomplete due to a system failure is detected and processed.
- At the end of an online Procedure execution that detects patient book assignment changes
- During the first data entry transaction (including opening a CRF in the RDC Surround) for any patient marked as requiring expectedness recalculation

Expectedness calculation is *not* run automatically in the following situations:

- For test mode patients after their assigned DCI Book is successfully revalidated, because this may happen many times in test mode during DCI Book definition
- Following a batch validation in test mode, even if patient book assignments have been changed
- In production mode, when an active DCI Book has been made Provisional and then reactivated
- When a patient was assigned to a different Book while "frozen" and is subsequently unfrozen

You may want to run expectedness calculation manually in these situations.

When and How to Run Expectedness Calculation Manually

In this section:

- [When to Run Expectedness Calculation Manually](#)
- [How to Run Expectedness Calculation Manually](#)

When to Run Expectedness Calculation Manually

Because the system runs expectedness calculation automatically each time patient data is updated and after patient book assignment changes, there are very few situations in which you need to explicitly calculate expectedness. They are:

- After activating an enhanced DCI Book that has patients assigned
- After "unfreezing" patients whose assigned book was activated while they were frozen
- After making an existing study flexible in the Clinical Study States window: you must then set any active books to Provisional and reactivate them, and then run Calc Expectedness in Test for all patients and all books. (If a nonflexible study has production data you cannot make the study flexible.)

In addition, if there is a system failure during a batch data load or batch data delete, expectedness calculation may not have completed. You may want to run the job manually. If you do not run the job manually, and if the book is valid, the system will run the job automatically for each patient during the next data entry transaction for the patient or the next batch validation.



Note:

Run Calc Expectedness in Test or Prod in the same situations for an Enhanced DCI Book with no rules defined as you would for a Book with rules to ensure that the correct CRFs are displayed for patients in RDC Onsite.

How to Run Expectedness Calculation Manually

The test and production modes of the job are available as follows:

- The test mode option is available if the DCI Book has a status of Active or a status of Provisional and a validation status of Warning or Success.
- The production mode option is available only if the DCI Book has a status of Active.

To run the job:

1. Select **Calc Expectedness in Test** or **Calc Expectedness in Prod** (to run on test or production patient data) from the **Special** menu in the **Maintain Enhanced DCI Books** window.
2. In the pop-up window, select one of the following:
 - **All Patients**: This option forces full expectedness recalculation for all patients. This option should never be necessary but is provided just in case.
 - **Only Patients Requiring Recalculation** runs full expectedness recalculation only on patients who have been identified as requiring recalculation: those who have been "unfrozen" since the DCI Books was last activated, or whose data expectedness has not been successfully calculated since the Book was last activated.
3. In the pop-up window, select one of the following:
 - **Current Book Only**
 - **All Books in Current Study**
4. Click **OK**. The job runs.

When the job finishes, a message appears saying whether or not the calculation job has run successfully.

Data Entry and Expectedness

In this section:

- [Data Entry in Remote Data Capture Onsite](#)
- [Data Entry in Oracle Clinical](#)

Data Entry in Remote Data Capture Onsite

Remote Data Capture (RDC) Onsite displays blank CRFs in the Casebooks page only if the corresponding DCI is *expected* for the patient. In addition, when a data entry operator clicks **Next** in the data entry window, the next CRF expected for that patient in the current CPE appears, regardless of which CRFs may be expected for other patients in the same study.

DCIs that are the target of a rule are expected for a patient only when:

- The Interval containing their CPE is expected for the patient.
- The trigger condition of the DCI rule is met for the patient.
- The status of the patient's RDCI for the trigger DCI is Batch Loaded, Pass 1 Complete, or above.
- The patient is not frozen.
- The RDCI is at subevent 0.
- The DCI that contains the trigger is expected.

Note:

The Blank Casebook report, including the version with header information for a particular patient, does not consider expectedness for the patient. It includes all DCIs in all Intervals, whether or not they are expected for the patient at the time the report is generated.

If sites are printing the blank casebook and using it for data entry, be sure to provide instructions for entering data for the correct DCI Book pages. You may want to provide a visual cue for trigger Questions in the graphic layout.

Data Entry in Oracle Clinical

If a DCI Book with rules is used for manual data entry in Oracle Clinical:

- **All DCI Book pages are available** for data entry whether they are expected for a patient or not. You must provide instructions to data entry operators to ensure that they enter data for the correct DCI Book pages.
- **Interval and DCI Rules are triggered** if data entered through Oracle Clinical meets the defined trigger conditions. If you use a mix of data entry methods in a single study, DCIs and Intervals that become expected for a patient due to data entered in Oracle Clinical are displayed as expected for the patient in RDC.

Assigning Patients to a DCI Book

To assign patients to a DCI Book, in the Oracle Clinical Navigation tree select **Design**, then **Patient Positions**, then **Patients**, and then click **Patient Positions**, and then click **Assign DCI Book**; see [Assigning Patient Positions to DCI Books](#) for more information.

Note:

For flexible studies, it is **required** that all patients in the study be assigned to a DCI Book. Unassigned patients are not retrieved in RDC Onsite.

If a patient's book assignment is removed while a user is working with that patient's data in RDC Onsite, **errors can result** until the user issues another query, refreshing the set of patients retrieved.

Defining a Traditional DCI Book

You use a DCI Book to sequence the order of DCIs and their default keys during the data entry login process. This sequencing simplifies the process of entering CRFs, which are themselves sequenced in a predefined order. Most companies enter CRF information from a clinical record book organized in CPE sequence. You can use Oracle Clinical's DCI Book feature to create the electronic equivalent, with the DCIs, which correspond to the CRF pages, ordered as the information in the clinical record book appears. In this way, a DCI Book is like an index to the clinical record book. (When you add a DCI to the DCI Book Pages, the system automatically adds it to the DCI Book Constraints and marks it as "Planned in Book." You can specify DCI version usage for a book, and set a policy for which DCIs can or cannot be used for unplanned events. See [Defining a Traditional DCI Book](#). See also [Using CRF Page Tracking](#), for instructions on enabling DCI Book page tracking.)

A study can have many DCI Books for different purposes, or just one for the whole study. You give each DCI Book a unique name. A login entry operator chooses the DCI Book appropriate for a CPE's CRFs.

You can define the pages of the DCI Book by selecting the **DCI Book Pages** button to invoke the DCI Book Pages window.

Note:

Oracle recommends not creating non-sequential CPE numbering for pages in DCI books. Among other problems, non-sequential CPEs do not display in RDC.

For more information, see:

- [Creating a DCI Book](#)
- [Defining DCI Book Constraints](#)

Creating a DCI Book

You define a DCI Book by navigating to **Definition**, then **DCIs**, and **DCI Book**. The Maintain DCI Book form opens, displaying the DCI Books window. Enter the name and a description of the DCI Book. If you define more than one DCI Book, you can select one as the default to offer when you choose a DCI Book at login.

You give each DCI Book page a page number, the name of a DCI, and, optionally, the name of a CPE. Omitting the CPE might be useful when you have a set of optional DCIs not pre-allocated to particular CPEs.

The page numbers are the page numbers of the first page of the CRF that corresponds to the DCI. The page numbers must be unique; if you have multi-page CRFs corresponding to a single DCI, the page numbers will not be continuous. You do not have to enter the pages sequentially, but the form sequences them in ascending page order when you re-query. See the following section for enabling CRF page tracking in a study created in Oracle Clinical 3.1 or later.

When you add a DCI to DCI Book pages, the system also adds it to the DCI Book Constraints table.

Defining DCI Book Constraints

This functionality is the same in traditional and enhanced DCI Books; see [Defining DCI Book Constraints](#) for information.

Using CRF Page Tracking

Oracle Clinical provides the capability to track each page of a paper CRF online. This capability has three primary benefits. First, it allows you to identify when only a page or two from a document is missing (rather than the entire CRF). The missing page can then either be located or explicitly identified on line as missing. Second, it allows discrepancies to be identified as being on a particular page. This would be helpful if you were discussing a discrepancy with someone by phone. Third, if you are scanning CRFs into the system, the concept of pages (versus documents) becomes very important. Scanners enter documents a page at a time, so it is the concept of a page that is central.

Using this feature requires enabling options at the following Oracle Clinical levels, then the tasks required at each level are described in detail in separate sections:

- Study Definition
- DCI Definition
- DCI Book Definition
- Patient Enrollment or Patient Maintenance
- Data Entry

For more information, see:

- [Enabling CRF Page Tracking for a Study](#)
- [Defining DCIs to Track CRF Pages](#)
- [Defining DCMs to Track CRF Pages](#)

- [Defining DCI Books to Track CRF Pages](#)

Enabling CRF Page Tracking for a Study

Page tracking must be enabled as part of the definition of a study. You cannot enable page tracking in a study created prior to Oracle Clinical Release 3.1 if data has been entered for that study or if the study is defined as Flexible.

To include page tracking as part of the study definition, you must select the **Enable CRF Page Tracking** box on the Clinical Study States form. To access the Clinical Study States form, select **Conduct**, choose **Security**, then select **Clinical Study States**.

Defining DCIs to Track CRF Pages

Page tracking actually happens at the DCI level, since the DCI is the equivalent of the CRF. Once CRF page tracking is enabled for the study, complete the following fields for each DCI that you want to track pages for. The fields provide Oracle Clinical with the specifications needed to recognize the information expected on a particular CRF page.

CRF Page Tracking Enabled? specifies whether page tracking is enabled for this DCI. If page tracking was not enabled at the study level, you cannot enable it here.

Page tracking for this DCI is not enabled if this parameter is not selected. However, you may choose to complete the remaining fields in preparation for using page tracking at a future time.

Default Page Numbering Scheme establishes the default page numbering scheme that Oracle Clinical will use for the pages of an expected DCI—a DCI that is specified in the default DCI Book for the specific CPE. This numbering scheme describes how each page of the CRF will be designated in the DCI. This field has a list of values with the following entries:

- **SEQUENTIAL** – Designate each page by a sequential integer, for example, 1, 2, 3.
- **ALPHANUMERIC** – Designate each page by an integer plus an alphabetic suffix; for example, 1a, 1b, 1c.
- **NUMERIC** – Designate each page by an integer plus a numeric suffix; for example, 1.1, 1.2, 1.3.
- **USER-DEFINED** – A sequence defined by a user-defined database function.

For the user to be able to select the user-defined numbering scheme, its value needs to have been entered in the installation reference codelist CRF PAGE NUMBERING SCHEME.

Unplanned Page Numbering Scheme establishes the default page numbering scheme that Oracle Clinical will use for the pages of an unplanned DCI—a DCI that is unexpected. A DCI is regarded as expected if it is in a DCI Book, has a clinical planned event assigned and an actual event number of zero. Anything else is regarded as unexpected. This numbering scheme describes how each page of the CRF will be designated in the DCI. This field has a list of values with the following entries: SEQUENTIAL, ALPHANUMERIC, NUMERIC.

It is possible to assign an event to a DCM within an unplanned DCI. In this case, the system uses the unplanned scheme for recording pages for that RDCM.

Expected No. of Pages establishes how many physical pages will be in the DCI. This number cannot be changed once production data has been collected with the DCI.

Oracle Clinical does not require you to invoke page tracking for every DCI used in a study with page tracking enabled. You can disable page tracking for an individual DCI. To access the Data Collection Instruments form, from the **Definition** menu, select **DCIs**, then choose **DCIs**.

If your DCI consists of one DCM and is complete on a single page or multiple DCMs all complete on one page, then no further definition is required for CRF page tracking. However, if any DCMs span multiple pages, then you must complete a Page Definition form that describes what Question to expect at the end of each page of each DCM in the DCI. To learn about these definitions, read the next sections.

Defining DCMs to Track CRF Pages

If a DCI spans multiple pages, then Oracle Clinical must be able to determine which page every piece of information falls on. The first step is to specify the pages on which each DCM falls. Use the Data Collection Modules window on the DCI form. The CRF page tracking fields on this form are:

Start Page - The DCI page on which the DCM begins. All pages are given relative to the DCI, and the value entered in this field is validated to check whether this value conforms to the numbering scheme defined for that DCI in the DCI form.

End Page - The DCI page on which the DCM ends. All pages are given relative to the DCI.

To provide the additional information Oracle Clinical requires:

1. From the **Definition** menu, select **DCIs**, then choose **DCIs**.
2. Select the DCI.
3. Click the **DCI Modules** button.
4. Select the DCM.
5. Scroll the form horizontally until you reach the fields **Start Page** and **End Page**.
6. Complete these fields using the descriptions provided above.
7. Save
8. Click the **Back** button to return to the DCI form.

To expand additional DCIs, repeat Steps 2-7.

For more information, see:

- [Defining DCM Pages](#)

Defining DCM Pages

When a DCM spans multiple DCI pages, Oracle Clinical must know where on the page the information splits. To do this, specify the final Question on the page on the Data Collection Module Pages window.

This procedure is only required for DCMs that span multiple pages. Note that you do not provide information about the last page of the DCM. The system assumes that Questions coming after the last Question on the previous page are on the last page.

The CRF page tracking fields on this form are:

Field	Usage
Page Relative to DCI	The DCI page for which you are providing information. All pages are given relative to the DCI and are defaulted by the system.
Question Name	The name of the Question that appears last on the page.
Question Occurrence Sn	The Question Group occurrence sequence number that appears last on the page.
Last Repeat on Page	If this Question belongs to a repeating Question Group, that is the last repeat instance that appears on the page. Leave this space blank if the Question is not repeating.
Repeat?	Indicates whether this Question belongs to a repeating Question Group; defaulted by the system.

To provide the additional information Oracle Clinical requires:

1. In the Data Collection Modules window of the DCI form, place the cursor in the row of the DCM you need to expand.
2. Click the **DCM Module Pages** button.
3. Place the cursor in the first row of the page.
4. Scroll the form horizontally until you reach the final fields on the form.
5. Complete these fields using the field descriptions above.
6. Repeat Steps 3-5 for each page.
7. Click the **Back** button to return to the DCM form.

Defining DCI Books to Track CRF Pages

The DCI Book is the vehicle for applying the page tracking features of Oracle Clinical to data entry. Once you establish the page tracking criteria for each DCI, group them in study-specific DCI Books. Each study has a default DCI Book. Each patient is also linked to a DCI Book. If you do not explicitly assign a patient to a DCI Book, Oracle Clinical assigns that patient to the default DCI Book for the study. You can change to a different DCI Book in data entry. [Defining a Traditional DCI Book](#) further describes defining DCI Books without page tracking.

For more information, see:

- [Creating the DCI Book](#)
- [Defining DCI Book Pages](#)
- [Describing Each DCI Page](#)

Creating the DCI Book

DCI Books are flexible and allow you to truly match your physical CRF book to your online DCIs. As with the DCI definition, you must complete a series of drill-down forms

to describe the configuration of the DCI Book to Oracle Clinical. Begin with the Maintain DCI Book form.

To create a DCI Book definition:

1. From the **Definition** menu, select **DCIs**, then choose **DCI Books**.
2. Complete the fields:
 - **Status** – Database status for the DCI Book; controls the usage and attributes. Choices are P for Provisional, A for Active, or R for Retired. P is the default value for a newly created DCI Book. Since DCI Books are now definitional elements—accurate page tracking is dependent on them—you cannot delete them once you use them to enter production data.
 - **Status Comment** – Optional field allowing you to comment on changes made to the **Status** field.
 - **Retirement Reason** – Optional field allowing you to comment if you retire a DCI Book.
3. Save. To exit the form without specifying CRF page tracking information, select **Exit**.

Defining DCI Book Pages

If you plan to utilize CRF page tracking, you must now provide additional information about each DCI Book page. Use the following field descriptions to understand how the entry in each field relates to CRF page tracking:

Field	Relationship to CRF page tracking
Display #	Number that represents the DCI's place in the sequence of the DCI Book. The list does not have to be sequential; you can rearrange it so that DCI #4 precedes #2.
Start Page #	Page number assigned to the first page of the physical CRF. Since this number matches the actual number on the physical CRF, the format of this number can be different for each DCI in the book. It is also editable. The only restriction is that there cannot be two identical entries within the column.
DCI Name	Name of the DCI.
Clinical Planned Event	Clinical planned event with which this DCI is to be used.

To define DCI Book pages:

1. In the DCI Book form, place the cursor in the row of the DCI Book you need to expand.
2. Click the **DCI Book Pages** button.
3. Add a row to this form for each DCI that should be part of this book. Use the field descriptions above and field help to complete each field.

It is important to understand the relationship between the **Display #** and the **Start Page #** fields. The display number describes the relative position of the DCI within the book. It is always an integer, but integers in this column do not have to be sequential. The Start Page Number is specific to the DCI. It describes the actual first page of the physical CRF, and each entry in this column can be based on a unique page numbering format.

When you enter a new DCI, if you are using a sequential, numeric, alphanumeric, or user-defined format, Oracle Clinical also calculates how many pages were in the previous DCI and generates an entry in the **Start #** field. You can change this entry, but you can only use the

numbering scheme defined for that DCI; if the Start Page Number is in a different format from the DCI's numbering scheme, the system gives a warning message.

**Note:**

Do not use identical characters twice within the column. For example, do not define two DCIs that start on page 2a. This can result in missing pages in the patient data and blank casebook reports.

For each DCI you add to this form, complete the steps in the next section.

Describing Each DCI Page

For each DCI described using the previous section, Oracle Clinical provides a DCI Book Physical Pages form. The previous form describes the overall parameters of each DCI within the book—how many pages and the page number of the first page. This form follows up with specific details about each physical page in the DCI. The information in this form is system generated, but you can overwrite fields, as required.

The fields on this form are:

Field	Definition
Seq within DCI	Number that identifies the position of the page within the DCI.
Book Page #	Page number assigned to the individual page that aligns with the number displayed in the first column. This field is editable so you can explicitly match it to its physical counterpart.
Description	Optional field describing the individual page.

Annotating CRFs

As part of your internal processes and/or a component of a regulatory submission, you may find it necessary to annotate the CRFs you use in a protocol. Oracle Clinical provides the capability to create an annotated CRF from a DCI graphical layout via the Graphic Layout Editor (GLE). The annotated CRF that you create in Oracle Clinical uses the names of the fields you setup in the GLE, as well as any other attributes you specify. The annotations are either located adjacent to each field or else a number is used to associate each field with its annotation in a footnote. Because this process is automatically generated and extensively customizable, the process of annotating your CRFs using Oracle Clinical is can be integrated into your normal CRF design workflow.

This section discusses where in your CRF design workflow you may find it advantageous to generate annotated CRFs, describes the procedures you use to create them, and defines the tools you use in these procedures.

For more information, see:

- [Discussion](#)
- [Procedures](#)
- [Using the CRF Annotation Tool Window](#)

Discussion

Although you can use an FLT or DCM block layout to create an annotated form at any time during the CRF design process in order to aid your internal mapping and/or auditing procedures, the primary purpose of creating an annotated CRF in OC/RDC is for inclusion in regulatory submissions. For this usage, you start with a DCI form layout. Oracle recommends that you create an annotated CRF for each version of a DCI form layout you use and, if you modify a form during a study, that you create an updated annotated CRF for that form.

Note that the process of creating an annotated form from a DCM Block layout or an FLT layout is identical to that described for the annotated CRF.

The process of creating an annotated CRF is straightforward. When the DCI graphical layout is finalized, you use the GLE to create an "annotation-ready" PDF of the layout and open it in the PDF DEW. When you open this file in the PDF DEW, the system recognizes that it is annotation-ready and only the CRF Annotation toolbar button is active.

You then open the CRF Annotation tool and select a series of settings that control both the type of annotations to include in the PDF and how you want it displayed. Then you save the annotated form to your file system.

When the GLE sends a copy of the DCI layout to the PDF DEW there is no further connection to the GLE or to Oracle Clinical. The file that is displayed in the PDF DEW is standalone – you can copy, save, or delete it and there is no impact on the DCI layout in OC. If you subsequently update the DCI form in the GLE, those changes are not transferred to a previously generated annotated CRF that is open in the PDF DEW. In that case, you would simply create another annotated CRF.

Since the annotated CRF is separate from the DCI layout in the GLE, you can use the tools included with Adobe Acrobat to modify it as you see fit. It is a discrete PDF file, that you can save, add further comments to, make a copy of, or delete, all within the functionality of Acrobat. In addition, you can re-open the CRF Annotation Tool and change its settings as often as necessary, either to fine-tune the information in the annotations or to create multiple copies form with different annotation configurations.

For more information, see:

- [Definition](#)
- [Requirements](#)
- [Annotation Placement Styles](#)

Definition

An annotated CRF is generally defined as a blank CRF with markings, or annotations, that coordinate each datapoint in the form with its corresponding dataset name. Essentially, an annotated CRF communicates where the data collected for each Question is stored in the database.

In Oracle Clinical, the annotated CRF provides linkage between database objects and the DCI and DCM definitions, and DCM Questions and Question Groups, as these definitional objects are laid out in the associated DCI layout.

Requirements

You must have Adobe Acrobat 6.0.3 installed on your system in order to generate annotated CRFs. In addition, the current version of the OPA PDF plug-in must be installed. Note that the PDF DEW that you use to manage annotations is comprised of the Adobe Acrobat and OPA plug-ins running in an instance of Microsoft Internet Explorer. If you open an annotated CRF file in the Adobe Acrobat window, the RDC toolbar will not be displayed.



Note:

Oracle recommends setting your screen to 16 bit or greater color quality; this ensures proper display of the Text and Background color buttons.

Annotation Placement Styles

There are two "styles" for placing the annotations on the form: **In Place** and **Index**. You use the Annotation Placement Setting setting to determine which to use. There are certain differences between the two that may affect which you select.

Procedures

In this section:

- [Annotating a DCI Form](#)
- [Annotating all DCI Forms in a Single Study](#)

- [Working with an Annotated CRF in Adobe Acrobat](#)

Annotating a DCI Form

The annotated CRF that this process described here generates, provides all of the components that are required for an annotated CRF, suitable for inclusion in regulatory submissions. You initiate the process of creating an annotated CRF from the Graph Layout Editor (GLE).

The procedure described here is also applicable if you want to annotate a Form Layout Template (FLT) or a DCM layout.

For more information, see:

- [Procedure](#)
- [Notes](#)

Procedure

To create an annotated DCI form:

1. Open the DCI form in the GLE and click the **Create Annotated Layout** command in the File menu. The system opens the Print Preview Annotated Layout window.
2. Select a report server and click the **OK** button. The system opens the PDF in an Internet Explorer instance, with the RDC Toolbar displayed.
3. Click **Annotated CRF** on the RDC toolbar. The system opens the CRF Annotation Tool window; see [Using the CRF Annotation Tool Window](#).
4. In the CRF Annotation Tool window, select the options that will display the required annotations in the form.
5. Click the **OK** button. The system creates the annotations as PDF comments and displays them in the CRF.

Notes

1. Because the annotations are Adobe Acrobat comments, you can use any of the tools that are available in Acrobat to change how annotations are displayed, printed, or saved.
2. If you want to change how any of the annotations are constructed or displayed, open the CRF Annotation Tool, change settings, and click the **OK** button to re-display the annotations.
3. Use the Adobe Acrobat **Save** button to save the annotated CRF as PDF file.

Annotating all DCI Forms in a Single Study

To generate all Annotated DCI Forms in a specified Study (all Form versions of all DCI's) in a single request, perform the following:

1. In a web browser, enter the URL for the upgrade utility. For example, `https://<Oracle Clinical 5.2.2 compatible application server>:<Port>/opa53/ocdcitemplate.htm`.

 **Note:**

Make sure you have access to this feature. Instructions are available in section B.1 of the *Oracle Clinical Remote Data Capture Onsite Administrator's Guide*.

2. Log in to the upgrade utility using your Oracle Clinical user ID and password.
The Study Level CRF Generation Tools form lists all the DCI Form Definition Enabled studies that your user ID can access.
3. Select a study by clicking a row. Make sure the entire row is highlighted in blue to indicate that the study has been selected. Although you can select multiple studies, Oracle recommends that you generate forms for only one study at a time.
4. Click **Annotated Layouts**.
5. In the Annotated Layouts window, select the desired Report Server.
6. Click **Continue**.

A job is initiated to generate an Annotated CRF for every DCI layout in the study for which a DCI Form Version has been generated previously. Note that this job does not run in the background and will take some time.

The Generation Status field in the lower left of the form shows each DCI form version as it is being processed. On the same form, when the job completes, you can select View Log to see whether the application encountered any errors during the execution of the job.

Annotated CRF's generated through the Study-level option will remain as individual output files (that is, not a single batched output for the entire Study) created in the specified Report Server's log directory.

Working with an Annotated CRF in Adobe Acrobat

When you generate an annotation from the Graphical Layout Editor, Oracle Clinical creates a standalone PDF file that you can manage as needed. This includes adding comments to the file that are in addition to the annotations. Essentially, you can use all of Adobe Acrobat's functionality to further modify the annotated CRF file.

For more information, see:

- [Displaying Acrobat Toolbars](#)
- [Saving an Annotated CRF](#)
- [Opening a Saved Annotated Form](#)
- [Methods of Organizing and Viewing Annotations](#)

Displaying Acrobat Toolbars

Because annotated CRFs are displayed in the PDF DEW, you do not have access to the Adobe Acrobat menus. Therefore, you should display several of the Acrobat toolbars so you can access certain features that assist in the review and maintenance of your annotation files. You may find these toolbars helpful:

- Basic

- Commenting
- File
- Tasks

To access any toolbar, right-click in the toolbar area and select an item in the drop-down list. You can also move the toolbars once they are displayed.

Saving an Annotated CRF

You save an annotated CRF in the same way that you save any other PDF file in Adobe Acrobat. You can save multiple copies of the annotated PDF.



Note:

Ensure the **File** toolbar is displayed, which gives you access to the **Save Copy** button.

To save a copy of the annotations, click the **Save a Copy** button, which is in the Acrobat File toolbar. The system displays the **Save a Copy...** window, which you use to select a directory and file name for the new file. The file is saved in PDF format.

Opening a Saved Annotated Form

When you create an annotated form, the system creates a PDF file that you can then save, copy, and modify further. When you save and close the annotated form and the PDF DEW closes, use this procedure to re-open it.

1. Open Microsoft Internet Explorer.
2. Choose **Open...** from the **File** menu.
3. In the Open window, use Browse to locate the saved annotated form and open it. The form will have a *.pdf extension.
4. When the form opens, you have access to the Acrobat tools and the PDF DEW Annotation. Use either to review or modify the annotated file.

Notes

1. When the form opens in the browser window, the Adobe Acrobat and the OPA plug-ins are activated. This gives you access to all of Acrobat's functionality, as well as that of the CRF Annotation Tool. In order to use the CRF Annotation Tool, the form must be opened within a Microsoft Internet Explorer window.
2. You can open an annotated PDF file in Adobe Acrobat, however the RDC toolbar, which is a component of the PDF DEW and includes the **Annotated CRF** button, is only available when you use Internet Explorer to open the file.

Methods of Organizing and Viewing Annotations

Because annotations are PDF comments, you can use the Adobe Acrobat Comments List feature to list, review, and modify them. You access the Comments List either by clicking the **Comments** tab in the lower left-hand corner of the PDF DEW or the **Comments List** command in the Review & Comment drop-down list.

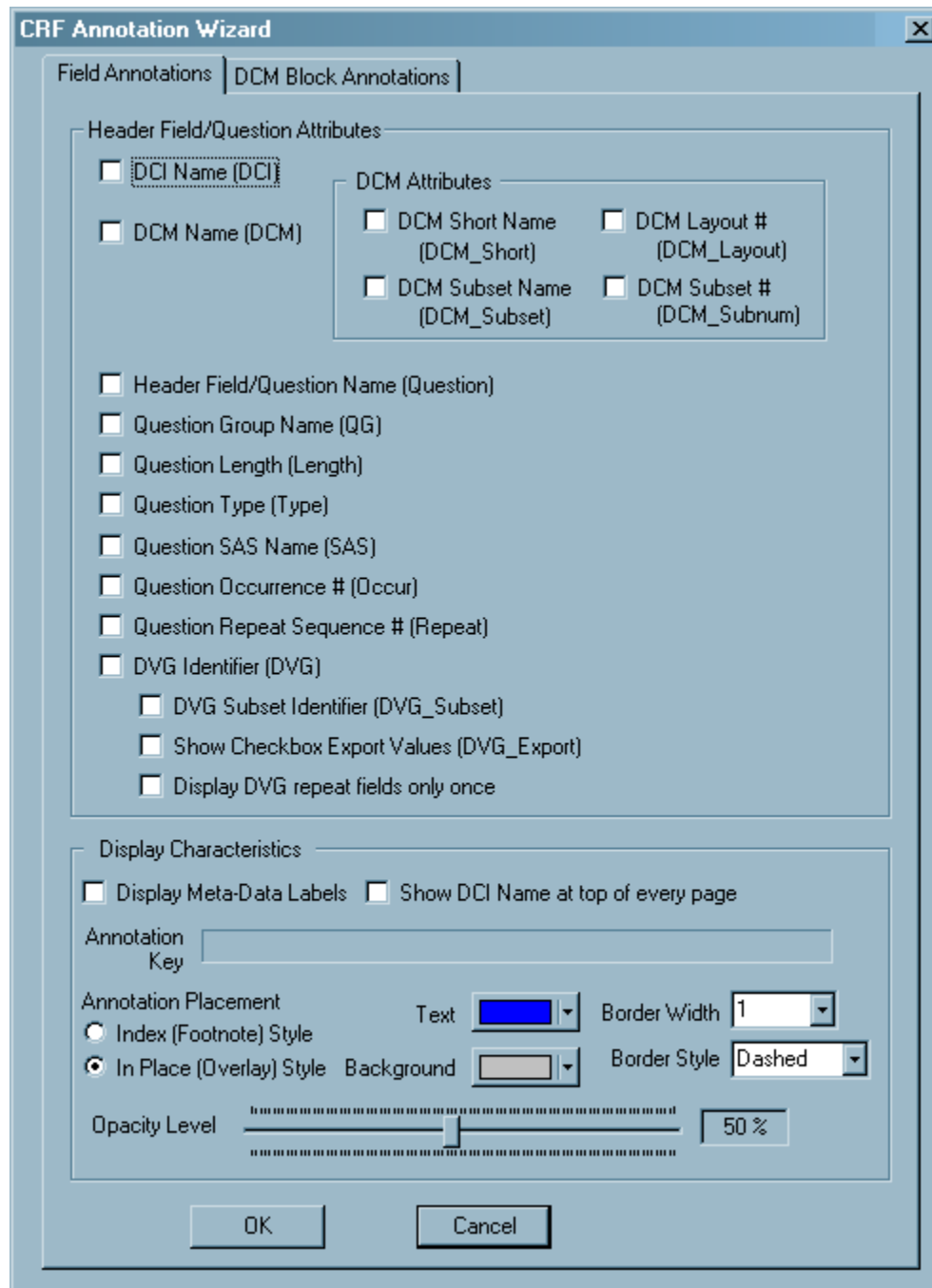
Using the CRF Annotation Tool Window

When you generate an annotated layout, as described in [Procedures](#), the GLE opens the form in version of the PDF data entry window (DEW) that provides the capability to annotate the form. In this context, only the **Annotation Tool** button is active in the DEW.

 **Note:**

The CRF Preferences tool is also active in the PDF DEW, however, none of its settings affect the annotated CRF.

Figure 11-1 CRF Annotation Tool - Field Annotations tab



For more information, see:

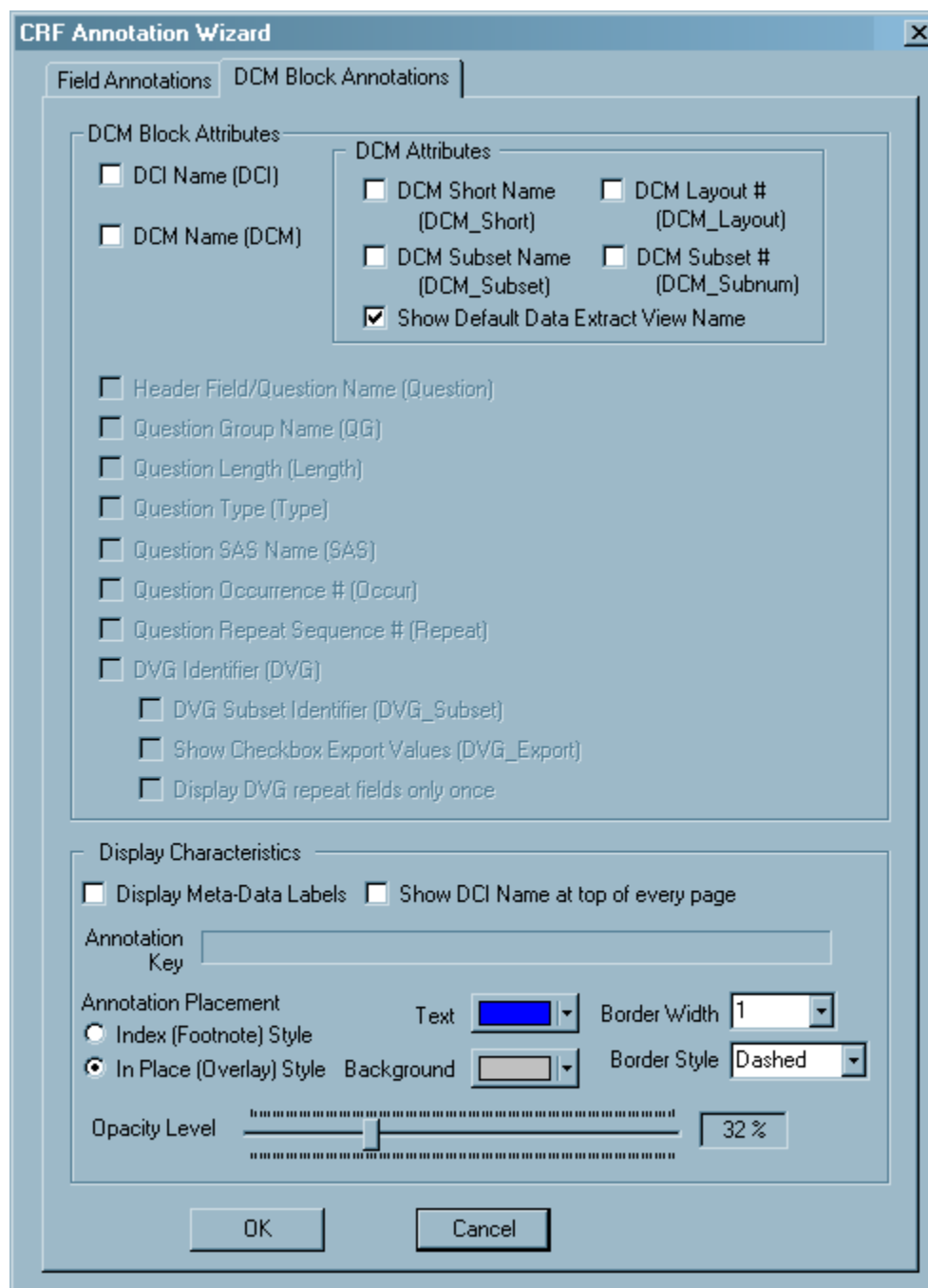
- [Access](#)
- [Components](#)
- [DCM Attributes Section](#)

Access

You access CRF Annotation Tool by clicking **Annotate CRF** on the RDC Toolbar in the DEW. This button is only active when you use the Create Annotation action in the GLE.



Figure 11-2 CRF Annotation Tool - DCM Block Annotations Tab



Components

The CRF Annotation Tool window is comprised of two tabs: Field Annotations and DCI Block Annotations. As [Figure 11-1](#) and [Figure 11-2](#) illustrate, the set of configurations on each tab are similar. However, there are certain differences, which are based on the what each type of annotation entails.

Both tabs include a Display Characteristics section, which allows you to control certain formatting aspects and the location of the annotations. The Field Annotations tab includes a Header Field/Question Attributes section, which you use to configure annotations associated with the DCI and DCM header data fields. The DCM Block Annotations tab includes a DCM Block Attributes section. This section includes settings for include DCI and DCM names, and a subsection that consists of settings for DCM attributes. The section also displays settings from the Field Annotations tab in read-only mode.

DCM Attributes Section

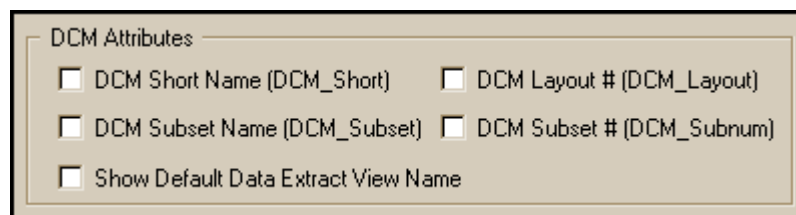
The DCM Attributes section, which is present on each tab, controls DCM-specific annotations at the header/Question field or the DCM block level, depending on which tab you use. The graphic in [Figure 11-3](#) depicts the section that is present in DCM Block Annotations tab. The section in the Field Annotations tab is identical, with the exception of the Show Default Data Extract View Name setting, which is only present in the DCM Block Annotations tab.

The settings that are the same on each tab are: DCM Name, DCM Short Name, DCM Layout and DCM Subset. The settings on each tab are independent of one another. The fundamental difference between the tabs is in determining which annotation is affected.

The DCM Attributes settings in the:

- Field Annotation tab control whether the item is included in the annotations for each DCM field
- DCM Block tab control whether the item is included in the annotation that is displayed at the top of each DCM block.

Figure 11-3 DCM Attributes Section – DCM Block Annotations Tab



For more information, see:

- [Components of the Field Annotations Tab](#)
- [Components of the DCM Block Annotations Tab](#)
- [Components of the Display Characteristics Section](#)

Components of the Field Annotations Tab

The Field Annotations tab, which is depicted in [Figure 11-1, "CRF Annotation Tool - Field Annotations tab"](#), consists of two sections: the Header Field/Question Attributes section, which you use to configure annotations at the field level, and the Display Characteristics section, which you use to modify how these annotations are displayed. The settings in the Header Field/Question Attributes section are listed and described in [Table 11-2](#). Each listed setting is selected using a check box. The meta-data name that corresponds to each attribute is in parentheses after the corresponding annotation label.

Table 11-1 Header Field/Question Attributes Section

Label	Description
DCI Name (DCI)	Includes the DCI Name in each DCI header, DCM header, and Question field annotation. This is the only attribute that is not applicable to a form originating in the DCM layout Editor.
DCM Name (DCM)	Includes the DCM name in each DCI header, DCM header, and Question field annotation.
DCM Short Name (DCM_Short)	Includes the DCM Short Name (if available) in each DCI header, DCM header, and Question field annotation.
DCM Layout # (DCM_Layout)	Includes the DCM layout number in each DCI header, DCM header, and Question field annotation.
DCM Subset Name (DCM_Subset)	Includes the DCM Subset Name in each DCI header, DCM header, and Question field annotation.
DCM Subset # (DCM_Subnum)	Includes the DCM Subset number in each DCI header, DCM header, and Question field annotation.
Header Field/Question Name (Question)	Includes the Question name or the header field name in the corresponding annotation for each DCI header, DCM header, and Question field annotation.
Question Group Name (QG)	Includes the Question Group name in each Question field annotation.
Question Length (Length)	Includes the Question length value in each Question field annotation. In an NLS Study, this value is the length defined in NLS application, which may differ from Question Length defined in Oracle Clinical.
Question Type (Type)	Includes the Question type value in each Question field annotation
Question SAS Name (SAS)	Includes the SAS name in each Question field annotation.
Question Occurrence # (Occur)	Includes the Question occurrence number in each Question field annotation.
Question Repeat Sequence # (Repeat)	Includes the Question repeat sequence number in each Question field annotation.

Label	Description
DVG Identifier (DVG)	For a repeated DVG in a repeating Question Group, includes the DVG identifier in each Repeating Question field annotation.
DVG Subset Identifier (DVG_Subset)	Includes the DVG subset identifier in each Repeating Question field annotation.
Show Checkbox Export Values (DVG_Export)	For a DVG field that is represented by a checkbox group, includes each checkbox export value in each Repeating Question field annotation. In an NLS Study, the DVG_Export values are the Local Language translation that is defined in Discrete Value Translation Form in the OC/NLS Application.
Display DVG repeat fields only once	For repeated DVG entries, this setting limits the inclusion of the DVG identification information to the first Question field annotation in the group.

Components of the DCM Block Annotations Tab

The DCM Block Annotations tab, which is depicted in [Figure 11-2, "CRF Annotation Tool - DCM Block Annotations Tab"](#), controls the annotations that are displayed at the start of each DCM block. It consists of two sections: the DCM Block Attributes section, which you use to configure annotations at the DCM Block level, and the Display Characteristics section, which you use to modify how these annotations are displayed. The components of the DCM Block Attributes section is further divided into the DCM Attributes Section, which is described above, and two other settings, which are listed and described in [Table 11-2](#). Each setting is selected using a check box. The meta-data name that corresponds to each attribute is in parentheses after the corresponding annotation label. Unless noted otherwise, each setting is present on both tabs.

Table 11-2 Components of the DCM Block Attributes Section

Label	Description
DCI Name (DCI)	Includes the DCI Name in each DCM Block annotation. This is the only attribute that is not applicable to a form originating in the DCM layout Editor.
DCM Name (DCM)	Includes the DCM name in each DCM Block annotation.
DCM Short Name (DCM_Short)	Includes the DCM Short Name (if available) in each DCM Block annotation.
DCM Layout # (DCM_Layout)	Includes the DCM layout number in each DCM Block annotation.
DCM Subset Name (DCM_Subset)	Includes the DCM Subset Name in each DCM Block annotation
DCM Subset # (DCM_Subnum)	Includes the DCM Subset number in each DCM Block annotation.
Show Default Data Extract View Name	Includes the name of the default data extract view in DCM Block annotation.

Components of the Display Characteristics Section

The Display Characteristics section is present on each tab. Its components are listed and described in [Table 11-3, "Display Characteristics Section"](#). The settings in these sections are identical in both tabs, however you manage the display of annotations independently on each tab. This allows you set up the display of header-level annotations differently then field-level annotations.

Table 11-3 Display Characteristics Section

Label	Description
Display Meta-Data Labels Setting	Displays the meta-data name in angle brackets with each annotated component.
Show DCI Name at top of every page Setting	Displays an annotation with the name of the DCI at the top of each page.
Annotation Key Field	A display-only field that provides a real-time representation of the annotations, based on the choices you have made for other settings.
Annotation Placement Setting	A radio button setting that allows you to choose between the two types of annotation display; either Index or In Place
Text Setting	A drop-down that displays a color-chooser palette, which you use to select the color that you want the text of each set of annotations.
Background Setting	A drop-down that displays a color-chooser palette that you use to select the color that you want for the background of each annotation.
Border Width	A drop-down list that you use to select the thickness of border around each annotation. The values are in pixels. The choices are: 0, 1, or 2 pixels.
Border Style	A drop-down list that you use to select the type of line that is used for the border. The choices are Dashed or Solid.
Opacity Level	A slider that you use to select the opaqueness of each annotation. This affects how much underlying text you can view through the annotation text and background. Measured as a percentage, with 5% as the dimmest and 100% as the darkest display of the annotation.
OK	Use this button to generate an annotated CRF with the current settings.
Cancel	Use this button to revert to the PDF DEW with changing settings.

11.3.3.3.1 Display Meta-Data Labels Setting

This setting causes the meta-data name of each object to be included in the annotation key string. It is active on both tabs and controls both types of annotations: when you change this setting on one tab, it is also changed on the other.

When you change this setting, its effect is immediately displayed in the [Annotation Key Field](#). The meta-data names are appended to each relevant setting in the window. Selecting this setting causes each annotation to include the meta-data identifier for

each setting in the annotation string and to place the actual value in angle brackets, with a colon between the two items.

For example, if you select the following settings in the Field Annotations tab and do not select the Display Meta-Data Labels setting, the resulting dot notation in the Annotation Key field is displayed in the top graphic in [Figure 11-4](#):

- [DCI Name \(DCI\)](#) – DEMOG
- [DCM Name \(DCM\)](#) – DEM
- [DCM Short Name \(DCM_Short\)](#) – PHYS
- [DCM Subset Name \(DCM_Subset\)](#) – BP

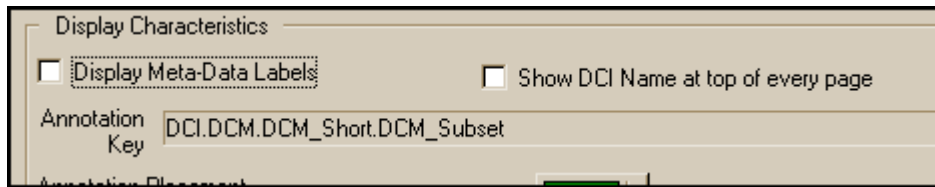
The actual annotation that is displayed adjacent to each field is:

DEMOG.DEM.PHYS.BP

When you select the Display Meta-Data Labels setting, the resulting dot notation in Annotation Key field is displayed in the lower graphic in [Figure 11-4](#). The actual annotation that is displayed adjacent to each field is:

DCI:<DEMOG>.DCM:<DEM>.DCM_Short:<PHYS>.DCM_Subset:<BP>

Figure 11-4 Example Usage of the "Display Meta-Data Labels" Setting



Description of "Figure 11-4 Example Usage of the "Display Meta-Data Labels" Setting"



Description of "Figure 11-4 Example Usage of the "Display Meta-Data Labels" Setting"

Show DCI Name at top of every page Setting

This setting displays the name of the DCI at the top of each page in the annotated form. It is active on both tabs and controls both types of annotations: when you change this setting on one tab, it is also changed on the other. This setting is useful for CRFs that are longer than one page, where you do not want to include DCI Name (DCI) setting with each field annotation.

Annotation Key Field

This field is not a setting, but it displays a generic string, in dot-notation format, that is based on the settings that are currently selected in the tab. As you change settings, the contents of the string change, showing you how much information will be included in each annotation, based on the choices you have made in the window. It also changes the generic string based on the [Display Meta-Data Labels Setting](#).

Annotation Placement Setting

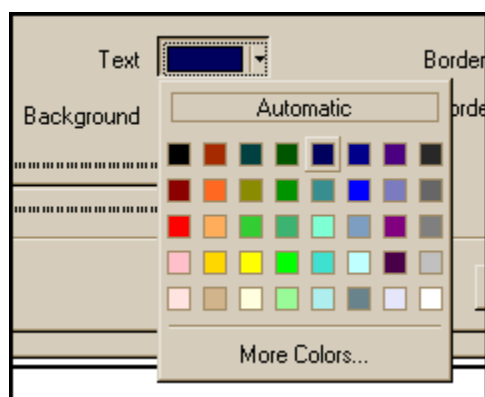
This setting determines how the annotations are displayed on the CRF page. The choices are Index and In Place. Index style uses numbered footnotes to link to the actual annotations, which you view using the Acrobat Comments tool. In Place style displays each annotation adjacent to the corresponding data field.

This setting is only active on the Field Annotations tab, however it controls how all annotations is placed in the form (see "[Components of the Field Annotations Tab](#)").

Text Setting

This setting allows you to specify the font color that is used for the annotations, using a standard color palette to choose. The font color can be different for Field and the DCM annotations, so you set this for each type of annotation. Using different text and background colors is useful for differentiating between the different types of annotations.

Figure 11-5 Color Palette for Text Setting



Description of "Figure 11-5 Color Palette for Text Setting"

Note:

If the Color Quality setting for your monitor is not set to 16 bit or greater, the color display for the Text and Background button may not reflect your choice in the Color Palette tool. However, when you generate the annotations, the correct color is displayed in the PDF.

Background Setting

This setting allows you to specify the background color that is used for the annotations, using a standard color palette to choose. The color can be different for Field and the DCM Block annotations, so you set the background color for each type of annotation. The standard color palette is depicted in [Figure 11-5](#).

12

Creating Graphic Layouts

This chapter outlines the process of producing Graphic Layouts.

In Oracle Clinical, the term *layout* means the graphic and typographic representation of a data collection definition. (See [Data Entry and Report Layout Design](#) for a comparison of the graphic-based layout system to the character-based system.) Here are some of the Graphic Layout system's advantages:

- You can use Graphic Layouts to produce these outputs:
 - You can create DCI Forms for the RDC Data Entry Option.
 - You can print out formatted blank CRFs.
 - You can produce formatted Patient Data Reports (PDRs).
- You lay out DCMs and DCIs separately in order for these data collection definitions to retain their flexibility for re-use.
- If you revise a Graphic Layout after you start collecting data—for instance, if you add a Question or modify the layout—you can create a new layout version. (See [Maintaining Form Layout Versions](#).)

Graphic Layout Terms

These are the main Graphic Layout terms:

- **DCM Graphic Layouts** are DCM definitions that include visual formatting. The Questions in a DCM Graphic Layout are collected into *blocks*.
- **DCI Graphic Layouts** contain DCM Graphic Layouts and header and footer information.
- **DCI Forms** are used to generate the CRF for RDC Onsite HTML data entry window and the PDF Patient Data Report (PDR) and Blank Casebook Report. A DCI Form can contain up to 2000 objects.

Note:

Each page must fit on a printed page or you will not be able to see all the data in the Patient Data Report. The system converts the layout separately for the PDF reports and HTML data entry. When you create a layout, test both uses.

- **Form Layout Templates** contain DCI Graphic Layouts and define the areas for DCIs, headers, and footers.
- **Page Definitions** determine the overall dimensions of a CRF or PDR printed page or window.
- **Objects** are text and graphic elements like fields, lines, boxes, and prompts. There are **input objects** you inherit from the DCIs and DCMs, and **static objects** you add to a layout.

[Figure 12-1](#) is a diagram of the basic process for creating DCI Forms.

For more information, see:

- [Creating a DCI Form](#)
- [DCI Form Layout Areas](#)
- [Enabling DCI Forms](#)
- [Configuring Study-Level DCI Form Settings](#)
- [Configuring DVGs and DCI Modules](#)
- [Maintaining DCM Graphic Layouts](#)
- [Maintaining DCI Graphic Layouts](#)
- [Maintaining Form Layout Templates](#)
- [Editing Layouts](#)
- [Creating and Maintaining Conditional Blocks](#)
- [Troubleshooting Graphic Layout Problems](#)

Creating a DCI Form

As an introduction to the Graphic Layout system, here are steps that utilize default values and starter definitions to produce a basic DCI Form. See [Figure 12-1](#).

1. Enable DCI Forms in a study: Navigate to **Conduct, Security, and Clinical Study States**, and select the **DCI Form Definition Enabled** and **DCI Forms Entry Enabled** boxes for a study; see [Enabling DCI Forms](#).
2. Create or select an active DCM; see [Defining DCMs](#).
3. Generate a DCM Graphic Layout: In the DCM window, select **Special**, then choose **Graphic Layout**, then **Generate Default from Scratch**. Accept the defaults and click **Generate**. The system creates a Graphic Layout definition of the DCM; see [Generating Default DCM Graphic Layouts](#).

Alternatively, if you already have a character layout, select **Generate Default From Character Layout**.

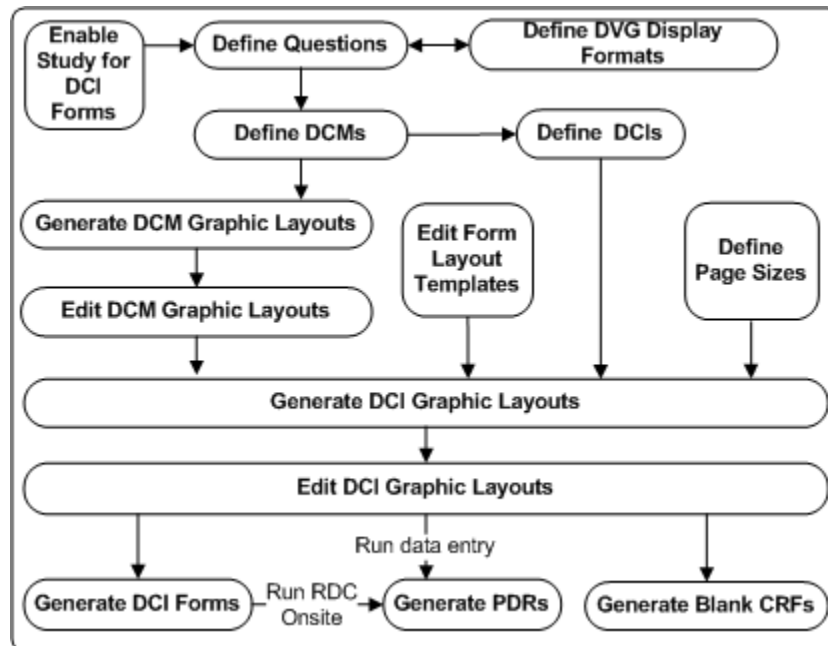
You can now edit the DCM Layout; see [Editing DCM Graphic Layouts](#). While in the editor you can generate a preview of the layout; see [Previewing DCM Graphic Layouts](#).

4. In the DCMs window, click the Graphic Layout tab, select the DCM's **Available?** box, and **Save**.
5. If necessary, create a default DCI, or add the DCM to an existing DCI; see [Defining DCIs](#).
6. Generate a default DCI Graphic Layout from the DCI (see [Generating Provisional DCI Layouts](#)):
 - a. From the **Special** menu, select **DCI Layout**, then **Generate Provisional Layout**. The **Attributes for DCI Graphic Layout Generation** dialog box opens that includes several modifiable settings that are explained later.
 - b. Click the **Generate** button. The system creates a definition of the DCI's data and formatting characteristics. You can now edit your DCI Layout; see [Maintaining DCI Graphic Layouts](#).

At this point, you can generate a preview of the blank CRFs that you can print.

7. Generate a DCI Form: Select the DCI with the DCI Layout you generated in the last step, and navigate to **Special, DCI Layout, and Generate Provisional DCI Form**. Click **Generate**. The system creates the DCI Form.
8. Select your DCI and click the **DCI Form Versions** button. If this is the first time you have generated a DCI Form from this DCI, it is Version 1. Change its **Status** value from P(rovisional) to A(ctive), save, and click the **Back** button.
9. If necessary, add your DCI to a DCI Book. When you are satisfied with the form, change the Book's **Status** to A(ctive); see [Defining DCMs](#) .

Figure 12-1 Graphic Layout Process Workflow



DCI Form Layout Areas

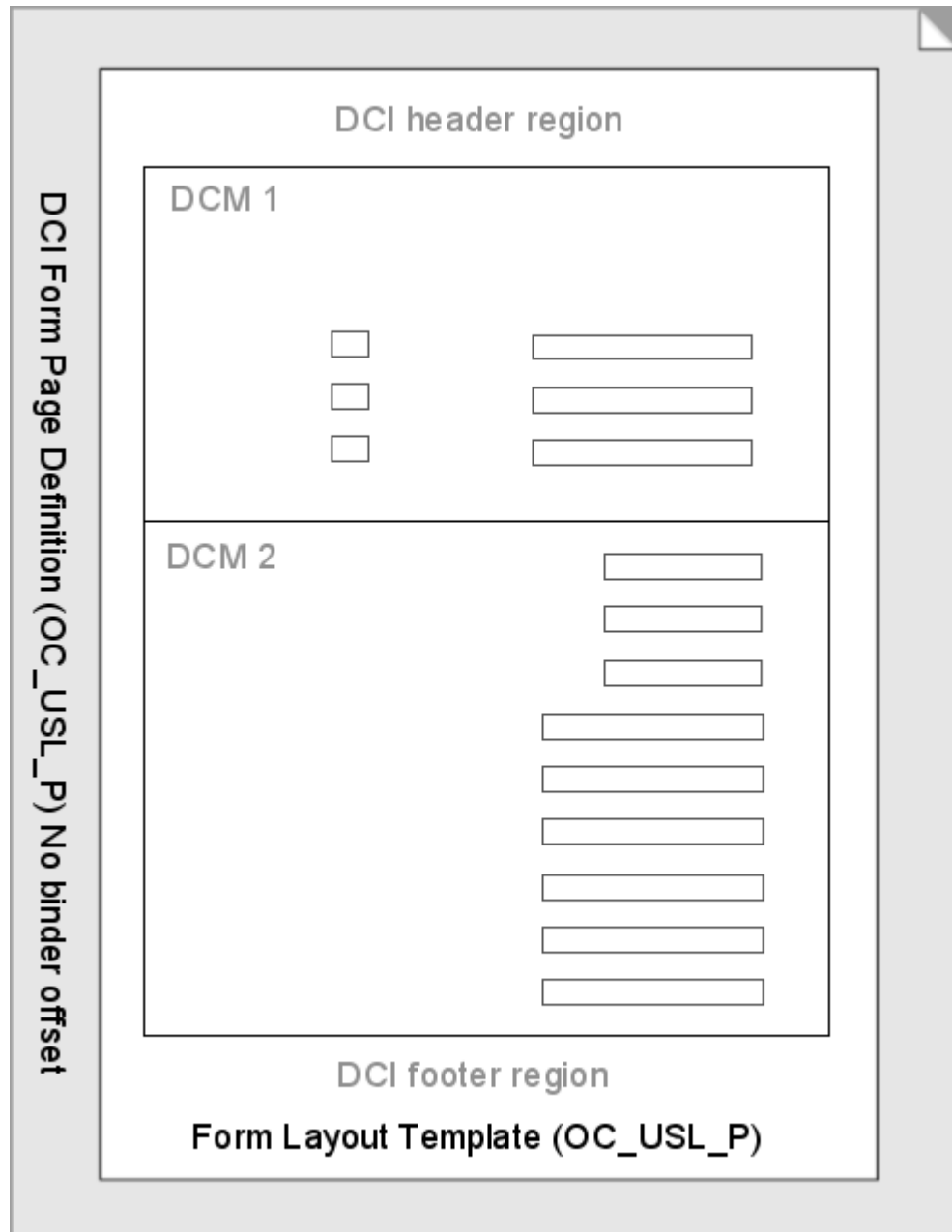
This section introduces the DCI Form's graphic terrain by describing its parts. When you generate a DCI Form, you render the set of nested data definitions as graphic representations. Each representation fits into an area. From the interior to the edges, here are the relationships of the areas:

- Questions fit into *blocks* in DCM Layouts.
- DCM blocks fit into *DCM regions* of DCI Layouts.
- DCI Layouts inherit the dimensions of their associated *Form Layout Templates*. Form Layout Templates define the dimensions that can hold the header, footer and DCM regions.
- The DCI Form Generator positions Form Layout Templates into *DCI Form Page Definitions*. The Page Definition specifies the size of the page. The *Binding Offset* determines the positioning of the Form Layout Template within the page.

The following sample DCI Form Layout illustrates the layout areas of a DCI Form based on a starter DCI Form Page Definition in portrait orientation (OC_USL_P), and the default US_USL_P Form Layout Template. Because of the difference in size between the Page

Definition and the Form Layout Template, it has top and bottom margins of 36 points (1/2 of an inch) each, and side margins of 54 points (3/4 of an inch) each.

Figure 12-2 DCI Form Layout Areas



For more information, see:

- [Creating DCI Form Page Definitions](#)

Creating DCI Form Page Definitions

A Page Definition is the paper size in print, or the overall form size on screen. Oracle Clinical provides starter definitions for the standard paper sizes. However, you can

define pages for custom paper sizes or to fit on a particular screen device. In [Figure 12-2](#), the visible part of the Page Definition is light gray, and extends to the edges of the illustration. This section describes how to create them.

For more information, see:

- [Defining Page Definitions](#)
- [About Points](#)
- [About the Starter Page Definitions](#)
- [Sizing Margins](#)
- [Setting Binding Offsets](#)

Defining Page Definitions

You maintain Page Definitions in the DCIF PAGE DEFINITION Installation Codelist: From the **Admin** menu, select **Reference Codelists**, choose **Installation Codelists**, and query DCIF PAGE DEFINITION.

You define a page's dimensions by entering a codelist long value with this syntax:

```
HEIGHT:x WIDTH:x BINDING OFFSET:x
```

In the above syntax line, *x* is a value in points.

About Points

The only available unit of measure for Page Definitions is *points*. Points have these properties:

- There are 72 points to an inch, 28.3 points to a centimeter.
- A standard A4 page is 494.4 by 842 points.
- A standard US letter (8 1/2 inches by 11 inches) page is 612 by 792 points.
- Points and screen measurements have a relative correspondence. A layout appears much larger on a monitor screen set to 1280 by 1024 pixels than it does on a screen set to 1600 by 1200 pixels. Although all units of measure—including point measurements—are relative on the screen, they are absolute in print.

About the Starter Page Definitions

Oracle Clinical provides starter Page Definitions that you can in any DCI Form based on standard A4 and US Letter page sizes. There are landscape and portrait orientation definitions for each page size. Their names indicate their page size and orientation. The starter pages have no binding offsets, which means that the DCI Form Generator centers Form Layout Templates inside them. These are the Page Definition names and the long values that define them:

- **OC_USL_P** HEIGHT:792 WIDTH:612 BINDING OFFSET:0
- **OC_USL_L** HEIGHT:612 WIDTH:792 BINDING OFFSET:0
- **OC_A4_P** HEIGHT:842 WIDTH:595.4 BINDING OFFSET:0
- **OC_A4_L** HEIGHT:595.4 WIDTH:842 BINDING OFFSET:0

Sizing Margins

A DCI Form's margin area is equal to the difference between the Page Definition size and the Form Layout Template size. Most organizations standardize on a particular page size, so determining the margins in a layout most likely requires defining the size of your Form Layout Templates. See [Sizing Form Layout Templates](#) for sizing Form Layout Templates.

Sizing Left and Right Margins

When you generate a DCI Form, the DCI Form Generator subtracts the Form Layout Template's width from the layout's designated DCI Form Page Definition width. It then divides the difference by two to determine the width of the left and right margins. In [Figure 12-2](#), the example has side margins 54 points (3/4 inches) each. A standard U.S. letter is 612 points wide. The starter Form Layout Template OC_USL_P is 504 points wide. This leaves 108 points difference in the width, or 54 points in each long margin. For portrait orientation, you can shift the margin to the left or right. See [Setting Binding Offsets](#).

Sizing Top and Bottom Margins

The system performs the same operation on the height. In [Figure 12-2](#), the example has a Page Definition height of 792 points and a Form Layout Template height of 720 points. The difference is 72 points, so the top and bottom margins are each 36 points high. For landscape orientation, you can shift the margin up or down. See [Setting Binding Offsets](#).

Setting Binding Offsets

You can shift the template along one axis by changing the Page Definition's Binding Offset value. The binding offset parameter enables you to increase the gutter space in a PDR DCI form. The list of eligible Page Definitions for a DCI Layout or PDR generation is limited to Page Definitions with binding offsets that fit.

The DCI Form Generator subtracts the offset value from the Portrait Page Definition's short dimension, then re-centers the Form Layout Template on the reduced axis. Therefore, adding a binding offset requires changing the offset value in the Page Definition and adjusting a dimension of a Form Layout Template.

The binding offset is always along the long dimension of the Page Definition, regardless of the page orientation. You can use positive or negative values. If you define a square page with a binding offset, the system offsets the Form Layout Template to the left or right.

Setting Portrait Offsets

For portrait orientation layouts, a positive offset value shifts the Form Layout Template toward the right side of the page, increasing the left margin. A negative value shifts the Form Layout Template toward the left side of the page, increasing the right margin. For example, if the left margin is 26 without an offset, an offset of negative six (-6) changes the left margin to 20 points.

Setting Landscape Offsets

For landscape orientation layouts, a positive offset value shifts the Form Layout Template toward the top of the page, increasing the bottom margin. A negative value shifts the template toward the bottom of the page, increasing the top margin. For

example, if the top margin is 40 without an offset, an offset of 10 changes the top margin to 30 points.

Enabling DCI Forms

You enable the definition of DCI Forms and data entry using DCI Forms at the study level by logging in to Oracle Clinical and opening the Clinical Study States form: Navigate to **Conduct, Security**, and then **Clinical Study States**. Query and select your study, then select its **DCI Forms Definition Enabled?** and **DCI Forms Entry Enabled?** boxes. Save your work and exit.

Note:

You must download the latest version of the Oracle Clinical PDF Plugin from My Oracle Support, as described in the *Oracle Clinical Installation Guide*, to use the Annotated Layout feature.

You can enable the definition of DCI Forms by default for all studies on a local database through a local database setting. If you do this, when you create a study through Easy Study Design, you do not have to enable the study in the Clinical Study States window. From the **Admin** menu, select **DCI Form Local Database Settings**, then **DCI Form Definition**. Set **DCI Forms Definition Enabled** to Yes. See the *Oracle Clinical Administrator's Guide* for a description of local database configuration.

Configuring Study-Level DCI Form Settings

Once you have enabled DCI Form definition and collection, you configure default generation settings such as default layout templates and header prompt text.

You can configure the default generation settings for a study in the DCI Form Local Study Settings form. To open the form, log in to Oracle Clinical and navigate to Design, then open the DCI Form Local Study Settings menu item.

About the DCI Form Local Study Settings Form

The DCI Form Local Study Settings form organizes the configuration settings into categories. The following list describes the settings by category. (See section, "DCI Form local database settings" in the *Oracle Clinical Administrator's Guide* for more information.)

- An unlabeled column that contains plus (+) or minus (-) symbols. Clicking a cell in this column displays or hides a category's settings.
- **Setting** This column contains several general configuration categories and their configuration settings. You must click the plus symbol to the left of a category to display its settings. The categories are headings: you do not set any values in the category rows.
- **Value** This column displays the current values for the settings. If a cell is blank, its value has not been set.
- An unlabeled column that includes list of value (LOV) buttons for list of values-controlled settings. If a setting does not have an list of values, its LOV cell is inactive.

- **Inherit From Local DB Setting**, if selected, indicates that a setting is using the default setting set at the local database level. If you enter a value different from the default value, the system deselects this field. If you are unable to deselect a setting's **Inherit From Local DB Setting** box or change a value for a setting, the system is preventing you from changing the value because it is enforcing the default value at the local database level. See the *Oracle Clinical Administrator's Guide* for more information about database-level DCI Form configuration.

 **Note:**

Changing the database-level setting does not override study settings already in effect. It enforces settings for studies that are created after the change.

- **Local DB Value** displays the default values.

For more information, see:

- [Configuring Local Studies](#)

Configuring Local Studies

The following sections describe each of the DCI Form Local Study Settings form's configuration categories:

- [DCI Form Runtime](#)
- [General](#)
- [Graphic Layout Editor/Updater](#)
- [Graphic Layout Generator - General](#)
- [Graphic Layout Generator - DCMs](#)
- [DCI Form Generation Defaults](#)
- [Default Settings for Showing DCM Header Fields](#)
- [Default DCM Header Field Prompts](#)
- [Version Migration](#)
- [Patient Data Report](#)
- [Validation](#)
- [Options in Group Verify dialog](#)
- [Options in Group Approve dialog](#)
- [Source Data Verification Plan Defaults](#)

DCI Form Runtime

The settings in this category are:

- **Disable Delete CRF Icon** Select **Y** to disable the CRF icon in the RDC Onsite Data Entry window. Select **N** to enable it. When the icon is enabled, users can click it to delete the CRF and its data.

- **Display Label for DCM Question** Select the source for the label of each field in each DCM; either the **SAS label**, the **Question name**, or the **default prompt** of the corresponding Question definition.
- **Display Visit Owing Interval on MPC Page?** If set to **Y**, the multipatient casebook page in RDC Onsite displays the Interval—phase, period, or subperiod—of which each visit is a part.
- **Enable Entry of Investigator Comments** If set to **Y**, Investigator comments are allowed in RDC Onsite.
- **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) for the field in the Patient Positions table in RDC Onsite, enter the label text you prefer.
- **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; see [Using CRF Page Tracking](#) for more information.
- **Represent Disabled Blocks as** This setting applies to studies using conditional in-form branching. The possible settings are:
 - **Hidden** The horizontal section of the form representing the conditional target block is completely hidden, as if the form is 'collapsed' or folded. If there are questions that appear on the form after the conditional blocks, those questions appear immediately after the conditional source question. For multi-page forms, the locations of page breaks are preserved relative to questions in the form. Therefore, when blocks are hidden, questions from the next page are not moved to fill the resulting additional blank space on the page.
 - **Greyed** The horizontal section of the form representing conditional target block is visible but inactive.
- **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.
- **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.
- **Use Customizable Patient Identifier?** If set to **Y**, the system displays the Reference field in the Patient Positions table in RDC Onsite. You can change the label for the field using the **Customizable Patient Identifier Label** setting.

General

There is one setting for this category: **Default Unplanned Use Allowed for DCIs not in Book**. This setting controls the default value for the **Unplanned Use Allowed for DCIs Not in Book** field on the DCI Book Constraints form. If this field is set to **Y**, then unplanned use of any DCIs defined in the study is allowed. Otherwise, the only DCIs available for unplanned use are those that are already used in the book and which are so designated.

This setting affects only new DCI Books. Changing the setting does not change the DCI Book Constraint setting for an existing DCI Book.

See [Defining DCI Book Constraints](#) for more information on constraining DCI usage.

Graphic Layout Editor/Updater

There is one setting for this category: **Enforce Length as Field Size**. If set to **Y**, the layout editor does not allow the user to drag the field to make it shorter than the character length defined for the Question. When set to **Y**, if you increase the Question's length in the **Study DCM Questions** window, the system sets the **Needs Update** flag for the layout.

Note:

This setting has no effect if the generated size of the field is based on the Question's DISPLAY LENGTH instead of LENGTH, per the **DCM Question Attribute for Determining Field Width** setting; see [Graphic Layout Generator - DCMs](#).

Graphic Layout Generator - General

The settings in this category are:

- **Default Checkbox Check Style** Click the LOV cell to choose a default symbol for check marks. This setting populates the **Checkbox Style** field in the startup dialog for the DCM Graphic Layout Generator. You can choose a check, circle, cross, or square.
- **Default Checkbox Shape** Click the LOV cell to choose a default box shape. This setting populates the **Checkbox Shape** field in the startup dialog for the DCM Graphic Layout Generator. You can choose between a circle and a square.
- **Default Checkbox Size** Click the LOV cell to choose a default box size, in points. There are 72 points per inch, or about 28 points per centimeter. (You specify the list of values by populating reference codelist DCIF CHECKBOX SIZE.)
- **Default Landscape Form Layout Template** This attribute determines the default Form Layout Template during DCI Layout generation, if no portrait Form Layout Templates are large enough. The DCM Layout generation defaults to the Portrait Form Layout Template.
- **Default Portrait Form Layout Template** Click the LOV cell to choose a default portrait Form Layout Template from the listed choices.
- **Field Font Size** Click the LOV cell to choose a monospaced font size, in points. There are 72 points per inch, or about 28 points per centimeter. (You specify the list of values by populating reference codelist DCIF FONT TYPESIZE.)
- **Field Font Typeface** This list is based on the DCIF Typefaces table, which is not modifiable. The only choice is Courier.

Note:

The Layout Editor does not support the Symbol font. The generated layout may show boxes instead of characters.

- **Prompt Font Size** The prompt is an entry field's label. Click the LOV cell to choose a font size, in points. There are 72 points per inch, or about 28 points per centimeter. (You specify the LOV by populating reference codelist DCIF FONT TYPESIZE.)
- **Prompt Font Typeface** This list is based on the DCIF Typefaces table which is not modifiable. The choices include Courier New, Arial, Times New Roman and Symbol.

Graphic Layout Generator - 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 for each Question's response:

- 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 ancillary pages of the report.

Note:

See also the **Enforce Length as Field Size** setting here: [Graphic Layout Editor/Updater](#).

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.

See [Creating DCI Form Page Definitions](#) for more about Page Definitions.

Default Settings for Showing DCM Header Fields

You control the default display of DCM header fields in this category. These settings can be overridden at the DCI module level.

Note that you use next category, [Default DCM Header Field Prompts](#), to define default text for these fields' labels.

The settings in this category are:

- **Default for Show Blank Flag?** Set to **Y** to enable displaying an indicator to set DCMs to Blank status in DCM headers.
- **Default for Show 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?** 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 DCM has any lab Questions. If there are no lab Questions for a DCM, **Show Lab** is set to **N** for DCI Modules regardless of this setting.
- **Default for Show Qualifying Value?** Set to **Y** to enable displaying the DCM qualifying value in the DCM header. If a DCM has no Qualifying Question or has a Qualifying Question with no default value, Show Qualifying Value has a value of **N** for the DCI Module record, regardless of the value of this setting.
- **Default Visit Display Code** This setting determines whether and how the subevent is displayed with the visit name. Note that the user cannot directly enter the subevent number, regardless of the option you choose. The user selects the subevent (or just the visit) from a list of values in the Visit Name field. The information is displayed in the format you select here:
 - **NAME ONLY** The system displays only the **Visit Name**.
 - **NAME+SUB#** The system displays the **Visit Name** and **Subevent Number** in one field with a dot (.) separator; for example, CHECKUP.1.
 - **NAME/SUB #** The system displays the **Visit Name** and **Subevent Number** in separate fields.

 **Note:**

If you do not want to display any visit information, set the next setting, **Hide Visit by Default?** to **Y**.

- **Hide Visit by Default?** Set this value to **Y** to hide the visit identifier by default. The system sets the **Visit Display** setting in the DCI Modules form to

HIDDEN, overriding the previous setting. Exceptions: if there is no defined Clinical Planned Event (CPE, or visit), or the **Use DCI Date** setting is not selected in the DCI Modules form, the Visit must be displayed; the Visit Display setting for the DCM defaults to the **Default Visit Display Code** setting (above) and the user is prevented from setting it to HIDDEN.

Default DCM Header Field Prompts

You can control the default display prompts of DCM Header fields in this category. You control the display of these definitions in the previous category, [Default Settings for Showing DCM Header Fields](#).

The settings in this category are:

- **(Internal) Comment Prompt** Enter the prompt text for an internal comment field.
- **Blank Flag Prompt** Enter the default prompt text to identify a header indicator that a DCM is blank.
- **Data Comment Prompt** Enter the default prompt text to identify a DCM header **Data Comment** field.
- **Date Prompt** Enter the default prompt text to identify a DCM header **Date** field.
- **Generate DCM Header Divider?** Enter Yes to include a line between a DCM Header and the DCM.
- **Lab Prompt** Enter the default prompt text to identify a DCM header **Lab Identifier** field.
- **Length for (Internal) Comment Prompt** Enter a number that indicates the maximum number of characters a **Comment** field can hold.
- **Length for Data Comment** Enter a number that indicates the maximum number of characters a **Data Comment** field can hold.
- **Length of Visit Name** Enter a number that indicates the maximum number of characters a **Visit Name** field can hold.
- **Subevent Prompt** Enter the default prompt text to identify a DCM **Subevent Identifier** field.
- **Time Prompt** Enter the default prompt text to identify a **Time** field.
- **Visit Name Prompt** Enter the default prompt text to identify a **Visit Name** field.
- **Visit Name+Sub# Prompt** Enter the default prompt text to identify a combination **Visit Name** and **Subevent Identifier** field.

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 and migrate data to the new version.

The settings in this category are:

- **Allow Migration of Approved Documents?** Set this value to Yes if under some circumstance the system should migrate an RDCI to a new layout version. Set this value to No to prevent all RDCIs from migrating to new layout versions.
- **Allow Migration of Classic RDCIs?** If you have data you collected in the character-based layout system, and you have created DCI Form layouts for the

corresponding DCIs to DCI Form layouts, set this value to Yes if you want the system to render its RDCIs in the new DCI Forms layouts. Set this value to No if you do not want these RDCIs to be viewed in RDC Onsite.

 **Note:**

For instructions on completing this restriction, see the section "Allowing Access to CRFs Entered via Oracle Clinical" in the *Oracle Clinical Remote Data Capture Onsite Administrator's Guide*.

- **Allow Migration of Locked Documents?** Set this value to Yes if you want to allow the system to migrate locked RDCIs to new layout versions.
- **Default Reason to Retain Approval Verification** Select the default reason to supply if approvals or verifications are retained during DCI Form version migration. If **User Override to Reverse Approvals?** or **User Override to Reverse Verifications?** is set to **N**, the user running the migration cannot specify a different reason (these settings are covered below).

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. If **User Override to Reverse Approvals?** or **User Override to Reverse Verifications?** is set to **N**, the user running the migration cannot specify a different reason (these settings are covered below).

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' approval status to unverified. If set to **N**, DCI Form version migration keeps verified RDCIs' approval status as verified.
- **Last Migrateable Entry Status** Specify highest status at which version migration is allowed. 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 the particular execution of Form Version Migration should reverse the status of all approved RDCIs migrated.
- **User Override to Reverse Verifications?** If set to **Y**, the user running Form Version Migration can specify whether the particular execution of Form Version Migration should reverse the status of all verified RDCIs migrated.

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 blank casebook report (patient-specific and not patient-specific) 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.
- **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** If set to **Y**, the audit history for CRF fields 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. **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?**. The Validate Patient job, which is run from RDC, includes running TMS-related derivation Procedures on the whole study. A user who runs the Validate Patient job is recorded as creating these derived responses for all sites and patients in the study. This may be a problem if the user does not have privileges for all sites in the study.

If set to **Y**, TMS-related Procedures are executed for the whole study during site and patient validation.

If set to **N**, TMS-related Procedures are not executed at all during site and patient validation. However the TMS-related Procedures are included in Batch Validation.

Options in Group Verify dialog

The LOV of each of following settings consists of the four possible combinations of CHECKED vs. UNCHECKED and UPDATEABLE vs ENFORCE:

- CHECKED, ENFORCED: Check box is checked and user cannot change it
- CHECKED, UPDATEABLE: Check box is checked but user can change it
- UNCHECKED, ENFORCED: Check box is unchecked and user cannot change it
- UNCHECKED, UPDATEABLE: Check box is unchecked but user can change it

Exclude batch-loaded CRFs

Batch-loaded CRFs are not included in the Verify action.

Exclude CRFs with discrepancies

CRFs with one or more discrepancies are not included in the Verify action.

Exclude non-migrated CRFs

Non-migrated CRFs are not included in the Verify action.

Include CRFs for this visit only

This setting applies only if the action is invoked from the multi-patient casebook page and it applies to both the Verify and the Undo Verification dialogs.

Options in Group Approve dialog

The LOV of each of following settings consists of the four possible combinations of CHECKED vs. UNCHECKED and UPDATEABLE vs ENFORCE:

- CHECKED, ENFORCED: Check box is checked and user cannot change it
- CHECKED, UPDATEABLE: Check box is checked but user can change it
- UNCHECKED, ENFORCED: Check box is unchecked and user cannot change it
- UNCHECKED, UPDATEABLE: Check box is unchecked but user can change it

Exclude batch-loaded CRFs

Batch-loaded CRFs are not included in the Approve action.

Exclude CRFs with discrepancies

CRFs with one or more discrepancies are not included in the Approve action.

Exclude non-migrated CRFs

Non-migrated CRFs are not included in the Approve action.

Include CRFs for this visit only

This setting applies only if the action is invoked from the multi-patient casebook page and it applies to both the Verify and the Undo Verification dialogs.

Include verified CRFs only

Only verified CRFs are included in the Approve action.

Source Data Verification Plan Defaults

Use these settings to set default values for site-level SDV plans created for this study. These settings can be overridden at the study site level in the SDV Site Planing tab in Remote Data Capture by a user with the necessary privileges.

To set the defaults for a study, access **Design**, and then **DCI Form Local Study Settings**. To make the changes for the database, access **Admin**, and then DCI Form Local Database Settings.

Exclude batch-loaded CRFs?

When checked, indicates that even if a patient is identified as requiring 100% SDV, batch-loaded CRFs for that patient should not be presented in the RDC Surround as requiring SDV. The default is UNCHECKED.

Exclude non-migrated CRFs

When checked, indicates that even if a patient is identified as requiring 100% SDV, CRFs for that patient that were entered via RDC Classic or Oracle Clinical and that have not been migrated to a DCI Form Version should not be presented in the RDC Surround as requiring SDV.

Initial Patients for 100%SDV

The number of initial patients to be automatically selected for 100% SDV. Use a number between 0 and 999.

Patient auto-select rate for 100%SDV

The rate at which newly SDV eligible patients should be tagged for SDV. For example, if the rate is 20%, every fifth patient is selected.

See [Setting Up Partial Source Data Verification](#) for more information.

Configuring DVGs and DCI Modules

The following are configuration and generation procedures you perform at this stage of implementation:

- [Configuring DVGs](#)
- [Configuring DCI Modules' Graphic Layout Settings](#)

Configuring DVGs

DVGs provide a set of named values for a Question. You associate DVGs to Questions in the DCM Questions window (from the **Definition** menu, select **DCMs**, choose **DCMs**, click the **DCM Question Groups** button, and then the **DCM Questions** button). Depending on how you define certain DVG attributes, the values for the DVGs display as single boxes, sets of boxes, or—during RDC Onsite data entry—as single box pull-down lists. Navigate to **Glib**, then **Discrete Value Grps**, and then **Discrete Value Groups**. Select the DVG in the Discrete Value Groups window and attend to the fields in the fifth row of the form. For more information, see [Example 7-9](#).

Configuring DCI Modules' Graphic Layout Settings

The DCI Modules window has DCI Forms settings. To maintain the controls, navigate to **Definition**, then **DCIs**, choose **DCIs**, select the DCI, and click the **DCI Modules** button. On the far side of the sliding canvas, there is a group of settings that fall under the title, DCM Header Generation. See [Adding DCMs to DCIs](#)

Maintaining DCM Graphic Layouts

A DCM Graphic Layout is a definition that describes the location of static labels, masks, and graphics, and prompts and fields that make up a DCM. The definition is editable in the DCM Graphic Editor. DCM Graphic Layouts comprise the data entry section of a DCI Graphic Layout.

For more information, see:

- [Generating Default DCM Graphic Layouts](#)
- [DCM Graphic Layout Statuses](#)

Generating Default DCM Graphic Layouts

This section describes enabling and generating DCM Graphic Layouts from your DCM definitions to use in DCI Forms at the study level. See [Defining DCMs](#) for information about maintaining DCMs.

The Default Layout Generator creates a file that describes the DCM's Questions in the DCM and their locations within the confines of its associated Form Layout Template values. You cannot generate a default layout if there are no Questions in a DCM or from a retired DCM.

The Default Generator lays the DCM's Questions out left to right, top to bottom, adding more blocks as necessary. No text or field may span a block boundary. Each Question Group starts on a new line. The blocks are numbered consecutively in descending order from the top.

You invoke the graphic Default Layout Generator from the **Special** menu, by selecting **Graphic Layout**. These are your choices:

Generate Default From Scratch

Choose this menu option to generate a new default layout. If you have already edited a layout, the generator returns a message so that you know. If you continue, the generator creates a new default version of the layout that does not include your

manual changes. If you want to save a previous DCM layout versions you have to create a new DCM Subset Layout (a new row in the DCM window.)

1. In the DCM window, from the **Special** menu, select **Graphic Layout**, then choose **Generate Default from Scratch**. The **Attributes for DCM Layout Generation** dialog box opens, displaying default values. You can accept the defaults.

The Form Layout Template value sets the dimensions of your DCM's work area. (For the starter portrait templates, a P stands for Portrait, L for Landscape orientation.) The Block Width, Standard/Max Allowed Height and First Block Height are adjustable within the confines of the template. They determine the areas of the Questions in the DCM. The **Checkbox Shape** and **Checkbox Size** fields control the display of DVG boxes.

2. Click the **Generate** button.

You can now perform several operations:

- You can edit your DCM Layout: From the **Special** menu, select **Graphic Layout**, then choose **Edit**. See [Editing DCM Graphic Layouts](#) for DCM-specific editing instructions.
- You can generate a preview of the layout: Select **Action**, then **Print Preview**.
- If you have enabled automatic PDF Annotations and have full Adobe Acrobat installed with the Oracle Clinical Acrobat plug-in, you can generate the annotations: Select **Action**, then **Print Annotated CRF**. The Print Preview window opens where you can click the **Annotated CRF** button to open the Annotation window. See [Annotating CRFs](#) for more information about setting the values in this window.

 **Note:**

Under some circumstances when creating more than one annotated CRF in a session, the browser incorrectly displays the annotations from a previous layout. You can tell if it is a "stale" version because when you select Annotate, the CRF is already populated with annotations. If your DCI Form has the wrong annotations, stop the Acrobat.exe process from the Windows Task Manager's Processes tab and regenerate the layout.

Generate Default from Character Layout

If you laid out this DCM in the character-based layout system, you can migrate it to the graphic system by choosing **Special**, and then the **Graphic Layout** menu option. The system has algorithms to translate a character-based DCM Layout definition to a graphic-based system definition.

Update from DVG Definition

If you have updated one or more DVGs or DVG subsets used by Questions in a DCI DCM, you can update an existing DCI Layout to reflect the changes without generating another default layout. Run the Update from DVG Definition utility from the DCMs window: From the **Special** menu, select **Graphic Layout**, then **Update from DVG Definition**. The utility propagates changes to the following properties:

- Active Values
- Display Type Code
- Checked Flag Value

 **Note:**

The utility does not propagate any changes related to check box layout. To propagate changes to check box properties, you must generate a new default layout.

Edit

From the **Special** menu, select **Graphic Layout**, then choose **Edit** to edit the Graphic Layout for the selected DCM. For complete instructions about editing graphic DCM Layouts, see [Editing DCM Graphic Layouts](#).

Browse Layout

From the **Special** menu, select **Graphic Layout** to view the Graphic Layout for the selected DCM Layout.

DCM Graphic Layout Statuses

Navigate to **Definition**, then **DCMs**, choose **DCMs**, select your study, and then choose the **Graphic Layout** tab. Find the DCM and, if necessary, its Subset and Layout. This window displays related information about the state of a DCM, including:

- **Available?** Before you can generate a DCI Graphic Layout that uses a DCM Layout, you must generate the DCM Layout and select the **Available?** box for the DCM. You cannot select this box if the DCM requires editing, verification, or if there are pending post-edit updates.
- **Edited?** A check mark here indicates that a Graphic Layout has already been generated and that layout work has already been saved. This is a read-only field.
- **Needs Editing/Verification?** A check mark here indicates that either the layout is invalid or there have been post-edit updates to the data definitions. To test a layout's validity, from the Layout Editor, select **File**, then choose **Validate**; see [Validating Layouts](#).
- **Post Edit Updates?** A check mark here indicates that there have been changes to the DCM's data definition that impact the DCM Layout. The system selects this field if you add a Question, change whether a Question is collected or displayed, remove a Question, or change the number of repeats for a Question. If selected, you must review and save the layout from the Layout Editor. This is a read-only field.

When definitional updates are made that impact layouts, Oracle Clinical sets the flags as follows:

- The DCM layout is selected for **Needs Edit /Verification**.
- The DCM is not selected as **Available**.
- Affected DCI layouts are selected as **Needs Regeneration**.

Data entry continues to make use of the DCM/DCI layouts in the existing, active form version. The newly edited layouts do not go into use in data entry until a new form version is generated; see [Maintaining Form Layout Versions](#).

When you edit the layout to synchronize it with the definitions and successfully validate and save the layout, the system sets the flags as follows:

- The **Needs Edit / Verification** settings for the DCM layout is not selected.
- The **Edited** setting for the DCM layout is selected.
- The **Available** setting for the affected DCM layout remains unselected. (You must manually set this value). The setting cannot be selected if the **Needs Edit / Verification** is selected.
- The **Needs Regeneration?** setting for the DCI layout remains selected.

Maintaining DCI Graphic Layouts

A DCI Graphic Layout is a graphic description of a DCI used for input in generating an HTML data entry form and PDF template for the Patient Data Report and Blank Casebook Report. This section describes how to generate and edit a DCI Graphic Layout. The Maintain DCIs window has all of the tools to create initial DCI Graphic Layouts. You can also work from the Provisional window. To open Maintain DCIs, navigate to **Definition, DCIs, and DCIs**. See [Defining DCIs](#) for information about maintaining DCIs.

In this section:

- [Viewing DCI Graphic Layouts' Status](#)
- [Generating Provisional DCI Layouts](#)
- [Generating DCI Forms](#)
- [Maintaining Form Layout Versions](#)
- [Migrating Data to New Oracle Clinical DCI Form Versions](#)

Viewing DCI Graphic Layouts' Status

The Maintain DCIs window and the DCI Modules window display the status values for a DCI's Graphic Layouts. For more information, see:

- [Viewing Status Values on the Graphic Tab](#)

Viewing Status Values on the Graphic Tab

The Maintain DCIs window has three tabs: General, Page Tracking, and DCI Layout. The parameters related to DCI Forms are on the DCI Layout tab. On this tab, you can view a summary of the status of the layouts for the DCI Form version record. You cannot change the values here. These are the values on the DCI Layout tab:

- **Active Layout Version #** The version number of the active layout. If the field is empty, the DCI has no active layout. This field is read-only.
- **# of Retired Layouts** Displays the current number of retired versions of the layout for this DCI
- **PDR Page Definition** Page Definition of the Provisional Layout the generator uses when it creates a Patient Data Report. The field is empty if there is no Provisional Layout.
- **DCI Form Page Definition** Page Definition to be used when DCI Form is generated
- **Prov. Layout Version #** Version number of the Provisional Layout. The field is empty if there is no Provisional Layout.
- **Needs Regeneration?** There is a check mark in this field if the DCI Form Layout must be generated or regenerated. This is set if the DCI Module definition has changed since

the last DCI Form Layout was generated, or if there have been definitional changes to its DCM definitions.

- **Needs Edit?** There is a (Y) in this field if you have to edit the provisional form layout before you can generate a DCI Form. Post-edit updates can also set this flag.
- **Post-Edit Layout Updates?** If there have been any changes to the data definitions that comprise a DCI since the last edit, the system sets this value to Y. Conversely, if there have been no changes, this value is blank.
- **Prov DCI Form Generated?** If a DCI Form corresponding to the Provisional Layout has been generated, the system sets this value to Y.

Generating Provisional DCI Layouts

To generate a default Provisional DCI Graphic Layout, follow these instructions

1. Navigate to **Definition, DCIs**, then **Provisional DCIs**. Choose a study that is enabled for DCI Forms. (See [Enabling DCI Forms](#).)
2. Select a DCI that has DCMs that are available for DCI Forms. (Set the DCMs' **Available?** box on the Graphic Layout tab of the Maintain DCMs window.)
3. From the **Special** menu, select **DCI Layout**, then choose **Generate Provisional Layout**. The **Attributes for DCI Graphic Layout Generation** dialog box opens. Here are the parameters you can modify:
 - **Unit of Measurement:** You cannot change to another measure here. (You set Unit of Measurement for the local database. Navigate to **Admin**, and **DCI Form Local Database Settings**. See the *Oracle Clinical Administrator's Guide*.)
 - You can choose another template. The list only displays templates that can accommodate the largest DCM block size in the DCI.
 - The **PDR Page Definition** is the page size for printouts of the DCI Form's Patient Data Report and viewing and printing the layout. The list of values displays the definition's name, height, width, and binding offset. The list excludes definitions where either height or width are smaller than the associated Form Layout Template's height or width dimensions, or if the binding offset would cause the form layout template to touch or overlap the edge of the Page Definition. You can change the value of the PDR Page Definition after the DCI Form is generated if necessary. See [Setting Binding Offsets](#).
 - The DCI Form Page Definition for Page Definitions for DCI Forms. The list only includes definitions where both height and width are at least two points larger than the Form Layout Template's height and width.
 - The **Checkbox Shape** determines the shape for all boxes generated. You can change this value for all the DVG Questions in the group.
 - **Checkbox Size:** You can choose from a list of values. (You specify the list of values by populating reference codelist DCIF CHECKBOX SIZE.) You can change the size of individual boxes in the layout editor, but not the shape.
4. Click the **Generate** button. The system creates or overwrites the Provisional Layout for the DCI.

Generating DCI Forms

When you have a default DCI Graphic Layout, you can generate the DCI Form that will be used for RDC data entry and for the generation of Patient Data Reports. (If you want to edit the layout first, navigate to **Special, Graphic Layout**, and **Edit Graphic Layout**. See [Editing Layouts](#) for general editing instructions, or [Editing DCI Graphic Layouts](#) for DCI-specific editing instructions.) The DCI must be provisional and its DCI Graphic Layout must be valid before you can create a DCI Form.

DCI Forms are stored in the database in two formats: one in HTML to be used for data entry in RDC Onsite and another in PDF to be used for generating the Patient Data Report and Blank Casebook Report. *DCI Form* refers to both representations.

To generate a DCI Form:

1. From the Maintain Provisional DCIs window, select the DCI. Navigate to **Special, DCI Layout**, and **Generate Provisional DCI Form**.
2. The system prompts you for a Reports Server. If you have difficulty connecting, compare your error message to the topic descriptions in the [Troubleshooting Graphic Layout Problems](#).
3. Click **Generate**. The DCI Form Generator, which resides on the Reports Server, transforms the DCI definition into both an HTML and a PDF-readable definition. The system creates a new version of the DCI Form.

Note:

Each CRF page must fit on a printed page or you will not be able to see all the data in the Patient Data Report. The system converts the layout separately for the PDF reports and HTML data entry. There are likely to be small differences in pixel counts. When you create a layout, test both uses.

You can generate formatted Patient Data Reports of received data from both Oracle Clinical and RDC. From Oracle Clinical, navigate to **Conduct, Conduct Reports, Data Validation**, then **Graphic Patient Data Report**. The Blank Casebook Report can be generated only from RDC Onsite or the command line. Parameters allow you to customize the output. For information about the parameters, see the *Oracle Clinical Remote Data Capture User's Guide*.

Maintaining Form Layout Versions

Oracle Clinical stores your DCI Graphic Layouts and their associated DCI Form records, if any, as DCI Form Layout Versions (FLVs). You can have multiple FLVs for a DCI. You set their status in the Maintain DCI Form Versions window. The window displays the details and history of all DCI Graphic Layouts and associated DCI Form generations for a selected DCI. To access the Maintain Form Versions window, navigate to **Definition, DCIs**, and **DCIs**. Select a DCI and click the **DCI Form Versions** button.

For more information, see:

- [Form Layout Version Rules](#)
- [About the DCI Form Versions Window](#)

- [Changing DCI Form Version Status](#)
- [Changing a PDR's Page Definition](#)

Form Layout Version Rules

You can have multiple versions of DCI Graphic Layouts and if their associated DCI Forms. You create new versions when you want to change the layout of a DCI Form that has already been replicated or used to collect production data. This is useful when a protocol amendment requires you to add or delete Questions. You can also make use of this capability to change to otherwise refine the layout, including changing the header and footer.

The DCI Forms Versions window is inactive if any of these criteria are true:

- The selected DCI Layout has the status Retired
- If you are in Locate DCI mode
- If you are in Query-only mode
- If DCI Form definition is not enabled for the study. (If the **DCI Form Definition Enabled?** box is not selected. See [Configuring Local Studies](#).)

These rules apply when changing the status of a form layout version:

- You can change the status to Active if the DCI is Active and the DCI Form has been generated for the Provisional FLV.
- You can change the status to Provisional only if there is no production data entered against the FLV, none of the other versions have the status Provisional, and the study has been replicated by design or by disconnected replication (not applicable for GLIB DCIs), and the FLV is not retired.
- You can change the status to Retired only if production data has been collected using this FLV or the study definition has been replicated via regular or disconnected replication (excluding for Global Library Level DCIs for the last condition).
- If you change an Active version to Provisional or Retired, a warning dialog box stating, "Warning: You need an active version for deriving the current version in RDC and Form Version Migration. Continue? [Yes] [No]." If you select No, the system changes the status back to the previous status.
- If you change a Retired or Provisional FLV to Active and an Active FLV exists and the above conditions do not prevent you from doing so, a dialog box states, "Changing status to Active will retire existing active form layout version. [OK] [Cancel]." If you choose OK, the system retires the Active version, and activates the current version. Otherwise no action is taken.

About the DCI Form Versions Window

This window maintains DCI Layout and DCI Form versions. For a Provisional DCI Layout, you can change DCI Layout Status, Description, Page Definitions for Printing, and DCI Form Generation. From this window you can perform all of the DCI Form activities that you can perform from the DCI window, including to generate DCI Layouts, edit or browse DCI Layouts, generate DCI Provisional Forms, and delete Provisional and Active layouts. In addition, you can copy from an existing DCI Form version record to create a new DCI Form version, you can change the PDR Page Definition, and you can specify a description for a DCI Form version. There are rules

that govern these activities. The system prevents you from violating the rules. The following section describes these rules.

Changing DCI Form Version Status

A DCI Form version can have one of three statuses: **P** (Provisional); **A** (Active); or **R** (Retired). Across all of a DCI's versions, here are the rules about status assignment:

- Only one DCI Form version can have the Provisional status.
- Only one version can be Active.
- There can be any number of retired versions.
- You can only edit the Provisional version.
- If the DCI is Active, you can change the form version status to Active.

Each time you generate a DCI Form, the system creates a new version and gives it the next available number.

Changing a PDR's Page Definition

The system chooses the PDR Page Definition when you print the PDR Report or blank CRFs. When a PDR is printed, the individual Received DCIs are printed using the Page Definition defined for the DCI Form version associated with that RDCI. The current form version is associated with the RDCI at the time of data collection and can be changed by migrating data. Migrating data is covered in the next section.

Migrating Data to New Oracle Clinical DCI Form Versions

You can control the migration of patients to a specific DCI Form version of a DCI by assigning the patients to DCI Books. The DCI Books specify the preferred version for its DCIs through the DCI Book Constraints window. (For more information on version constraints and setting preferred versions, see [Defining DCI Book Constraints](#) . You control the migration by assigning only the patients you want to migrate to a modified book. The preferred version of the DCI Book also controls the version of the form to be used for data entry in a new CRF. See [Running Form Version Migration](#) for instructions on executing the migration.

In this section:

- [Assigning Patients for Migration](#)
- [Form Migration Examples](#)
- [Maintaining Approvals and Verifications During Form Version Migration](#)
- [Configuring Default Settings](#)
- [Running Form Version Migration](#)

Assigning Patients for Migration

A DCI's preferred version setting in its DCI Book constraints specifies the target version for the migration. By assigning only specific patients to the DCI Book that is being migrated, you can control what patients are migrated to what version. It is possible to set up different DCI Books to point to different versions so that some patients are migrated to one version while other patients assigned to a different DCI Book are migrated to different versions. This is particularly useful for controlling migration on a site-by-site basis.

Migration and assignment of a DCI Book to a patient must occur at the location that owns the patient. However, the generation and creation of DCI Form versions, creating DCI Books, and defining DCI Book constraints occurs at the study owning location. Version migration is a cooperative effort between sponsor and site.

There are two ways to assign a patient to a DCI Book:

- You can automatically assign patients to a DCI Book when data is entered using a DCI Book, depending on how the BOOK_USAGE and the OCL_STATE reference codelists are configured.
- Patients can be initially assigned or changed to a book through the Maintain Patient Positions window.

Depending on the study migration settings, all or some of the received data assigned to the DCI Book migrates, even though the data was entered against other DCI Books.

Form Migration Examples

The following are examples of some of the possibilities available with version migration.

Example 12-1 Adding a Question to a DCI Form and Selectively Rolling Out the Change

You decide to add a Question named PREGNANCY to a DCM named AE and remove Question SMOKING. The AE DCM is used in several DCIs and DCI Books. There is already RDC received data entered via DCI Forms. No patients are assigned to a DCI Book. You want to roll this out for new entry to all locations, but you want to migrate existing data only at the current location. Because data has already been collected, you need to create a new DCI Form version. Here are the steps:

1. From the DCMs window, navigate to the **DCM Questions** window.
2. Mark the SMOKING Question as not collected in subset.
3. Add the PREGNANCY Question to the DCM.
Edit the Graphic Layout to include the new Question to the DCM.
4. Edit the Graphic Layout to include the PREGNANCY Question. Set **Available** to Y on the Graphic Layout tab.
5. For each DCI that uses the DCM where you want to incorporate Question PREGNANCY, if you want to incorporate the new Question:
 - a. If no Provisional Form Version exists, either (1) generate a new Provisional Form layout or (2) copy an existing form version to serve as a basis for the new Provisional Form Version.
 - b. Edit the existing Provisional Form layout or the one created by the previous step. This will automatically incorporate the new Question.
 - c. Verify you want to use the new layout.
 - d. Generate the DCI Form version.
 - e. Make the DCI Form version active. No changes should be needed for the DCI Book Constraints: by default, the preferred version for the DCI Book Constraint is CURRENT.
6. Replicate the definition to each location. The system collects new data using the new DCI Form version.

7. At the current location, assign the patients for that location to a DCI Book that includes the relevant DCI. There must be a DCI Book Constraint record for the DCI. If it is an unplanned DCI, Unplanned Use Allowed must be checked.
8. Migrate data for the specified DCI Book. (See [Running Form Version Migration](#).)
1. Modify or create a Form Layout Template to incorporate the changes.
2. Generate a new Provisional Form Layout Version for the DCI. If you created a new Form Layout Template, specify the new Form Layout Template.
3. Generate the DCI Form version.
4. At each location, replicate the new definition and migrate the data based on that DCI Book. (See [Running Form Version Migration](#).)

Maintaining Approvals and Verifications During Form Version Migration

The Form Version Migration window has controls to retain or reverse approvals or verifications for CRFs that are being migrated to a new form version. Before the patch, form version migration reversed all approvals and verifications. With the new controls on the Form Version Migration window, you can indicate whether a form version migration warrants either re-approving or re-verifying, the affected CRFs.

You can also set default settings at the database and study levels.

Setting Up the Reference Codelists

There are two reference codelists that provide the set of possible Reasons for retaining or reversing approvals and verifications. APPROVE VERIFY RETAIN CODE populates a list of reasons for retaining a book's approvals and verifications, and APPROVE VERIFY REVERSE CODE populates a list of reasons for reversing a book's approvals and verifications.

You can provide a single, generic reason for each codelist, as described in the following section. Or, you can use these codelists to characterize the types of form changes that do and do not require reversal of verifications and approvals. For example, perhaps a new label for a field requires re-approval, but the addition of a field does not.

In the Form Version Migration window, if the user applies the same action (Retain or Reverse), to both Verifications and Approvals, then Oracle Clinical populates the same reason to both Verifications and Approvals. The user can change either reason.

Setting Up Codelist APPROVE VERIFY RETAIN CODE

To modify the list of reasons for retaining a book's approvals and verifications:

1. In Oracle Clinical, select **Admin**, then **Installation Codelists**.
2. Query APPROVE VERIFY RETAIN CODE in the Name field. The upgrade includes one entry:
 - **Short Name:** DFLT_RETAIN_R You cannot change this value.
 - **Long Value:** The default text is: (Sample) Form update does not warrant reversal. You can change this text.
 - **Description:** You can add a description in this field.
3. Add more entries. Both the Short and Long values appear in the list of values displayed in the Form Version Migration window.

Setting Up Codelist APPROVE VERIFY REVERSE CODE

To modify the list of reasons for reversing a book's approvals and verifications:

1. Query APPROVE VERIFY REVERSE CODE in the Name field. The upgrade includes one entry:
 - **Short Name:** DFLT_REVERSE_R You cannot change this value.
 - **Long Value:** The default text is: (Sample) Form update requires reversal. You can change this text.
 - **Description:** You can add a description in this field.
2. Add more entries. Both the Short and Long values appear in the list of values displayed in the Form Version Migration window.

Configuring Default Settings

This section describes configuring approval and verification retention options at the database and study level.

Configuring Default Settings at the Database Level

If you have Superuser or Administrator privileges, you can set the default Retention or Reversal policy for approvals and verifications at the database level:

1. In Oracle Clinical, select Admin, then DCI Form Local Database Settings. Then click on the Version Migration plus symbol. The Version Migration category includes the following new settings:
 - **Default reason to retain approval/verification** Choose an entry from the list of values. The list of values draws from the APPROVE VERIFY RETAIN CODE reference codelist.
 - **Default reason to reverse approval/verification** Choose an entry from the list of values. The list of values draws from the APPROVE VERIFY REVERSE CODE reference codelist.
 - **Default setting for Reverse Approval Status** Select Y or N.
 - **Default setting for Reverse Verification Status** Select Y or N.
 - **Last Migrateable Entry Status Choose the data entry mode from the list of values** Choose a data entry status from the list of values.
 - **User Override to Reverse Approvals?** Select Y or N. If you choose Y, the user executing the form version migration can change the default reversal or retention policy for approvals.
 - **User Override to Reverse Verifications?** Select Y or N. If you choose Y, the user executing the form version migration can change the default reversal or retention policy for verifications.
2. Save.

Configuring Approval and Verification Settings at the Study Level

You can enable or disable approvals and verifications at the study level:

1. In Oracle Clinical, select Design, then DCI Form Local Study Settings. Then click on the Version Migration plus symbol. The Version Migration category includes the following new settings:

- **Default reason to reverse approval/verification** Choose an entry from the list of values. The list of values draws from the APPROVE VERIFY REVERSE CODE reference codelist.
- **Default reason to reverse approval/verification** Choose an entry from the list of values. The list of values draws from the APPROVE VERIFY REVERSE CODE reference codelists.
- **Default setting for Reverse Approval** Select Y or N.
- **Default setting for Reverse Verification Status** Select Y or N.
- **Last Migrateable Entry Status** Choose the data entry mode from the list of values
Choose a data entry status from the list of values.
- **User Override to Reverse Approvals?** Select Y or N.
- **User Override to Reverse Verifications?** Select Y or N.

2. Save.

Running Form Version Migration

In Form Version Migration window you can retain or reverse Approvals and Verifications in the existing DCI book response data.

- Migration occurs for all patients assigned to the selected DCI Book.
- Migration occurs for all patients assigned to the selected DCI Book, and all CRFs entered using a DCI where the (new) form version is later than the version used for the initial data entry.

The following graphic shows the new controls for verifications and approvals in the Form Version Migration window:

The screenshot shows a software window titled "Migrate Form Version by Book (Study: HANDSON)". The window contains the following elements:

- DCI Book Name:** A text input field with a dropdown arrow.
- Audit Comment:** A text input field.
- Test Run:** A checkbox that is checked.
- Approvals Section:** Contains radio buttons for "Retain" and "Reverse" (selected), and a "Reason" dropdown menu with "DFLT_REVERSE_R" selected.
- Verifications Section:** Contains radio buttons for "Retain" and "Reverse" (selected), and a "Reason" dropdown menu with "DFLT_REVERSE_R" selected.
- Footer:** A note stating "*Other than Audit Comment, all the fields are mandatory" and four buttons: "Exit", "Submit Job", "Job Status", and "Change Study".

To run a form version migration:

1. In Oracle Clinical, select **Definition**, then **DCIs**, and finally, **Migrate Form Version by Book**. The Migrate Forms Version by Book window opens.

2. If necessary, you can change the study by clicking the Change Study button.
3. Select the DCI Book from the DCI Book Name list of values.
4. If available, choose to retain or reverse Approvals in the DCI Book, and choose a reason from the list of values. If you cannot override the default settings, you are prevented at the database or study levels.
5. If available, choose to retain or reverse Verifications in the DCI Book, and choose a reason from the list of values. If you cannot override the default settings, you are prevented at the database or study levels.

 **Note:**

If you choose to retain both Approvals and Verifications, upon choosing a reason in either field, Oracle Clinical automatically chooses the same reason in the other field. You can change the reason to another choice, if necessary. The same happens for reversals: If you choose to reverse both Approvals and Verifications, upon choosing a reason in either field, Oracle Clinical automatically chooses the same reason in the other Reason field. You can change the reason to another choice, if necessary. If you choose to retain Approvals but reverse Verifications, or vice versa, then you must choose a reason for both approvals and verifications.

6. Select Test Run to perform a test run of the migration. Oracle Clinical returns the number of RDCIs eligible for migration. It does not commit the form migration to the database. To execute the Form Version Migration, unselect Test Run.
7. Click **Submit Job** to queue the migration job.
8. You can click the Job Status button to check on the job's progress.

Maintaining Form Layout Templates

DCI and DCM Graphic Layouts inherit their shape from its DCI Form Layout Template. It defines the *work area*, which is the area between the header and footer, and the size of the header and footer when previewing the layout or when generating the DCI Form. This area is used for the DCM and response related fields when generating the DCM and DCI Layouts. The contents of this area can be edited in the DCM or DCI Layout Editor.

You maintain DCI Form Layout Templates at the Global Library level. You can generate different definitions of your templates to accommodate different page sizes and orientations. From the **Glib** menu, select **Form Layout Templates**, and choose **Maintain Form Layout Templates**. (For provisional work, navigate to Maintain Prov Form Layout Templates.) The Form Layout Template window has two tabs: The Form Layout Templates tab displays the values for one template; the Multi Form Layout Templates tab displays a scrolling list of all available templates.

 **Note:**

When you edit a Form Layout Template, the changes do not cascade to existing DCM Layouts, DCI Layouts, or DCI Forms.

The status controls the availability of the Form Layout Template. Set to A (Active) if you are ready to use. You cannot edit the layouts of active Form Layout Templates. To edit an active Form Layout Template, change the status to P, edit the layout, then change it back to A. Set the status to R (Retired) if you don't want the layout to be available for use and you don't want to allow editing of the layout.

For more information, see:

- [About the Starter Form Layout Templates](#)
- [Copying a Form Layout Template](#)
- [Validating Form Layout Templates](#)

About the Starter Form Layout Templates

Oracle Clinical 4.6 includes four starter templates:

- **OC_A4_L**: Suitable for positioning on the ISO A4, Landscape (horizontal orientation) Page Definition
- **OC_A4_P**: Suitable for positioning on ISO A4 Portrait (vertical orientation) Page Definition
- **OC_USL_P**: Suitable for positioning on standard US Letter (8 1/2 x 11 in.) Portrait Page Definition
- **OC_USL_L**: Suitable for positioning on US Letter Landscape Page Definition

You cannot modify any of the starter templates. You have to make copies of them. To begin making your own templates, select a starter template that most closely resembles your needs and copy it.

Copying a Form Layout Template

To copy a template follow these instructions:

1. From the **Glib** menu, select **Form Layout Templates**, then click **Maintain Form Layout Templates** or **Maintain Prov Form Layout Templates**.
2. Select an existing template to be the basis of your new template.
3. From the **Special** menu, select **Copy Form Layout Template**. A dialog box titled **Copy Form Layout Templates** opens.
4. Choose a template from the list of values in the **Copy from Template** field. In this section, the template you choose here is called the *parent* template.
5. Name your new template in the **Copy to Template** field.
6. Click the **OK** button to create the copy.

You can then adjust how the default generator arranges and sizes different parts of the DCI in your template. The next section describes these tasks.

Validating Form Layout Templates

Unlike DCM and DCI Graphic Layouts, you cannot save an invalid Form Layout Template. Oracle Clinical separates the saving process from the validation process for layouts so that you can work on them over more than one session. Check the Form Layout Template's validity by selecting **File**, then **Validate**. If you receive any errors, the error message box includes descriptions of them. Correct the errors and try validating the template again.

Editing Layouts

The three types of layouts, DCM, DCI, and the Form Layout Template, have their own editors. However, all three editors use many of the same layout tools. This section describes these tools.

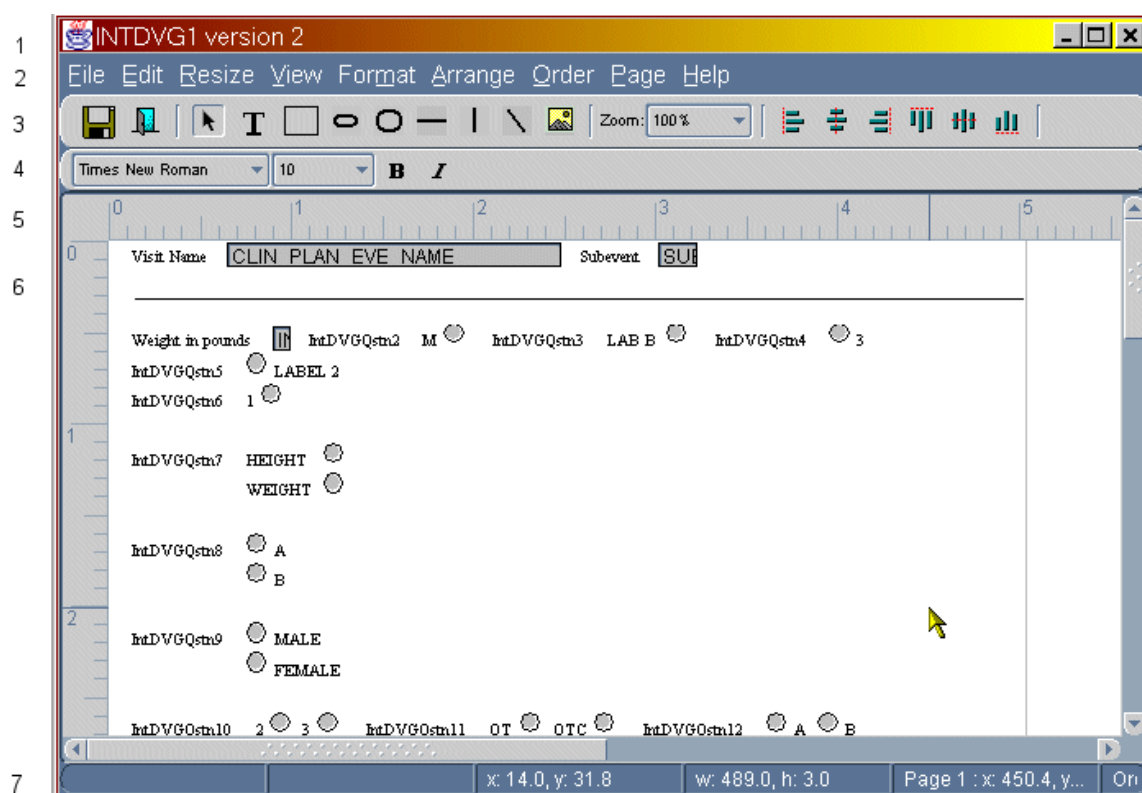
For more information, see:

- [Introducing the Editor](#)
- [Working with Layout Objects](#)
- [Formatting Masks](#)
- [Formatting Label and Prompt Text](#)
- [About Undo, Redo, and Revert](#)
- [Setting Cursor Visitation Order](#)
- [Validating Layouts](#)
- [Editing DCM Graphic Layouts](#)
- [Editing DCI Graphic Layouts](#)
- [Editing Form Layout Templates](#)

Introducing the Editor

This section describes the Graphic Layout Editor interface. [Figure 12-3](#) is the screenshot of a DCI Layout session. The interface is identical for the other Graphic Layout editors. It has a column of numbers along the left border. The following list describes the areas of the layout editor adjacent to the numbers.

Figure 12-3 Graphic Layout Editor Interface



1. The topmost bar is the Windows title bar. In a layout session, the title bar displays the name of the layout.
2. Row two is the main menu.
3. The third row is the main toolbar. It includes these icons:
 - **Save:** Click the floppy disk icon to save your latest work.
 - **Exit:** Click the doorway icon to close your session. The editor prompts you if there is unsaved work.
 - **Selector:** Click the arrow icon to turn your mouse pointer into an object selector.
 - **Create Label:** Click the T icon to create a free form text label.
 - **Rectangle:** Click the rectangle icon to create a rectangle.
 - **Rounded Rectangle:** Click the rounded rectangle icon to create a rectangle with rounded corners.
 - **Horizontal Line:** Click the straight horizontal line icon to create a horizontal line.
 - **Vertical Line:** Click the straight vertical line icon to create a vertical line.
 - **Free Form Line:** Click the diagonal line icon to create line at any angle.
 - **Insert Image:** You can import a graphic image into your layout. The graphic must be a .gif or .jpeg file. You store graphics for this purpose in a location you specify on the Application Server's registry variable RDC_DCIF_IMAGES, which is set during installation. Click the **Insert Image** icon in the toolbar. A dialog box opens. You can

either enter a graphic file name or click the **Browse** button to see a list of graphic files in the images directory and select a file, then click **OK**.

 **Note:**

If the registry variable RDC_DCIF_IMAGES_VALIDATE is set to Y, you must have access to the network location for the directory specified in the RDC_DCIF_IMAGES variable and you must select a file from that directory. In addition, the network location must be inside the firewall.

- **Zoom:** Adjust the size of the layout on your screen by dropping down the **Zoom** button and selecting a magnification value.
 - **Arrange:** There are six icons for arranging or aligning multiply selected objects.
4. Row four is the font formatting toolbar. (See [Formatting Label and Prompt Text.](#))
 5. Row five is the ruler. You can change it to centimeters, inches, or points. The ruler marks the position of the mouse pointer with a pale line. In this screen shot, you can see lines on the ruler at about five horizontal inches by two vertical inches.
 6. Workspace. The appearance of the workspace depends on your template configuration. Note the vertical line five inches from the left side of the workspace. This represents the defined right margin of this DCI Layout.
 7. Status bar: The status bar is divided into cells. The cells display geometric information:
 - The third cell from the left displays the coordinates of the upper-left corner of the selected object.
 - The fourth cell displays the dimensions of a selected object.
 - The fifth cell displays the coordinates of the mouse pointer, including the current block or page number. If the mouse pointer is outside the borders of the template, the block or page number displays as x.

For more information, see:

- [Choosing Unit of Measure](#)
- [Setting Grid Options](#)

Choosing Unit of Measure

You can change a template's unit of measure to inches, centimeters, or points. To change the unit of measure, select the **View** menu option **Unit of Measurement**. There are 72 points in an inch. There are 25.4 points in a centimeter. You set the unit of measurement each time you open the layout, or you can change the unit of measure permanently in the Maintain DCI Form Settings window. (From the **Admin** menu, select **DCI Form Settings** window.)

Setting Grid Options

The editor includes an option to display a superimposed grid of geometrically arranged dots over the workspace. The grid can simplify positioning objects. You display and set

the grid's properties from the **View** menu. To turn the grid display on, select the **Show Grid** box in the **View** menu.

You can set the editor to align an object to the grid when you move it with a mouse drag operation. To enable this feature, select the **Snap to Grid** box in the **View** menu.

You can change the distance between the points on the grid. Select the **View** menu option **Set Grid Size**. A dialog box opens with entries for changing the grid height and grid width. Note that the values do not have to be the same value, so you can create a rectangularly-spaced grid. This might be useful if you are laying out lines of regularly-sized text or rows of Questions.

Working with Layout Objects

This section describes the types of objects in a layout. Layout objects include any item in the layout, including lines, prompts, checkboxes, Questions, boxes, and graphic images.

Note:

A DCI Form, which is the generated version of a complete layout, can contain up to 2000 objects. If you exceed this number, the generation of the DCI Form Version may fail. If your layout approaches 25 pages or about 4 MB, you are approaching this limit.

For more information, see:

- [About Input Objects](#)
- [About Static Objects](#)
- [Inserting Flex Fields](#)
- [Selecting Objects](#)
- [Spell Checking](#)
- [Positioning Objects](#)
- [Grouping Selected Objects](#)
- [Freezing Selected Objects](#)
- [Resizing Objects](#)
- [Viewing and Modifying Object Properties](#)

About Input Objects

A layout inherits the data collection objects, or *input objects*, from its DCMs, DCIs, or layout templates. When you open a default layout, it contains only input objects, arranged according to the data definitions and the Default Generator's layout algorithms. An input object has these three parts:

- **Field:** A field is the response area for collecting data. It is the expression of a login or Question field. It has no visual expression. When you print out a blank CRF, the field is invisible. When you print out a PDR, the data is printed at the location of the field. In RDC Onsite, you enter data in the field.

- **Mask:** The mask is the visualization object that indicates the location and shape of a field on paper or the screen.
- **Prompt:** The prompt is text that identifies the input object.

When you move the mask, the prompt and the field automatically follow. But you can select and modify them individually. The field can be difficult to select independently because it occupies the same territory as its mask. Changing the mask to a line or reducing its size can simplify selecting its field.

About Static Objects

You can add objects that are not associated with any input objects. These can be graphics, labels, or geometric shapes.

Inserting Flex Fields

Flex fields are read-only fields that you can add to the headers and footers of a Form Layout Template. You customize flex fields to display values. The system determines the value of a flex field at run time by a flex field function that you customize. See "Customizing Flex Fields for DCI Forms" in the *Oracle Clinical Administrator's Guide* for instructions on customizing and enabling flex fields.

To insert a flex field, select the **Insert** menu and then choose **Insert Flex Field**. A dialog box opens displaying any available flex fields. (The dialog box limits you to ten available flex fields.) Check one or more flex fields and click **Ok**. The Form Layout Template editor inserts the selected flex fields into the non-display section of the header and footer. You can then drag and drop a flex field into the header or footer areas of the Form Layout Template. The Insert Flex Field operation has the following characteristics:

- To remove a flex field once it is part of the layout, drag and drop it into the non-display section.
- Once you add a flex field to a Form Layout Template, you cannot un-check it.
- You can edit the flex field object's label text in the layout.

Selecting Objects

There are four ways to select objects. Selected objects display *pull handles*, which are small black squares at the corners and midpoints of the object's boundaries.

Selecting Individual Objects

To select one response field, field label, or text object: with the selection tool, click inside the borders of the object. For a box object, either position the point of the selector over the border of the box and left-click, or position the cursor inside the box and click-drag in a diagonal direction.

Selecting Contiguous Objects

To select all the objects in an area of the layout: Click-drag the selector in a diagonal direction across the area.

Selecting Non-contiguous Objects

You can Control-click to select or deselect objects regardless of their geographic location in the layout. If you are Control-clicking a box, note that you must position the mouse pointer of the selector tool directly over its border.

Selecting all Objects

You can select all displayed items: Select the **Edit** menu and then **Select All**. This menu option excludes frozen items, and it does not work in Browse mode.

Spell Checking

If your installation enables spell checking, you can check the spelling of words in text objects in the body of a layout. See the *Oracle Clinical Installation Guide* for instructions to enable this option.



Note:

Spell check works for any English values if they are represented as single-byte text instead of multi-byte text.

You can check the spelling of one selected text object. Select the object, then navigate to **Edit** and choose either the **Spell Check Current** or the **Find Next Misspelling** option. The following sections describe these two options.

About the Spell Check Current Option

Choose the **Spell Check Current** menu option to check the spelling of a selected field. If there is a word that is not in the spell checker's dictionary, (an *unrecognized word*), a dialog box opens allowing you to choose from the following options:

- The **Replace With** field allows you to choose a near match from the **Suggestions** list.
- You can choose to **ignore** the item.
- You can add a new spelling to the dictionary by clicking the **Learn** button.
- You can choose to **Replace All** of the same occurrences of the word in the layout to the value in the **Replace With** field.

The Spell Check Current operation has the following characteristics:

- When the spell checker finds no unrecognized words in the selected text object, the checker returns the message "No Error".
- If you did not select a word when you invoked this menu option, you receive an error message.
- If you have selected more than one object of any type, you receive an error message.
- The option checks all of the words in the object.
- When you act upon an unrecognized word, the checker looks for other words that are not in the dictionary. When there are no other unrecognized words, the checker returns the message, "No Error".
- When there are no more unrecognized words, the dialog box closes automatically. The spell checker does not look for misspellings in the next text object; it only acts on the selected text object.

About the Find Next Misspelling Operation

You can check for unrecognized words continuing from your selected position. Navigate to **Edit**, then choose **Find Next Misspelling**. This operation processes each text object in turn, starting with the nearest text object after the current selected text object. The Find Next Misspelling operation has these characteristics:

- The operation searches for unrecognized words in text objects starting from the currently selected object and proceeding left to right down the layout.
- If you have selected more than one object, the operation starts checking from the upper-most, left-most object in the selection.
- If the operation finds an unrecognized word in an off-screen text object, it scrolls the layout to display the object.
- If the operation finds an unrecognized word in an off-screen text object, it scrolls the layout to display the unrecognized word.
- If there are no unrecognized words in the text objects after the selected object to the end of the layout, the operation starts searching for words in text objects with unrecognized words at the beginning of the layout.
- When there are no unrecognized words in the entire layout, then the checker returns the message "No error."
- If the operation finds a text object with an unrecognized word, it behaves in the same manner described in the [Spell Checking](#), "About the Spell Check Current Option".

Positioning Objects

This section describes the several methods for positioning objects. You can use them in conjunction with the **Snap to Grid** option for positioning objects to specified increments. (See [Setting Grid Options](#).)



Note:

RDC Onsite automatically displays a yellow icon to the right of each Extended Text Question response field. This icon is not displayed in the Graphic Layout Editor. Be sure to leave space for it (approximately 15 points) so the icon does not overlap adjacent fields.

RDC Onsite displays only part of Extended Text Question responses in the CRF. It displays the number of characters defined as the Question length, minus 24. If additional characters have been entered, it also displays an ellipsis (...). (The 24 characters are reserved for the extended text prefix, even though the prefix is not displayed in RDC Onsite.) The full text—up to 10,000 characters—is displayed only in a pop-up window that you do not need to design in the Graphic Layout Editor.

Note that the icon is also not visible in Print Preview from the Graphic Layout Editor.

Moving an Object with the Mouse

You can move an individual object or a group of selected objects with the mouse. For a rectangle object, place the tip of the mouse pointer on the perimeter of the rectangle. (Try to avoid the midpoints and corners, which are the locations of the grab handles.) Click and hold while you drag the rectangle to a new position. For any other type of object, place the tip of the mouse pointer within the object's perimeter, then click and hold while you drag the object to another position. To move multiple selected objects, click and hold on any one of the selected objects, then drag. The editor moves all selected objects.

Moving Objects with the Keyboard Arrows

You can move one or more selected objects with the keyboard arrows. This approach can allow more precise positioning than you can achieve with the mouse.

Positioning Objects from the Properties Dialog Box

You can specify an object's coordinates in the **Properties** dialog box. Select **Edit**, then **Properties**. This approach does not rely on a mouse's graphic resolution limitations, so you can arrange objects to within 0.1 point (or 0.001 centimeter, or 0.001 inch). You can position individual or grouped objects. Note that if you enter any value that violates the general layout parameters, the box closes but makes none of your changes.

The coordinates are the position of the upper-left corner of the object's boundary. The X Position value indicates the left edge of the object's distance from the left side of the block. The Y Position value indicates the object's distance from the top of block (not the top of the template).

Aligning Selected Objects

The Graphic Layout Editor has six tools for systematically aligning groups of selected objects by the objects' axes or borders. You invoke the tools either from the **Arrange** menu or from toolbar icons. Position the mouse pointer over a toolbar icon to read a description of how it aligns selected objects.

Oracle Clinical has two **Arrange** menu options for distributing selected objects:

- **Distribute Horizontally:** This menu option redistributes all selected objects so that the amount of horizontal space between each object is the same. The left-most and right-most objects retain their positions and the spaces between the intervening objects are equal.
- **Distribute Vertically:** This menu option redistributes all selected objects so that the amount of vertical space between each object is the same. The top-most and bottom-most objects retain their positions and the spaces between the intervening objects are equal.

These distribution options work differently according to which of a field's objects you select:

- If you select a field's visualization object, the distribution options treat all of that field's objects (prompt, input object, and visualization object), as a single entity and keeps their relative positions to each other the same.
- If you select a field's visualization object and input object, the distribution options treat all of that field's objects (prompt, input object, and visualization object), as a single entity and keeps their relative positions to each other the same.
- When you only select a field's prompt, the distribution options move the prompt independent of its visualization object.
- When you only select a field's input object, the distribution options move the it independently from its visualization object.

Grouping Selected Objects

You can *group* objects together so that they remain in the same relative position to each other. Grouping reduces the risk of inadvertently moving one arranged object without another, and to simplify moving groups of objects. For example, if you create a graphic and text that you want to treat as a unit, by grouping them, when you move one, the other moves with it. Grouping works for any set of selected objects. To group objects, select them, then select the **Arrange** menu option, **Group**. The **Group** menu item is only active when you have selected more than one ungrouped object.

You can *ungroup* grouped objects if you decide to change their relative positions. To ungroup objects, select them, then select the **Arrange** menu option, **Ungroup**. The **Ungroup** menu option is only active when you select a grouped object.

Freezing Selected Objects

You can *freeze* one or more objects so that they cannot change position or size. Freezing objects prevents them from being inadvertently moved or resized. To freeze objects, select them, then select the **Arrange** menu option, **Freeze**. The **Freeze** menu option is only active when you select a grouped object.

You can *unfreeze* objects if you decide to move them or resize them. To unfreeze objects, select them, then select the **Arrange** menu option, **Unfreeze**. The **Unfreeze** menu option is only active when you select frozen objects.

Resizing Objects

You can change the height and width of rectangles and text objects, and the lengths of text fields. You can also systematically resize a group of selected items in one operation.

Resizing Individual Rectangles

Select the object to expose its pull handles (small black boxes at the midpoints and corners of the object). Place the point of the mouse pointer over a pull handle. Click and hold while you drag the pull handle at least one character length.

Note:

If a field collects the response to a number Question that includes a decimal point, you must enlarge the box in order to avoid truncating data. The system does not automatically allow space to display the decimal point.

For example, if a Question is defined as a number with length 1,2 and a response of 3.56 is entered, if you do not manually enlarge the response field rectangle the value is truncated to 3.5.

Enabling Multiline Text Responses

In a single-line response field, if the text box is too short to display all the characters in a response, the text scrolls left and right so that you can view the entire response. If you set a text box's **Multiline** property to **Yes**, you can cause the response to 'wrap' at

the right-hand boundary of the text box. If the height of the multiline text box is insufficient to view all rows of the response, the text scrolls up and down.

Users entering text responses can use the Enter key to start a new paragraph in a multiline text box.

To make a text box multiline:

1. Select the text box.
2. In the **Edit** menu, select **Properties**. The Edit Properties window opens.
3. In the **Multiline** field, select **Yes**.
4. Click **OK**.

 **Note:**

Unlike single line text fields, multiple line text boxes do not resize in response to a change in font. If you do not leave enough room, the DCI form may cut off part of the entry.

In the Layout Editor, the text dimensions may not render to exactly the same text dimensions that appear in the generated DCI form. Use the Print Preview function to see how they will appear in the generated DCI form. If the form cuts off text, adjust the height of the bounding object in the Layout Editor.

Resizing Widths of Multiple Objects

You can select multiple objects using Shift+click or Ctrl+click and then use the options in the **Resize** menu to change the widths of the selected objects to a uniform size. The options are:

- **Resize to Smallest Width** makes all selected objects the same width as the selected object with the shortest width
- **Resize to Average Width** makes all selected objects the same width as the the average of all of their widths
- **Resize to Largest Width** makes all selected objects the same width as the selected object with the largest width

Specifying Size in the Properties Dialog Box

You can specify an object's size in the **Properties** dialog box. Select **Edit**, then **Properties**. This approach does not rely on a mouse's graphic resolution limitations, so you can size objects to within 0.1 point (0.001 centimeter, or 0.001 inch). Note that this approach does not work for grouped objects.

Viewing and Modifying Object Properties

If you select an object, you can review and modify many of its attributes. Select the object so that you can see its pull handles, then open the **Properties** menu item from the **Edit** menu. An **Edit Properties** dialog box opens. If the selected object is a field, the dialog box indicates parameters it inherits from its definition.

You can also perform several movement and sizing operations by entering new values in the dialog box. This is an alternate approach to positioning objects that does not rely on a

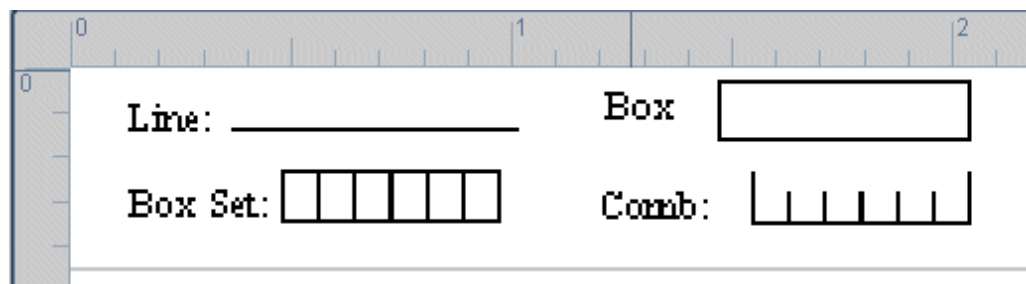
mouse's graphic resolution limitations. However, note that if you enter any value that violates the general layout parameters, the box closes but makes none of your changes.

If the values for a property of a multiple selection are the same for all selected objects, the editor displays the value. Otherwise the **Property** field is blank. If a property is not applicable to any of the objects, the Property field is inactive.

Formatting Masks

The Graphic Layout Editor gives you the option to alter the appearance of a field's visualization object, or *mask*. Masks superimpose graphics that represent field areas on paper or on the screen. [Figure 12-4](#) illustrates the four masks available for fields: line, box, box set, and comb. There are also square and circle masks available for DVG box fields. You can change box masks between squares and circles. (The editor deactivates square and circle selections if the mask you have selected is not for a box field.)

Figure 12-4 Masks



You can also choose for a field to have no visualization object at all.

For more information, see:

- [Changing Mask Formats](#)
- [Resizing Masks](#)
- [Formatting Comb and Box Set Masks](#)

Changing Mask Formats

You can change a mask from one shape to another, and from no mask to another. To change the format of a mask, first select it so that you can see the pull handles. From the **Format** menu, select the **Visual Appearance** menu option, and then select one of its options. Choose the **None** menu option to have no visualization object associated with the field: Only the input object is visible.

Resizing Masks

You resize a box or a line by selecting it and dragging its pull handles. The box set and comb masks have expanded controls described in the next section.

Formatting Comb and Box Set Masks

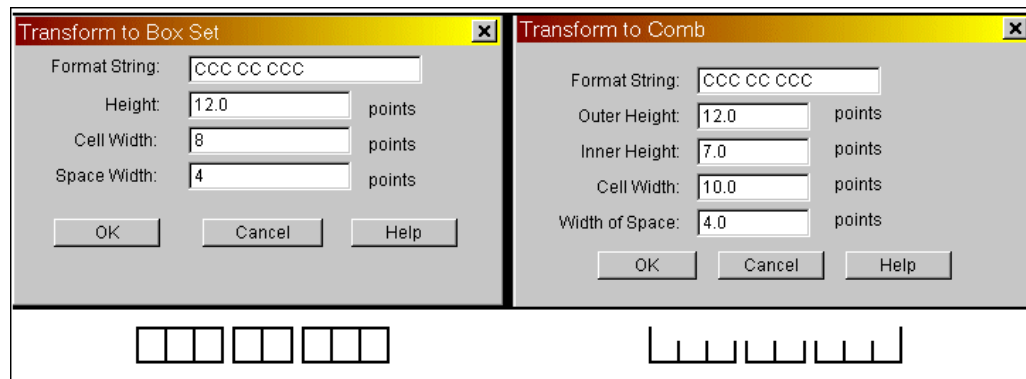
The comb and box set masks create cells for each of the characters in a response. The editor has an expanded set of controls to refine their appearance.

The box set and comb masks work well for paper renderings and for auto-populated fields in Form Layout Templates. However, if an end user enters data into an RDC DCI Form field with a box set or comb mask, the characters do not necessarily line up with the mask's cells. This is because the character spacing in a field is independent of the shape of its mask.

This section describes defining the appearance of a comb or box set mask. The process is the same for both masks.

1. Select a mask by positioning the tip of the mouse pointer on the border of a mask.
2. Navigate to **Format**, then **Visual Appearance**, and select **Box Set** or **Comb**. A **Transform** dialog box opens. You can also change the format of an existing Box Set or Comb by selecting the object and navigating to **Format** and then **Format Mask**.
3. In the **Format String** field, compose a string of characters to represent the cells in the mask. The total number of characters you enter in this field determines the mask's number of cells. Note that the mask can have a long string of C characters in it by default, so be sure you account for all of them. This list describes the behavior of the different characters:
 - The uppercase C character is a control character. The editor creates an empty cell with a border for each C in your string.
 - The Space character is a control character. The editor creates an empty space with no cell borders for each space in your string. Spaces have their own width control (see Step 6 below).
 - For any other character, the editor removes the border from that cell and prints the character instead.
4. Adjust the height of the mask:
 - If you are creating a comb mask, there are two height controls. Adjust the height of the interior vertical lines by entering a value in the **Inner Height** field. The first and last vertical lines are border lines, and you set their height in the **Outer Height** field. You can make them larger to give the mask the appearance of a start and an end. You control the height of the comb borders in between with the **Height** value.
 - If you are creating a box set mask, you set the vertical height of the box set in the **Height** field.
5. Adjust the cell width. Although you can drag the pull handles to change the dimensions of a mask, for combs and box sets this approach can be haphazard, especially if you format several masks to have the same appearance. Enter a value in the **Width** field to set all of the cells in the mask to a specific length.
6. If you include spaces in your format string, you can set their width independently of the other cells' widths. Enter a value in the **Width of Space** field. This allows you to narrow the width of the empty spaces in your mask without narrowing the other cells.
7. Click **OK** to save your settings.

[Figure 12-5](#) illustrates box set and comb settings, with the resulting mask below each of their dialog boxes.

Figure 12-5 Mask Settings and Resulting Masks

Formatting Label and Prompt Text

The font type and size for labels and prompts are set at the database and study levels; see [Configuring Study-Level DCI Form Settings](#). If you have the required privileges you can check or change these settings in the Design menu path. For a default layout, the DCM graphic layout editor displays labels and prompts using the font settings in the DCI Form Settings window.

You can override the font type, size, and emphasis characteristics of the text in individual labels and prompts from the editor's text format toolbar, which is a row of icons between the main toolbar and the ruler.

Figure 12-6 Font Toolbar

Note:

If you regenerate the layout, these manual changes are lost.

You cannot change the default font settings of field text.

For each of these operations, select a label or prompt outline so that you can see its pull handles. Use Control-click to select more than one object to format at the same time. After you make a change, double-click in the middle of a prompt or label to refresh the layout editor view. You can make these text format changes:

- **Change Font Type:** Click the first icon on the Font toolbar. This icon has the font type name of a selected label or prompt (Arial, for instance), as its title. A menu opens displaying available fonts. Select a font, and the menu collapses. Double-click the label or prompt to refresh the screen.
- **Change the Font Size:** Click the icon next to the font-type icon. The icon displays the font size of a selection as its title. A menu opens, displaying available font

sizes. Select a font size, and the menu collapses. Double-click the label to refresh the screen.

- **Toggle Bold Emphasis:** Click the **B** icon in the Font toolbar to change a normal weight selection to bold weight, or a bold weight selection to normal weight. Double-click the label or prompt to refresh the screen.
- **Toggle Italic Emphasis:** Click the **I** icon in the Font toolbar to change a normal slant selection to an italicized object, or an italicized selection to normal slant. Double-click the label or prompt to refresh the screen.



Note:

You cannot add **bold** or *italic* emphasis to text formatted with the Symbol font. If you do, the emphasized text is invisible.

If you move from one label or prompt to another, the font definition you see in the toolbar applies to the previously selected label or prompt until you change the settings for the currently selected field.

About Undo, Redo, and Revert

The editor retains a record of the state of the layout before your most recent editing action, and a record of its state when you last saved it.

To return your layout to its state before your most recent edit, from the **Edit** menu, select **Undo** (where the menu displays the type of last action you made). You can change it back to your last action by selecting **Edit**, then **Redo**.

To return the state of your layout to the last saved version, select **Edit**, then **Revert**. This option enables you to start over without having to exit the layout editor.

Setting Cursor Visitation Order

A data entry form has a *cursor visitation order*: the order in which the response fields become active as a data entry operator completes them. The default Layout Generator assigns visitation order values to each field from left to right, top to bottom, or according to the objects' Disp Seq# definitions. Until you use a layout editor's Order utility, fields retain their default visitation order values, even if you move them in the layout editor. If a default or modified layout does not have the most efficient or intuitive visitation order, you can resequence the values.

To display a layout's cursor visitation order, choose **Order**, then **Display**. The DCM Layout Editor replaces each response field's Question name with its cursor visitation order value. To simplify reordering, the values are multiples of ten. For instance, the DCM's first active field has a value of 10, and the next field in the visitation order has a value of 20. (You cannot adjust the visitation order from this menu path. See the following sections for changing the visitation order.) The system replaces the layout toolbar with the ordering mode toolbar.

There are three visitation ordering tools, each described in the following sections.

**Note:**

You cannot insert the visitation order value of a Question into the visitation order sequence of a repeating Question Group unless it is a member of the repeating Question Group.

For more information, see:

- [Ordering by Row and Column](#)
- [Ordering by Clicking](#)
- [Ordering by Entering Sequence Numbers](#)

Ordering by Row and Column

When the system generates the default layout, it assigns a visitation order to the fields from left to right, from top to bottom. You can assign this same visitation order scheme to a rearranged layout. The row/column visitation order might suffice for data entry, or you can use it as a starting point for further manipulation with one of the other ordering tools. From the **Order** menu, select **Order Current Page by Row/Column**. With this tool you remain in layout mode. (To view the results, from the **Order** menu, select **Display**. To return to the layout after viewing the visitation order, click in the toolbar.)

Ordering by Clicking

Use the Order by Clicking tool to set the visitation order by clicking each successive field in the layout: click once in the field you want to be first, then click once in each successive field in the new order. As you click each field, the system automatically adjusts the order values of the remaining fields. Note that once you have clicked as many times as there are fields, continued clicking assigns the last sequence number with each click.

When you finish ordering, click the door-close button in the toolbar to return to layout mode. Or, if you are unsatisfied with the results of your pass through the fields, click the door-close button. You can then reopen the Order by Clicking tool (or Order by Entering Sequence Numbers, described next), and start again.

Ordering by Entering Sequence Numbers

You can select a field and change its visitation order by entering a value. From the **Order** menu, select **Order by Entering Sequence Number**. You select a field and enter a number, and the system adjusts the values of all other fields accordingly. You can either enter the values manually or with the plus and minus incrementing buttons, described in the next sections.

Reordering Manually

To insert a field's visitation between two consecutively ordered fields, assign the inserted field the greater value. (You can also assign a value between the two values.) The system renumbers the fields to the new sequence. As an example, a rearranged layout might have a visitation order like:

Field	Visitation Order
WEIGHT	10
HEIGHT	30
BIRTHDATE	20

To sequence the visitation order of these fields from top to bottom, change the value of **HEIGHT** from 30 to 20 (or, for instance, to 15), and select another field. The system reorders the fields to:

Field	Visitation Order
WEIGHT	10
HEIGHT	20
BIRTHDATE	30

Reordering with the Plus and Minus Buttons

The Order by Entering Sequence Numbers tool also includes plus and minus buttons in the toolbar that you can use to change individual sequence values. You select a field, then click either button to change its value by one value. In the example in the previous section, to change **HEIGHT** from 30 to 20, you would select 30, then click the minus button. The results are the same as manually entering 20.

Validating Layouts

You can save your edit work at any time even if it violates its layout and definition rules. Oracle Clinical separates the saving process from the validation process for layouts so that you can work on them over more than one session. However, you cannot make a layout active until it is valid. You can test a layout's validity by selecting the **Validate** option from the **File** menu.

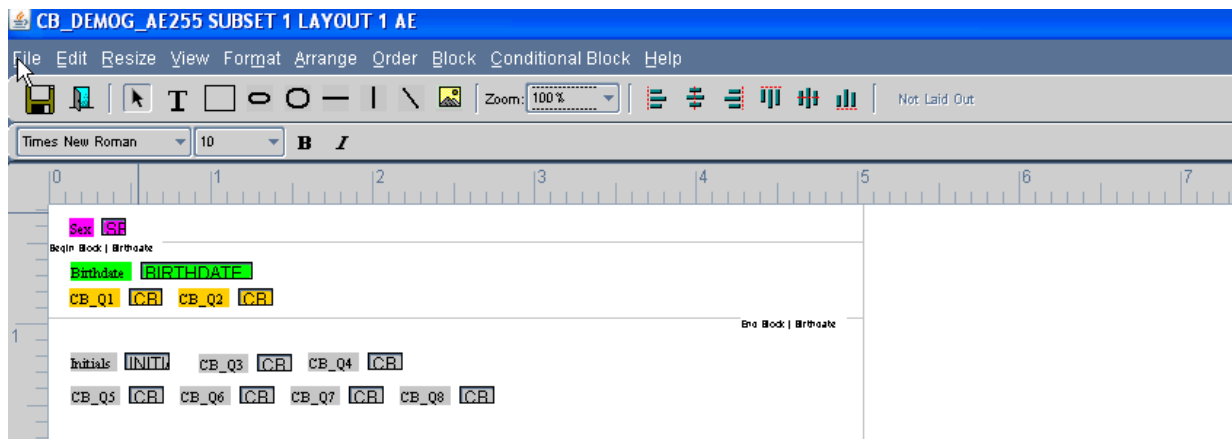
Editing DCM Graphic Layouts

This section describes editor activities that are exclusive to DCM Graphic Layouts. To edit a default DCM Graphic Layout, select the DCM in the Maintain DCMs window and navigate to **Special, Graphic Layout, and Edit**. The editor opens in a separate window. Since you cannot save any work you do in the Maintain DCMs window, a reminder message opens over it.

[Figure 12-7](#) is a screen-shot of a DCM in the layout editor. The DCM has only one block, so the area below the light gray line marks the lower boundary of the valid working area of the layout. Also note that the **Not Laid Out** button is active in the toolbar. This indicates that this DCM has new objects that were not in the DCM before the last generation.

The remainder of this section describes editor tasks that are specific to laying out DCMs. For comprehensive instructions about using the editor tools, see [Editing Layouts](#).

Figure 12-7 DCM Graphic Layout Editor



For more information, see:

- [Editing Blocks](#)
- [Editing Repeating Question Groups](#)
- [Handling Changes to DCM Definitions](#)
- [Previewing DCM Graphic Layouts](#)

Editing Blocks

A generated DCM Graphic Layout consists of one or more blocks. The Default Generator creates as many default-sized blocks as it takes to hold all of the Questions in the DCM. When you add a DCM to a DCI, blocks are the smallest divisible unit that you can add to a page. If you have a very long DCM that spans pages, the breaks between the pages happen at the block level.

You identify a block by its number in the layout. The topmost block is Block 1, and each block below it takes the next higher number. If you add or delete blocks, the editor rennumbers the remaining blocks: if you delete Block 1, the former Block 2 becomes the new Block 1. You can determine which block you are working in by checking on the right side of the taskbar. There is a display of the block number of the pointer's current position, and its block coordinates. If the pointer is below the last block or outside the work area, the block value is x. If you select an object, the taskbar displays its block number, the block's maximum height setting, and the selected object's coordinates.

Inserting Blocks

You can insert an empty block among existing blocks. Inserting a block between other blocks can simplify moving groups of objects, and allow you to leave laid out blocks intact while you move other objects around them. When you insert a block, the size of the block will default to the maximum height.

To insert a block, select **Block**, then **Insert**. A dialog box opens where you enter the number of the block the new block follows, and the height of the block. Inserting a block increments the block numbers of the blocks that follow the new block.

Dividing Blocks

You can divide a block into two blocks. Dividing blocks into smaller blocks can help reduce the length of layouts targeted for DCI Forms. You can then arrange some blocks from successive DCMs on one DCI Form page. To divide a block, scroll down the layout until the top ruler is at the position in the work area where you want to divide a block. Next, select **Block**, then **Insert Block Divider**. The height of the block is the height between two block dividers.

Deleting Blocks

If you have a block that contains no input objects, you can delete it from the layout. Determine the block number to delete by positioning your pointer over it and reading its number on the right side on the taskbar at the bottom of the editor, then select **Block**, and **Delete Block**. The **Delete a Block** dialog box opens. Enter the block number and click **OK**.

Resizing Blocks

You can change the dimensions of your work area within the confines of the DCM's associated Form Layout Template. Select **Block**, then **Set Block Size**. The **Set Block Size** dialog box opens, displaying the current maximum allowed height and width values, in points. Make your changes and click **OK**. Changing the height has delayed consequences, the width changes the width immediately.

Changing the height has two consequences:

- When you save, if any of the blocks are larger than the specified maximum height, you receive an error message.
- When you insert a block, the size of the block defaults to the maximum height. If you choose a number that is larger than the values the DCM inherits from its associated Form Layout Template, you receive an error message.

Editing Repeating Question Groups

This section describes the DCM Graphic Layout Editor's two utilities for editing repeating Question Groups.



Note:

All repeats of a repeating Question Group must be contained in the same block.

Setting Line Height for Repeating Questions

If a DCM has repeating Questions you can distribute all of them to a specific height in one operation. In addition to making their appearance uniform, this tool can also be helpful to compress the area they consume. Select any ungrouped object among the repeating Questions. Then select **Arrange**, and choose **Set Height Per Repeat**. A dialog box opens showing the maximum height. The height you are setting is the line height, which is the distance from the bottom of one line of type to the bottom of the next.

Ordering Repeating Questions

The editor has a utility for handling cursor visitation order in repeating Questions. Cursor visitation is the sequence of selected fields as you tab through an RDC DCI Form. (See [Setting Cursor Visitation Order](#) for a complete explanation.) By default, if you change the cursor visitation order of one repeating Question, all of the repeating Questions in the group automatically adjust to the same order. For instance, if you have 3 repeats of 4 Questions

with this ordering sequence, and you view their cursor visitation order, the Question sequence might look like this:

Sequence 1	Sequence 2	Sequence 3	Sequence 4
50	60	70	80
90	100	110	120
130	140	150	160

If you change the Question with order of 60 to have an order of 70 (and 70 to 60), the editor adjusts the other Questions to have the same relative repeating order:

Sequence 1	Sequence 2	Sequence 3	Sequence 4
50	70	60	80
90	110	100	120
130	150	140	160

Automatic ordering occurs in *locked mode*. In locked mode, you cannot change the sequence for one repeating Question sequence without changing the sequence in all of the others. You can change from the locked mode to unlocked mode, which allows you to order each Question manually. Unlocked ordering enables you to sequence a Question from another group into the sequence of repeating Questions. To change from locked mode to unlocked mode, select the **Order** menu option and deselect the **Lock Ordering of Repeats** box. You can then reorder the visitation order of the repeating Questions. You can relock the cursor visitation order by selecting the **Lock Ordering of Repeats** box.

Handling Changes to DCM Definitions

If new Questions are added to a DCM after you have edited its layout, Oracle Clinical stores the new Question objects in the layout editor's Not Laid Out area so that they do not disrupt your existing work. The Not Laid Out area is only accessible when there are new objects. When you add new Questions, the system sets the **Needs Edit** and **Needs Update** flags. You move them to the DCM Layout by cutting and pasting. Follow these instructions to move new objects into your DCM Layout:

1. Click the **Not Laid Out** button on the DCM Graphic Layout toolbar.
2. Select the new objects. You do not have to select all of the objects at once.
3. Cut the objects: Select **Edit**, then **Cut**, or use the keyboard shortcut Control-X.
4. Return to the DCM Layout by clicking the **Laid Out** button in the toolbar. The Not Laid Out area closes.
5. Paste the cut objects onto the DCM: Select **Edit**, then **Paste**, or use the keyboard shortcut Control-V.

When you move all of them from the Not Laid Out area, the DCM becomes valid.

If you remove Questions from a DCM that has a layout, the system checks the DCM version's **Needs Edit?** value. When you reopen the DCM's layout in the editor, the deleted Questions are gone, leaving the area they occupied empty.

Previewing DCM Graphic Layouts

You can preview either an HTML or PDF representation of your layout without creating a completely functional DCI Form. The preview shows all of the layout's masks and prompts. Previewing a layout the first time requires that you associate the layout with a DCI template: Select **Block**, then choose **Set DCI Layout Template**. Click the list of values (...) button to view a list of available Form Layout Templates. Choose a suitable template and click **OK**.

The Oracle Clinical Report Server has a transform tool for converting DCI Form objects into PDFs or HTML files, and opening the result in your browser. Note that the operation might take a while, and that there is no indication that anything is happening in the meantime.

Editing DCI Graphic Layouts

This section describes editing activities that are exclusive to DCI Graphic Layouts. Select **Special**, then **Graphic Layout**, and **Edit Graphic Layout**. While in the editor you can generate a preview of the layout: Select **Action**, and then **Print Preview**.

For more information, see:

- [Copying DCI Layouts](#)
- [Maintaining Pages](#)
- [Adding Pages](#)
- [Deleting Pages](#)
- [Incorporating Changes to DCM Graphic Layouts](#)
- [Sequencing DCM Header Visitation Order](#)
- [Validating DCI Layouts](#)

Copying DCI Layouts

You can copy an existing DCI Form version record from the DCI Form Version window. The DCI Form version record includes the DCI Layout. Follow these instructions:

1. Navigate to **DCIs**, then choose **DCIs**. Locate the DCI with the layout you want to copy.
2. Click the **Form Versions** button. The Maintain DCI Form Versions window opens.
3. Select the desired DCI Form version to copy.
4. From the **Special** menu, select **DCI Layout**, then choose **Create Provisional Layout as Duplicate**.

Maintaining Pages

In the editor, a DCI Graphic Layout is a column composed of one or more pages. The DCI's DCM blocks are stacked into the column. Blocks do not span page boundaries, so you manipulate the blocks and the pages to make the most economical use of your layout area. Like blocks, the editor numbers pages in a top-down sequence. If you add or delete a page, the editor renumbers the remaining pages in the column. When you position the mouse pointer over a page, the editor displays its number in the right side of the taskbar, which is the bottom panel of the layout editor's window.

Adding Pages

You can add pages to your DCI Layout to make it more suitable for data entry and printing. If you have one page in your layout, you can add a page after page one. In the DCI Graphic Layout Editor, select **Page**, then **Insert**. A dialog box opens. Enter a number and click **OK**. The editor rennumbers the blocks so that they are in sequence.

Deleting Pages

You can delete pages if they contain no input objects. Determine the page number of the candidate page by positioning the mouse pointer over it and reading the **Page** value in the taskbar at the bottom of the editor window.

Incorporating Changes to DCM Graphic Layouts

If you change a DCM Graphic Layout that is used by a DCI Graphic Layout, the system sets the DCI Graphic Layout's **Needs Edit/Needs Update** flag. The next time you modify the DCI Layout, the system replaces the old DCM Graphic Layout with the modified one.

Sequencing DCM Header Visitation Order

If a DCM has header fields such as Qualifying Questions or displayed comments, you can change their cursor visitation order within the DCM header. The cursor visitation order is independent between DCM headers. You control the display of the DCM headers in the DCI Modules window (see [Configuring DCI Modules' Graphic Layout Settings](#) . Choose a visitation order method from the **Order** menu. See [Setting Cursor Visitation Order](#) for more information.

Validating DCI Layouts

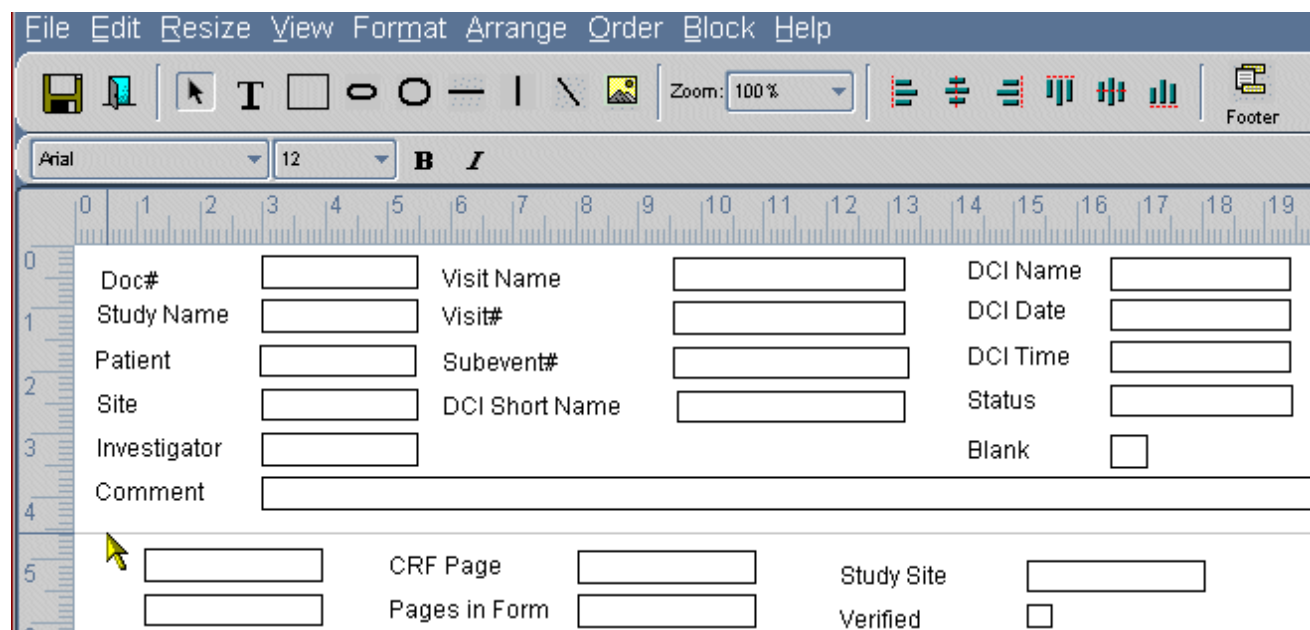
Oracle Clinical separates the saving process from the validation process for DCI Layouts so that you can work on them over more than one session. You can test a DCI's validity at any time by selecting **File**, then **Validate**. If there are no violations the test returns a "No errors" message. If the layout is invalid, the editor returns a message that describes the nature of each violation. A DCI Layout is invalid if any of its objects exceed the layout's boundaries or if there are any objects in the Not Laid Out area. If you try to save an invalid layout, the editor notifies you and asks if you want to save the layout anyway.

Editing Form Layout Templates

This section describes editing activities exclusive to Form Layout Templates. To open a Form Layout Template in the Graphic editor, follow these instructions:

1. Open the Maintain Form Layout Templates window (from the **Glib** menu, select **Form Layout Templates**), or the Maintain Provisional Form Layout Templates window (from **Prov Form Layout Templates**, select **Maintain Form Layout Templates**).
2. Select the template to edit.
3. Open **Special**, then select **Edit Form Layout Template**. An editor session opens as an additional window.

Figure 12-8 The Template Editor's Header Workspace



For more information, see:

- [About the Template Editor's Workspaces](#)
- [Sizing Form Layout Templates](#)
- [Previewing Form Layout Templates](#)
- [Sequencing Visitation Order in Form Layout Templates](#)

About the Template Editor's Workspaces

When you edit a template, you do not view the entire template at one time. You view either the header or the footer. You toggle between the header and footer workspaces by clicking the footer or header icon on the toolbar, or from the **View** menu, choose **Show Footer** or **Show Header**. [Figure 12-8](#) shows the header workspace, so to change to the footer workspace you would click the icon labeled **Footer** (or the **View** menu's **Footer** option). The template editor always opens to the header workspace.

About the Header Workspace

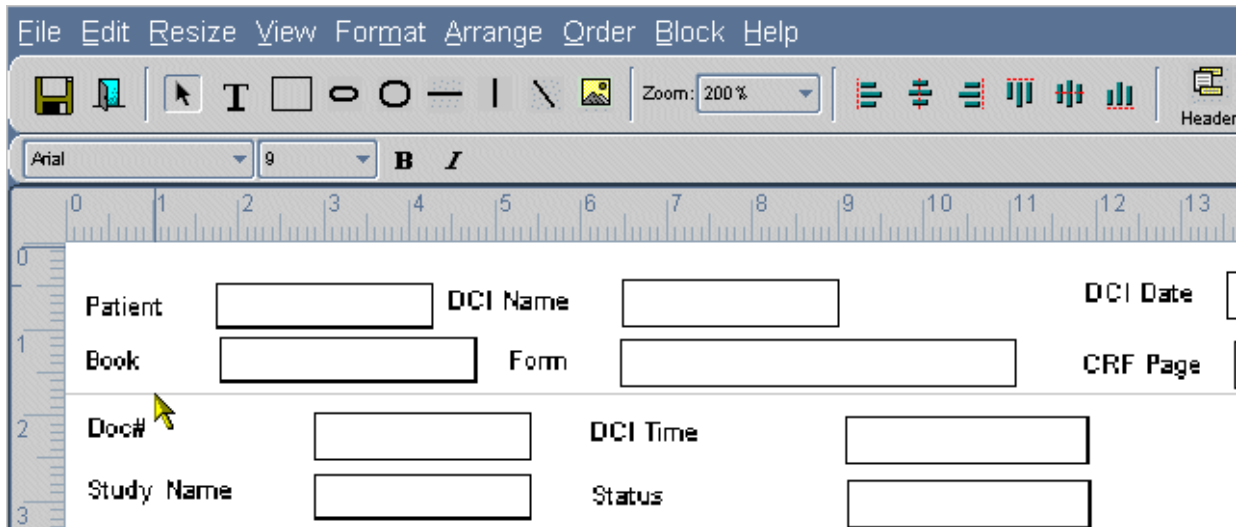
The template editor opens displaying the header workspace. The left side of the work area is the left margin of the template. The top of the work area is the top margin of the template. The work area contains two regions, separated by a horizontal line. In [Figure 12-8](#), the mouse pointer rests on the separator, 4.5 cm from the top of the workspace. (You can change these dimensions from the **Block** menu. See [Sizing Form Layout Templates](#).)

The horizontal line is a separator that marks the lower boundary of the header. The work space above the separator is the header block. All objects in the header block appear in the header of a DCI Form Layout created from this template. In the illustration, a **Comment** prompt and field are the lowest objects in the template's header block.

The work area below the separator is a holding area. This is different than the equivalent region in a DCI Form Layout, which, if viewed in an editor, would contain DCMs. When you

build a DCI Form based on a template, the objects in the holding area disappear. For instance, in the above illustration, the **Study Site** prompt and field are in the holding area, so they would not appear in a DCI Form created from this template. However, you can drag any of the out-of-bounds objects into the header block, or drag any header object into the holding area.

Figure 12-9 Template Editor Detail Showing Footer Workspace



About the Footer Workspace

To switch from the header workspace to the footer workspace, click the icon labeled **Footer** in the toolbar, or select **View**, then choose **Show Footer**. The editor switches to the footer workspace. Like the header workspace, the footer workspace contains two regions, separated by a horizontal line. In [Figure 12-9](#) the separator is 1.7 cm from the top workspace. The horizontal line is a separator that marks the lower margin of the template. The work space above the separator is the footer block. The top of the workspace is where the footer block begins at the bottom of the page. All objects in the footer block appear in the footer of a DCI Form Layout created from this template.

The work area below the separator is a holding area. All objects in the holding area are absent in DCI Forms derived from the template. You can move any of the objects into the footer block, or drag any footer object into the holding area.

Sizing Form Layout Templates

You can change the overall height of the Form Layout Template when you define the height of its headers, footers, and the DCM region. You can also change the width of the layout. Select **Block**, then choose **Set Block Size** to open the **Set Block Size** dialog box. Note that you cannot directly change the value in the **Form Template Height** field. When you click the **OK** button, the editor adjusts the values to equal the sum of the other three heights.

Previewing Form Layout Templates

You can create a PDF of your layout without creating a completely functional DCI Form. The PDF shows all of the layout's masks and prompts. The Oracle Clinical

Reports Server has a Transform tool for converting DCI Form objects into PDFs, and opening the result in your browser's Adobe Acrobat Reader plug-in. Note that the operation might take a while, and that the screen is inactive in the meantime.

Sequencing Visitation Order in Form Layout Templates

Form Layout Templates can have enterable fields in their headers and footers. You can control the order in which a user tabs through a DCI Form's output. From the menu, select **Order**. Then select one of the ordering methods. For more information about ordering, see [Setting Cursor Visitation Order](#).

Creating and Maintaining Conditional Blocks

You must create conditional blocks in the graphic layout editor for any indicator or conditional branching definitions that you want to be in effect for RDC Onsite data entry using your graphic layouts. For a comparison of indicator and conditional branching functionality, see [Designing Response-Dependent Branching](#). You must define indicator or conditional branching before you can create conditional blocks in the layout.

For both types of branching, the default graphic layout creates conditional blocks that include only the target Question.

- In **indicator branching**, you must add the remaining Questions in the Question Group to the conditional block. Do not add any other Questions to the conditional block or remove any of the Question Group Questions. During data entry the cursor always moves to the next Question in the Question Group when the remaining Questions become active.
- In **conditional branching**, you must define a conditional block for the target Question that includes the complete set of Questions that should be answered or skipped, depending on the response to the source Question. The conditional block can include the remaining Questions in the target Question's Question Group as well as Questions in different Question Groups in the same DCM.

You can also change the order of the Questions in the conditional block, including moving the target Question to a position other than first in the block. During data entry the cursor always goes first to the Question that appears first (topmost and leftmost) in the block when the conditional block Questions become enterable.

These are the required steps:

1. Define indicator branching in a Question Group or conditional branching in a DCM; see [Defining Indicator Branching](#) and [Defining Conditional Branching](#).
2. Choose whether to hide or gray out conditional blocks in HTML data entry using the Represent Disabled Blocks setting in DCI Form Local Study Settings; see [Configuring Study-Level DCI Form Settings](#).
3. Generate or update the DCM graphic layout; see [Generating Default DCM Graphic Layouts](#).
4. Create the conditional blocks; see [Creating Conditional Blocks](#).
5. Edit the conditional block if necessary.

For more information, see:

- [Creating Conditional Blocks](#)
- [Invalid Conditional Block Conditions](#)

- [Maintaining Conditional Blocks](#)
- [About Conditional Blocks in DCI Form Versions](#)

Creating Conditional Blocks

You must create all target conditional blocks for a given source Question at the same time. For indicator branching there is only one conditional block, but conditional branching may have many, one for each target Question.

To create conditional blocks for an indicator or conditional source Question:

1. In the Maintain Study DCMs or Global Library DCMs window, select the DCM containing the Question identified as conditional or indicator and navigate to the Special menu, Graphic Layout, and then Edit.
2. In the Layout Editor, open the Conditional Block menu and select **Create Conditional Blocks**. The system displays a list of all source Questions that do not yet have defined graphic target blocks, their Question Group name and a check box indicating whether or not the Question is an indicator Question. If the indicator Question check box is not selected, the Question is defined as a source Question for a conditional branch.
3. Select a source Question from the list and click **OK**.
 - For indicator Questions, the system creates a single conditional block consisting of the remaining Questions in the Question Group.
 - For conditional branch source Questions, the system creates a conditional block for each target Question specified for the selected source Question. The block contains only the target Question, if possible, given the horizontal placement of fields in the layout. Additional Questions are included if they are on the same horizontal plane as the target Question. This is an invalid condition that you must correct. You must also make sure that the Begin and End Block lines do not overlap prompt or response objects; see [Invalid Conditional Block Conditions](#).

If the target Question is part of a repeating Question Group, the begin block is before the first repeat, and the end block is after the last repeat.

For each conditional block, the system creates color-coded highlighting and separators and removes the highlighting and separators for any previously selected source Question; see [Conditional Block Display](#).
4. Edit the layout, if required. In addition to normal editing available for the rest of the DCM layout, you can add and remove Questions from conditional blocks used in conditional branching (but not in indicator branching):
 - Drag Begin or End Block lines to include or exclude Questions in the conditional block.
 - Drag prompt or response objects above or below the Begin Block and End Block lines to include or exclude them in the conditional block.
5. Save your layout. Oracle Clinical executes a set of validation checks, and then reports all failed conditions to you for correction; see [Invalid Conditional Block Conditions](#).

You can either save the layout in an invalid state and correct the invalid conditions later, or cancel and continue with your edits.

For more information, see:

- [Conditional Block Display](#)
- [Nested Conditional Blocks](#)

Conditional Block Display

The DCM Graphic Layout Editor represents conditional block objects as follows:

- The Begin Block and End Block lines mark the beginning and end of the conditional block and include the name of the source or indicator Question. You may need the Zoom feature to read the text.
- Color highlighting:
 - The source Question appears in magenta.
 - The target Questions appear in green.
 - The remaining Questions appear in orange.
 - If the Show All source Questions option (available from the Conditional Block menu) is in effect, the DCM Layout Editor distinguishes the currently selected source Question from the other source Questions with different colored highlighting.

Note:

You can remove the color highlighting by selecting **Clear Highlighting** from the **Conditional Block** menu and restore it by selecting **Show Conditional Blocks** from the **Conditional Block** menu.

- If a block spans pages, dashed lines indicate each page break.

Options from the Conditional Block menu include:

- **Show Conditional Blocks** displays, with color highlighting, conditional blocks for one or more source Questions that you select.
- **Suppress Branch** This feature allows "fall-through" Question sequencing like Oracle Clinical. Creating conditional blocks automatically includes an end-of-block demarcation before the next target Question. At the end of each block, branching goes to the next visible, enabled field.

If you select the Suppress Branch option for a conditional block, the editor redraws the end-of-block line as a dashed line, allowing the Questioning to continue to the next immediate Question rather than branching to the immediately following conditional block rather than branching around it..

If you select the Suppress Branch option after selecting an object that has no end-of-block line, or selects no object at all, the Editor prompts you to select an end-of-block line before continuing.

Nested Conditional Blocks

It is possible to *nest* conditional blocks that are the targets of one source Question inside a conditional block of another source Question if the inside, or *child* blocks can never become enterable unless the outside, or *parent* block is enterable. The child blocks must be the targets of a different source Question than the parent block. All conditional blocks for one

source Question must be contained in a single conditional block for the other source Question.

The system displays only Questions that are contained directly in the selected conditional block: the parent block displays only the Questions that are not included in the child blocks. Each child block displays the Questions included in the child block.

Invalid Conditional Block Conditions

It is possible to create conditional blocks that result in invalid layout generations. When you save your work, the system validates the conditional blocks. If there are invalid conditions, the system generates the invalid blocks anyway, issuing an error message, and saves your work. You can then re-position fields to fix the problems.

The following conditions are invalid:

- The source Question is on the same horizontal plane as Questions in the target blocks. The system cannot generate a line to physically separate the source Question from the set's Questions.
- The Begin and End Block lines overlap other objects (prompts or fields).
- An indicator Question target block includes Questions from another Question Group or does not include all Questions in the Question Group.
- A repeating Question Group is intersected by a Begin Block line or an End Block line.
- Two Begin Block lines or two End Block lines appear in succession, representing overlapping blocks for the same source Question.
- Two Begin Block lines from different source Questions or two End Block lines from different source Questions appear in succession AND the blocks for one source Question are not wholly contained within a single parent conditional block for the other source Question.
- A Begin Block line for a conditional block appears before (above) the End Block line for the same conditional block.
- A target Question lies outside of its originally defined target block.
- A source Question lies within one of its own target blocks.

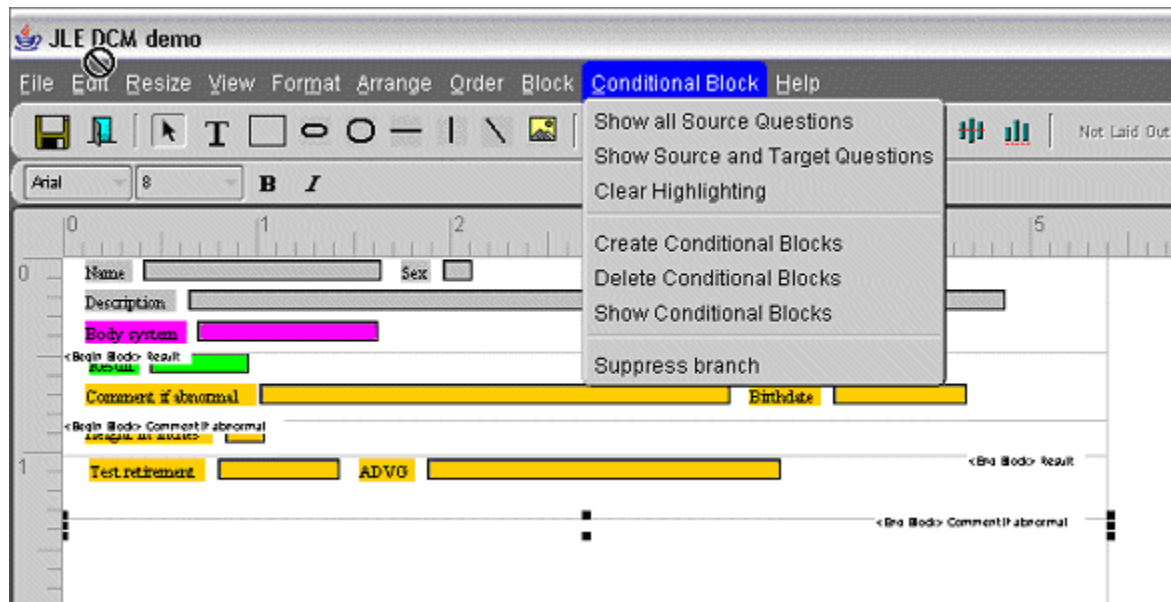
These actions have no impact on the content of the conditional blocks:

- Repositioning block objects within the block.
- Changing tab order.

Maintaining Conditional Blocks

This section describes the tasks available from the Conditional Block menu items. [Figure 12-10](#) illustrates the Conditional Branch menu in a form with a defined source Question (magenta color), a target Question (green color), and a conditional block of Questions (orange). In RDC Onsite, if the user selects the target response, the CRF displays or activates the target and block Questions.

Figure 12-10 The Conditional Block Menu in the Graphic DCM Layout Editor



The following lists describe the DCM Graphic Layout Editor's Conditional Block menu items:

- **Show All source Questions** highlights all source Questions in the color magenta. The DCM Layout Editor highlights all source Questions in magenta, whether their conditional blocks are defined or not.
- **Show Source and target Questions** displays the list of all source Questions in the DCM, whether their conditional blocks are defined or not. Select as many Questions as you want. The system then displays all selected source Questions in magenta and their target Questions in green.
- **Clear Highlighting** turns off all highlighting; see [Conditional Block Display](#).
- **Create Conditional Blocks** makes the target Question(s) for one indicator or conditional branching source Question conditional. See [Creating Conditional Blocks](#) for more information.
- **Delete Conditional Blocks** makes the target Question(s) for one indicator or conditional branching source Question be no longer conditional. This option undoes the work of the Create Conditional Blocks option.
- **Show Conditional Blocks** displays a list of source Questions for which conditional blocks have already been created. Select as many Questions as you want. The system then displays all selected source Questions in magenta and their target Questions in green.
- **Suppress Branch** Ordinarily, when a branching condition is satisfied, a single block is enabled for data entry. When data entry for that block is complete, the branch skips over the remaining blocks. However, by using the 'Suppress Branch' option at the end of a block, you can enable the target block and one or more of the following blocks. For example, if you have three blocks and the first two blocks end with a suppress branch condition, you have three possible ways to enable the conditional blocks:
 - If the target Question is in Block 1, Blocks 1, 2, and 3 are enabled
 - If the target Question is in Block 2, Blocks 2 and 3 are enabled

- If the target Question is in Block 3, only Block 3 is enabled

To suppress the branch at the end of a block, select the End Block line and click Suppress Branch. The system redraws the End Block line as a dashed line.

About Conditional Blocks in DCI Form Versions

The DCI Layout Editor does not display conditional blocks. To see which conditional blocks are included in a DCI Form Version, navigate to Definition, then DCIs. In the Maintain Study DCIs window, click DCI Form Versions and then Show Conditional Blocks.

Troubleshooting Graphic Layout Problems

Note:

Check My Oracle Support for troubleshooting tips.

Here are problems you may encounter when working with Graphic Layouts, and possible solutions to each one:

- [FRM 41213 Followed by a 771000 Error Message](#)
- [FRM-41214 Followed by a 777100 Error Message](#)
- [Nothing Happens When You Try to Generate a Provisional DCI Form](#)

FRM 41213 Followed by a 771000 Error Message

If generating a Provisional DCI Form fails with the error "FRM-41213: Unable to connect to the report server", it indicates that the Maintain DCIs Form cannot communicate with the Report Server to process a PDF generation request.

This can be caused if a previous job failed in such a way to cause the current connection from your session to the Reports Server to be disconnected. Try logging out and then logging back in. If this does not work, have an Administrator check if the Reports Server is running.

If you have Administrator privileges, log on to the Reports Server Computer. From the **Start** menu, select **Control Panel**, choose **Services**, check that the Reports Server has a status of STARTED. If not, start it. If you cannot start it, make sure Oracle Discover 4i Service is stopped.

FRM-41214 Followed by a 777100 Error Message

If you navigate to the **Special** menu, select **DCI Layout**, then choose **Generate Provisional DCI Form**, you may get the message:

FRM-41214: Unable to run the report

and followed by error message 777100: Either the PDF for the DCI Form failed to be generated, or the URL of its location is incorrect.

In that case, check these details:

- Is the DCI Form Layout valid? To check, open the Provisional DCI Form Layout in the editor, and run **File**, and then **Validate**. Fix any errors, and then try again to generate the DCI Form.
- Was the Reports job successful? The job may have run on the Reports Server instance, but failed because Reports thought that its input was invalid. If the job failed, ask the Queue Manager for details. Typical failure messages contain comments like these:
 - Object 'Rectangle_3751' is not fully enclosed by its enclosing object 'Body'. This occurs when the Form Layout Generator places an object too close to a boundary. Go back to the DCM Graphic Layout, and move objects in from the boundaries of the DCM Layout Region.
 - Body area is beyond print area. This occurs when the page selected for generating the DCI Form is not enough to hold the DCI Layouts. Choose a larger Page Definition.
 - BLOCK#_1_ is empty. This occurs when there are empty blocks in DCM and DCI Layouts. Remove the empty blocks through layout editor and retry DCI Form generation. Usually you will get this if the DCM Graphic Layouts are generated with small first block heights.
 - Image is not found. This occurs when Reports Server cannot find the image that was added to the DCI Layout. Check that the `\Software\Oracle\9iDS_HOME\RDC_DCIF_IMAGES` registry value is set on Web Server and Reports Server. This is the UNC path specification of the directory where you store DCI Layout images.
- Does the Form Layout have any empty pages? To check, open the Provisional DCI Form Layout in the editor. If there are any empty pages, the system returns this error.
- Is the output more than 20 pages? Due to technical limitations, the DCI Form may not generate. Try reducing the number of pages to less than 20.
- Try running a reference codelist report to a file. Navigate to **Admin**, then **Reports**, and **Reference Codelists**. Select a reference codelist to print, click **Job Details**, and change destination to FILE. If this does not work, see the Reports Server troubleshooting guide on Metalink.

If you have Administrator privileges, you can try these possibilities:

- Open Windows Explorer to the XMLTEMP directory. Generate the Provisional DCI Form and watch for the creation of the file `PDFGENDCIIDVERSIONS_SN First 10 characters of OC ACCOUNT timestamp MI:SStemp.xml`. If xml exists, save the file to a different folder before exiting the Maintain DCIs window. If it does not exist, check registry setting `OPA_XMLTEMP_UNC`, described below.
- Are the required registry values set? The report that generates the PDF reads these registry values on the Reports Server:
 - `\Software\Oracle\9iDS_HOME\OPA_XMLTEMP_HTTP`
This is the URL of the directory into which XML is downloaded, and PDF is written to be uploaded.
 - `\Software\Oracle\9iDS_HOME\OPA_XMLTEMP_UNC`
This is the UNC pathspec of the directory into which XML is downloaded, and PDF is written to be uploaded.
 - `\Software\Oracle\9iDS_HOME\RDC_DCIF_IMAGES`

This is the UNC pathspec of the directory in which image GIF files are to be found, for incorporation into the generated PDF.

- Is the directory pointed to by OPA_XMLTEMP_UNC shared, with read and write permissions, to the account used to set up the Report Server? Log on to the Reports Server computer as *domain\opareps*. Check that the path specified by the value of registry setting OPA_XMLTEMP_UNC exists. Copy the registry value to the clipboard, then look for the value using Window Explorer. If not, correct the value.
- Is the directory pointed to by RDC_DCIF_IMAGES shared, with read permissions to the account used to set up the Report Server? Log on to the Reports Server computer as *domain\account_used_to_set_up_the_Report_Server*. Check that the path specified by the value of registry setting RDC_DCIF_IMAGES exists. Copy the value to the clipboard, then look for the value using Window Explorer. If not, correct the value.
- Was the generated PDF document placed into the XMLTEMP directory? You must perform this test before exiting the DCI Form. Look in the directory of the registry setting OPA_XMLTEMP_UNC value. PDFGENDCIIDVERSIONS_SN *First 10 characters of OC ACCOUNT timestamp* MI:SStemp.xml.
- Is the Reports Server started as the account used to set up the Reports Server initially? Log on to the Reports Server, from **Control Panel**, then **Services**, and check the property of the Oracle Reports Server to see if it is started as *domain\account_used_to_set_up_the_Report_Server*.

Nothing Happens When You Try to Generate a Provisional DCI Form

If you try to generate a Provisional DCI Form through the **Special** menu in the Maintain DCIs window, and the system returns control to you without returning either an error message or a DCI Form, check the init.ora file for the database.

If the init.ora file contains one or more `utl_file_dir` parameter definitions, but does not contain a `utl_file_dir` parameter definition for the directory `<opahome>/oc/xmltemp` (which is the directory that Oracle Clinical uses during DCI Form generation), you need to add that parameter to the init.ora file.

During Oracle Clinical installation, the directory `<opahome>/oc/xmltemp` is created. For example, if your opa home directory is `usr/opapps`, then the directory `usr/opapps/oc/xmltemp` is created. Add a `utl_file_dir` parameter definition for this directory as follows:

1. Add the following two lines to the init.ora file, immediately before or after the other `utl_file_dir` parameter definitions:

```
UTL_FILE_DIR = /usr/opapps/oc/xmltemp
UTL_FILE_DIR = /usr/opapps/oc/xmltemp/
```

The second line is the same as the first except for the slash at the end. This line is necessary because of a bug: Oracle Clinical currently includes an extra slash in its references to files in the XMLTEMP directory.

2. Stop and restart the database.

13

Copy Groups

Use copy groups to immediately copy multiple database objects either between the Global Library and a study, or between studies. The advantage is that you make one copy operation for several objects instead of one copy operation for each object, and that you do not violate any data model rules in the process. The Copy Groups feature also has a Provisional branch with all of the same functionality.

You can create copy groups at either the study level or Global Library level. The definitional activities are the same at either level. However, most customers create and test copy groups at the study level, then promote them to the Global Library. This chapter describes study level activities. For promoting and maintaining copy groups at the study level, see [Maintaining Global Library Copy Groups](#) . You need to choose the variety of copy group to use, depending on your plans for it.

Copy groups can be **ad hoc**, **provisional**, or **permanent**.

Ad Hoc Copy Groups - If you do not intend to reuse a copy group, and you have no need to leave a record of the group's constituents, perform an ad hoc copy operation. You collect the definitions in the Copy Group form and move them somewhere else from the **Special** menu. If you do not save your work before exiting, the copy group does not remain in the database. You can perform an ad hoc copy operation from either the study or Global Library level.

Creating ad hoc copy groups is the same as creating global or study level copy groups. The difference is that after you invoke the copy operation from the **Special** menu, you exit the Copy Group form without saving your work.

Provisional Copy Groups - The Provisional Copy Groups windows in the Definition and Global Library subsystems provide a testing area for developing copy groups. When you are satisfied with them, you can promote them. Since the maintenance operations for provisional copy groups are the same as their counterparts, follow the same maintenance procedures as for their equivalents.

Permanent Copy Groups - If you expect to re-use a copy group at the study level, you can save it before exiting. The next time you perform a copy operation, you can choose it instead of re-creating it. You maintain study-level copy groups in the Definition subsystem. If your company enforces standardized copy group operations across studies, you can create a copy group in, or promote a copy group to the Global Library. You maintain these copy group records in the Global Library subsystem. (See [Maintaining Global Library Copy Groups](#) .)

For more information, see:

- [Creating Copy Groups](#)
- [Grouping Objects](#)
- [Adding DCMs](#)
- [Adding DCIs](#)
- [Adding Procedures](#)
- [Adding View Definitions](#)

Creating Copy Groups

The following instructions give an overview of creating a copy group. Each instruction is described in more detail later.

1. From the **Definition** menu, select **Copy Groups**, then choose **Copy Groups**. The Maintain Copy Groups form opens.
2. Enter values in the fields: **Copy Group Name**, **Domain**, and **Description**.
3. Click the **Copy Group Details** button.
4. Click the **DCMs** button. (Since all other copy group elements must relate to DCMs, this is the only active button at this stage.)
5. In the DCM list that displays, click the DCM to include. To select multiple DCMs simultaneously, hold the Control key and click each DCM or Shift-click a region.
6. To create a permanent library or study object, make the copy group active before exiting the form.
7. Save the DCM choices. Other object type buttons become active. Click each of the other appropriate buttons to include desired objects. Repeat the previous step as many times as necessary to include the other objects. When you have added all the objects, either:
 - Click **Exit** to store the copy group.
 - From the **Special** menu, select **Copy to Study** to immediately use it in a Study.
 - From the **Special** menu, select **Copy to Library** to immediately use it in the Global Library.

Grouping Objects

You add objects to the copy group in the Details form. From the **Definition** menu, select **Copy Groups**, then choose **Copy Groups**, and click the **Copy Groups Details** button.

The following rules control the objects you can add:

- Objects allowed in a copy group are DCMs, DCIs, procedures and view definitions.
- For DCMs, you specify whether to copy just the base subset or all subsets.
- All objects must be related to the included DCMs. Conversely, DCMs that are referenced by another object in the group cannot be removed from the group.
- Objects can only come from the study you are working in.
- Provisional and active objects can be added to a provisional copy group, but all objects must be active before a copy group can be made active.
- Objects are included in copy groups by reference. Changes to the underlying objects are reflected in the results of subsequent copy operations.
- Retired copy groups are not available for use and cannot be modified.

Adding DCMs

You can enter the DCMs to include in a copy group by selecting **Copy Groups** from the **Definition** menu, then choosing **Copy Groups**, clicking the **Copy Groups Details** button, and the **DCMs** button.

This following list describes the validation rules for adding DCMs to a copy group:

- If copying all subsets is specified, all subsets in the source DCM must be active or the copy procedure will abort.
- If a DCM with the same name exists in the target domain, Oracle Clinical allows the copy to proceed only if it is functionally equivalent to the DCM being copied. Functional equivalency is defined as:
 - Same Qualifying Question, if any
 - Same named DCM Question Groups with the same repeat value
 - Same name, short name, and subset names
 - Same Question (or occurrence) numbers in DCM Questions for each group with the same settings for `Collected_in_study_flag`, `Derived_flag`, and `DVG_ID`.

Depending on the setting of the **Base Subset Only?** field, either just Subset 1, Layout 1, or all subsets and layouts of a DCM are copied. Additionally, all child objects of the DCM are copied.

Adding DCIs

You add DCIs to a copy group by selecting **Copy Groups** from the **Definition** menu, then choosing **Copy Groups**, clicking the **Copy Groups Details** button, and the **DCIs** button.

When you add a DCI, all DCI modules are copied. The name and short name must not be present in the target domain. Explicit references to CPEs are treated as they are currently by DCI copying.

Adding Procedures

You add procedures to a copy group by selecting **Copy Groups** from the **Definition** menu, then choosing **Copy Groups**, clicking the **Copy Groups Details** button, and the **Procedures** buttons.

When you add a procedure to a copy group, the system copies the active version. The procedure name must not be present in the target domain. The system takes the mapping of the procedure to the DCM in the copy group to ensure that the copied procedure references the same equivalent DCM in the target.

Adding View Definitions

You can add view definitions to a copy group by selecting **Copy Groups** from the **Definition** menu, then choosing **Copy Groups**, clicking the **Copy Groups Details** button, and the **View Definitions** button.

When a view definition is used, the view definition and all of its components are copied. The name must not be present in the target domain. If the view definition references a library view

template, only the view definition is copied: the template remains a reference to the library template. If the source of the view definition is a study, then the view template is copied as well. The mapping of the view definition to the DCM in the copy group is used to ensure that the copied view definition references the equivalent DCM and DCM Questions in the target.

Data Extract

Oracle Clinical's Data Extract subsystem is a set of facilities to present patient and clinical study data that meets the demands of external applications related to statistical analysis and reporting. Oracle Clinical presents the data for external access through standard and custom database views created in study access accounts.

Oracle Clinical stores data in tables that are normalized and generic, in that the underlying table structure of the data is the same from one study to another.

This chapter describes the design side of these data extract features and processes. For runtime data extract tasks see the *Oracle Clinical Conducting a Study* manual. See the *Oracle Clinical Administrator's Guide* for information on regenerating and updating views.

Views created before Oracle Clinical Release 3.1 ("pre-3.1 views") continue to be supported, although you cannot have pre-3.1 views in the same study as views created in Release 3.1 and later. As of Release 3.1, Oracle Clinical creates views in the View Builder.

The View Builder is a toolkit that provides the means to perform the following tasks:

- Customize and interactively view default, DCM-based views.
- Create Questions derived at the time of data extract.
- Customize how views are created by, for example:
 - controlling the view mode
 - controlling the view granted for view access
 - controlling the default attributes of the view's Questions.
- Create Cross-study views by program, project, or user-defined study set.

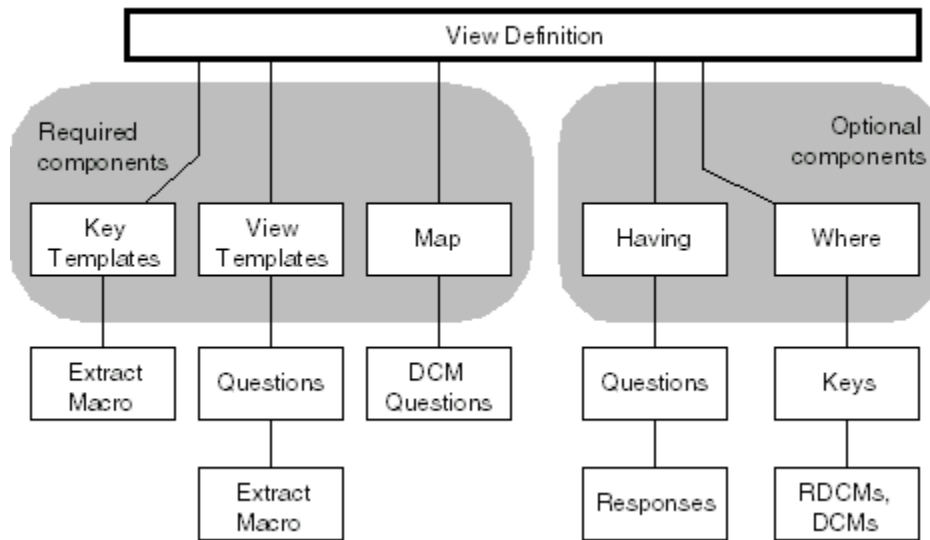
View data can be transferred into database tables for better system performance during user analysis, reporting, and browsing operations.

For more information, see:

- [The View Builder's Structure](#)
- [A Sample Workflow](#)
- [Customizing Data Extract](#)

The View Builder's Structure

The diagram illustrates the relationship of the structural elements in the View Builder. These elements are described in this section.



When you invoke Maintain Study View Definitions (from the **Definition** menu, select **Data Extract View Builder**) or Maintain Library View Definitions (from the **Glib** menu, select **Data Extract View Builder**), the components of a view definition are displayed.

The following components make up a view definition, which itself is the basic building block for data extract view creation in Oracle Clinical:

Component	Function
View Template	Defines the columns for the view based on simple and complex Questions.
Key Template	Defines the set of keys and key expressions for the view based on default and user-defined extract macros.
Mapping	Maps DCM Questions to View Template Questions.
Having Clause	Restricts data retrieved by response values to Questions (optional).
Where Clause	Restricts data retrieved by key values (optional).

A view definition always exists in the context of a mapping. In the context of a study set or a single study, you can query view definitions in an ad hoc fashion; they may or may not become an Oracle view in a study (set) access account.

Actions bearing on a view definition occur in the Global Library, and in the subsystems Definition and Conduct. For tasks in Conduct and data extract context in Conduct, see the manual *Oracle Clinical Conducting a Study*.

For more information, see:

- [Global Library](#)
- [Definition](#)

Global Library

In the Global Library, you define View Templates, Key Templates, and view definitions. You can also extend column definition, by means of extract macros, to include

multivariate expression definition. For the purposes of data extract only, you can also extend Questions or create complex Questions.

For more information, see:

- [About View Templates](#)
- [About Key Templates](#)
- [About Extract Macros](#)
- [About Complex and Simple Questions](#)
- [About Extended Question Attributes](#)
- [Example of Associating Extract Macros to Questions](#)

About View Templates

View templates define the columns for the view, based on response values and expressions. They are defined, created, and stored by the Global Librarian.

View templates are mapped to the source DCM Question. For example, a View Template might specify a view consisting of three numeric columns whose sources are inputs $N1$ and $N2$. The columns correspond to $N1$, $N2$, and $(N1+N2)/2$. This format can then be used in multiple actual view definitions by mapping $N1$ and $N2$ to source DCM Questions in a particular study.

For how to customize View Templates by means of complex Questions, see [About Complex and Simple Questions](#).

View templates can exist at the Global Library level as standard structures defining the Question columns for standard views. These View Templates can be copied as part of a local study view definition, or they can be incorporated by reference as part of a definition repeated over many studies.

View templates can be created in several ways:

- You can create default View Templates as part of default view definitions for a DCM. These templates, if unmodified, produce views just like the standard views of earlier versions of Oracle Clinical.
- You can produce View Templates as if you were using the Oracle Clinical V3.0 Query Builder—that is, by selecting particular DCM Questions to include in a view definition and its default format.
- You can create View Templates explicitly by specifying the column structure and characteristics. For standard views you can also determine column ordering, which applies to all views created from the same template and is consistent for all access accounts.

Note:

Avoid using "CON" as a View Template name. If you are using Windows 2000 or NT, the SAS view creation file con.sas will fail as "CON" is a reserved device name.

About Key Templates

Key templates define the keys and key expressions for the view, based on default and user-defined key values obtained from default or user-defined extract macros.

Key columns are the fixed part of the view definition structure and enable the non-aggregation of data as they set the key variables for a view. Key templates also enable you to specify the SAS names and labels of the key columns, and to order the view data. [Customizing Data Extract](#) summarizes how aggregation of data works, and the keys needed for non-aggregation of data.

Before Oracle Clinical 3.1.1, a Qualifying Question defined at DCM level was extracted as an additional column/variable. As of 3.1.1, a Key Template had the columns Qualifying Value and Qualifying Question. The Qualifying Question key includes the Qualifying Question ID from the DCM; the Qualifying Value key, regardless of whether the underlying DCM is qualified, includes the RDCM Qualifying Value column in a view with the generic column name of Qualifying Value and the attributes associated with the Qualifying Value key extract macro.

The data extract key DYNA_QUAL_VALUE allows dynamic Qualifying Questions to be part of views created with the View Builder. Behavior is as follows:

- In regular data extract views that reference just one qualified DCM, the new key includes a column containing the RDCM qualifying value. The name of the column, as well as Oracle and SAS attributes, comes from the Qualifying Question. When the DCM is not qualified or the view references more than one DCM, the column is omitted. This scenario is the same as for pre-V3.1.1 views.
- In Union views, both In-study and Cross-study, there are three possible outcomes.
 1. If at least one of the component views is qualified, a column is included according to regular view rules.
 2. If all the qualified DCMs are based on the same Qualifying Question, a column and attributes are created, according to regular view rules.
 3. If two or more component views are based on different Qualifying Questions, the view is omitted, with an error message.

About Extract Macros

With extract macros, you customize the way your data is formatted in Oracle and SAS views, because extract macros provide the keys for Key Templates. Default extract macros exist for standard Key Templates—the default templates for view definitions.

You specify an extract macro—SQL or SAS—when defining Questions or Question Group Questions so that the macro is incorporated in SQL and SAS view creation text, respectively.

Before you can make an extract macro active, which is required before it can reference Questions, you must assign a name for it, make sure the Question and the extract macro share the same Question data type, and validate the SQL code.

You can modify views by defining derived columns with any parsable SQL expression that references 0 or more Questions. Embedded function calls may also be used.

About Complex and Simple Questions

Complex Questions are not DCM Questions; they are created for the purpose of data extract. They are not part of default definition, so they have to be defined in the Global Library. Once a complex Question is defined, it references other Questions as well as the extract macro that combines them into a single value.

When you write an extract macro with more than one variable, you accommodate that extract macro by creating a complex Question, which can be broken down into simple Questions, each of which can then be associated with one of the variables.

Simple Questions are Questions that are collected and that are called "simple" to distinguish them from complex Questions.

So for a complex Question with extract macro parameters defined for it as *N1* and *N2*, you can map these two parameters to two simple Questions, *Q1* and *Q2*, respectively. You then map these simple Library (or study) Questions to DCM Questions.

In practice, this process of definition and mapping is largely automated in default cases, but you can control the structure if you want to customize the views.

Maintaining Question Details

Complex Questions combine information collected by multiple simple Questions. The **Details** button in the Questions window accesses the form used to associate the variables of the complex Question with the simple Question that collects the data. COMPLEX Questions are associated with extract macros and QUESTION_SET Questions are associated with Question Sets.

To access the Question Details window, from the **Glib** menu, select **Questions**, choose **Prov Questions**, and click the **Details** button. The **Details** button activates only for complex Questions, therefore only for Questions of type COMPLEX or QUESTION_SET. The extract macros and Question Sets must be defined before you try to associate them with the Question. Use this window to associate extract macros and Question Sets with provisional Questions.

About Extended Question Attributes

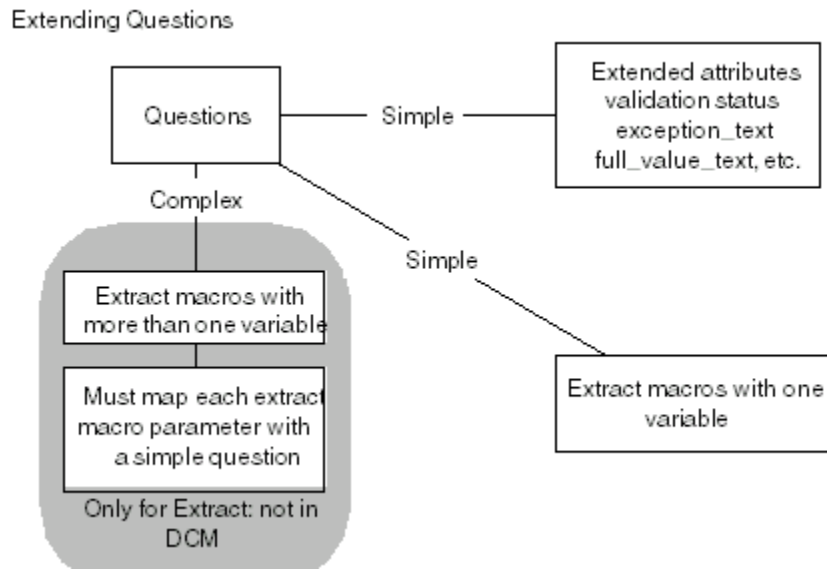
You can extend simple Questions for the purpose of data extract, so that you can choose among attributes when selecting a Question for use in data extract and map the selected attributes to extract macros; these extract macros can then be used to customize views.

Extended attributes, as attributes of the RESPONSES table, allow you to extend data extracted from the value text and validation status as well as from several additional attributes. When you create a simple Question—that is, not a complex Question—in the Global Library, default extended attributes are created. See the *Oracle Clinical Administrator's Guide* for a description of these extended attributes.

The following diagram shows the relationship between Question attributes, complex Questions, and extract macros.

Maintaining Extended Attributes

From the **Glib** menu, select **Questions**, then choose **Prov Questions**, and click the **Extended Attributes** button. Oracle Clinical provides additional attributes that can be associated with Questions to increase the information provided in extract views. Use this window to associate the desired extended attributes with a specific provisional Question.



Discrete Value Group Translation for Applicable Variables

You can determine on a view- by-view, Question-by-Question basis whether to expand the view definition to include discrete value group sequence numbers and/or long values, in addition to the data values for DVG Questions.

Specifying Explicit Access to Other Response Information

You can specify whether to base a column on various attributes of a response. These include handling of exception values, access to data comments, and access to validation status.

In the Data Extract Installation Configuration window (from the **Admin** menu, select **DX Installation Configuration**), you can decide whether to include Question attributes when defining view definitions from DCMs.

When you create new Questions, attributes with LONG_VALUE set to Y in the reference codelist DX_EXTENDED_ATTRIBUTES are created automatically. Two extended attributes set by default are VALIDATION_STATUS and DVG_SHORT_VALUE.

To change the extended attributes for existing Questions, you need to run a SQL script. For instructions, see the *Oracle Clinical Administrator's Guide*.

Example of Associating Extract Macros to Questions

The process of associating extract macros to Questions breaks down into four steps:

Step 1

Create the following two expressions, where *q1*, *q2*, *q3*, and *q4* are Questions, while *fDate* is a database procedure:

Expression Description	Expression Alias
Simple Numeric Expression	$0.3333 * (q1 + q2 + q3)$
Database Procedure	<code>fDate(q4)</code>

Step 2

For the simple numeric expression, create an extract macro named BPFUNCTION, where the corresponding SQL macro would be:

```
0.333 * (<VALUE_TEXT>N1 + <VALUE_TEXT>N2 + <VALUE_TEXT>N3)
```

For the database procedure, create an extract macro named PROCESSED_DATE, where the corresponding SQL macro would be:

```
fdate (<VALUE_TEXT>D1)
```

Step 3

In the Questions form of the Global Library, create a Question named BPCOMPLEX, with a Question type of COMPLEX, a datatype of NUMBER, and an associated extract macro named BPFUNCTION.

To continue working with the BPCOMPLEX Question and the BPFUNCTION extract macro, click the **Details** button, which shows you a window where the SQL macro for this extract macro is displayed, and where you enter the name of a simple Question against each variable listed (N1, N2, and N3 in this case). The simple Questions are available from a list of values. This procedure has broken down the complex Question into three simple Questions.

For the database procedure, create a Question, COLLECTION_DATE, in the Global Library Questions form, with a Question type of anything but COMPLEX, a data type DATE, and an associated extract macro, PROCESSED_DATE.

Step 4

Reference the complex Question BPCOMPLEX in the View Template form. Collect the simple Question COLLECTION_DATE in the DCMs form and accept the default view definition; or you can explicitly reference the Question in a custom view by mapping in the View Definition form.

Definition

As part of Definition, you can create a study set, which includes several studies for either a project, a program, or a selection of individual studies. Creating a study set allows you to create Union views across studies.

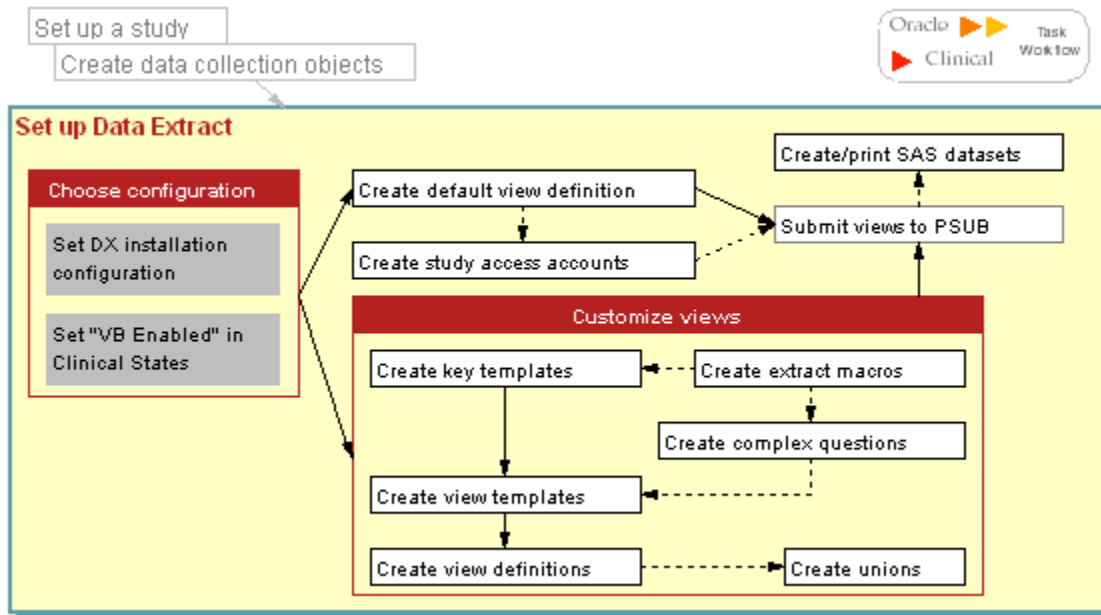
You can define view definitions and View Templates in Definition, and in Glib. View definitions can reference both study- and Global Library-level View Templates.

Although indexing may also be thought of as a Definition task, the menu entry is under Conduct, so the description of data extract indexing is in *Oracle Clinical Conducting a Study*.

A Sample Workflow

Figure 14-1 shows a sample workflow for data extract activities.

Figure 14-1 Simple Workflow for Data Extract



Before starting data extract you need to identify what data to access—a decision depending on the design of the clinical study, on possible problems encountered during study conduct, or on the expected analysis of the data extracted. These factors may also determine whether you accept system defaults and omit the optional steps in the data extract process.

For information on the DX Installation Configuration settings, see the *Oracle Clinical Administrator's Guide*.

Reviewing the Default View Definition

If the View Builder is enabled for your study, when you make a DCM active in Definition and save it, you can choose to say Yes when the Alert box asks if you want to create a default view definition. Accepting a default view definition results in a further Alert box message that a view definition has been created.

If you want to create a default view definition for an existing active DCM, select Default View Def from the Special menu. You can also update a view definition from the Special menu. View definitions created this way can then be submitted to PSUB so that you can obtain your views.

Customizing Data Extract

This section describes the various ways you can customize your data extract process. Many customizations begin with copying, so a section on that feature starts off this section.

For more information, see:

- [Copying](#)
- [Create Extract Macros](#)
- [Creating Key Templates](#)

- [Create Questions and View Templates](#)
- [Maintaining View Definitions](#)
- [Defining Study Sets](#)
- [Creating Union Views](#)
- [Creating Where Clauses](#)
- [Adding DISTINCT terms](#)
- [About Joining Fast Extract Views](#)
- [General Example](#)

Copying

As in other subsystems in Oracle Clinical, copying can play a large part in data extract tasks. The special points in copying for data extract are that:

- You can copy active objects that are in the user domain searchlist. These are displayed by the List function. For Key Templates, the **Special** menu does not allow copying if you have your cursor on a provisional object.
- You can add the domain of the object you want to copy to your user domain searchlist, if you need to.

Create Extract Macros

You can create SQL and SAS macros in the Global Library, and then associate them with complex Questions.

For more information, see:

- [Simplified Extract Macro Workflow](#)
- [Attributes of a SQL Macro](#)
- [Example](#)
- [Macro Data Type](#)
- [Some SQL*Plus Functions](#)
- [Some Examples of SQL and SAS Macros](#)
- [Some Hints for SAS Users](#)
- [Defining Extract Macros Using the Maintain Extract Macros Window](#)
- [Changing a Macro Definition](#)
- [Listing Available Functions in the Database](#)

Simplified Extract Macro Workflow

A typical workflow might follow this scenario:

1. A programmer writes an extract macro, which may be multi-variable.
2. A Global Librarian creates a complex Question and references each of its component Questions to a variable of the extract macro created in step 1.

3. A study data coordinator adds the complex Question referenced in step 2 to the View Template of a view definition, and then maps each Question in the complex Question to a DCM Question.

From the **Glib** menu, select **Data Extract View Builder** to access the Maintain Extract Macros form. In this form, you can:

- Activate or delete a provisional extract macro.
- Retire an active extract macro.
- Create and validate the SQL of an extract macro.

Before creating a macro, you should know about the macro's structure. The following sections describe this structure.

Attributes of a SQL Macro

A SQL extract macro is valid SQL text containing an embedded reference to variables. The variables might be patient data collected in the response table, such as Value Text, Exception Text, Data Comment Text, Data Change Reason Type Code, Audit Comment Text, Discrepancy Indicator, or Validation Status.

Keys may also be referenced in the SQL macro text.

For all response value text variables, you must also specify the Oracle Clinical data type—CHAR, NUMBER, DATE, or TIME.

The SQL macro may reference database functions. If you provide the SQL definition, it is validated for correct syntax.

Example

The kind of problem that would lend itself to solution by a SQL macro might be:

Problem and Goal

The response table contains the following four pairs of values and exception text for the Questions LPARM and LVALUE, where LPARM is a character parameter and LVALUE its numeric component.

Item	LPARM	LVALUE Value	Exception Text
1	HCT	89	-
2	HCT	92	-
3	-	90	-
4	HCT	-	+++

The object is to retrieve the lab data in one column that contains the value text when the value is valid, and the exception text when the value is invalid.

The resulting data should look like this:

Item	Value
1	HCT:89
2	HCT:92

Item	Value
3	90
4	HCT:+++

Solution

To achieve this display, define a SQL extract macro with the following variables:

c1 for the first character variable

n2 for the second numeric variable

The SQL macro HTPARM would be:

```
<VALUE_TEXT>C1||'|'||NVL(TO_CHAR(<VALUE_TEXT>N2),<EXCEPTION_VALUE_TEXT>N2)
```

Another alternative is:

```
<VALUE_TEXT>C1||<FULL_VALUE_TEXT>N2
```

To create a complex Question with this extract macro in the Global Library:

1. Call the Question HTPARM and define it as complex with a data type of CHARACTER.
2. Select the extract macro HTPARM and reference the Questions LPARM and LVALUE.
3. Map the Question LPARM to the variable C1 and the Question LVALUE to the variable N2.

To define a view with the complex Question HTPARM:

1. Include the Question, HTPARM, as an output column in a View Template.
2. Map the Question LPARM to a DCM_QUESTION for LPARM, and the Question LVALUE to a DCM_QUESTION for LVALUE.

Macro Data Type

The SQL macro itself has an Oracle data type, which is derived at compilation. (The **Compilation** field has a value of N when the SQL has not been validated, and a value of Y when the SQL has been validated.) When a SAS macro is defined as well as a SQL macro, the reference variable is always the same type in both cases (i.e., DATE, CHAR, DATETIME, or NUMBER).

Some SQL*Plus Functions

The following table lists some SQL*Plus functions you can use with extract macros.

Table 14-1 Extract Macro SQL*Plus Functions

Macro Name	Question Data Type	SQL Macro	Description
LOWER	CHAR	LOWER (0)	Lowercase.
INITCAP	CHAR	INITCAP (0)	First letter in each word in uppercase; all else in lowercase.
LPAD_1_BLANK	CHAR	LPAD (0, LENGTH(0)+1)	Left-pad with one blank.

Table 14-1 (Cont.) Extract Macro SQL*Plus Functions

Macro Name	Question Data Type	SQL Macro	Description
LENGTH	CHAR	SUBSTR(TO_CHAR(LENGTH(\0)), 1, LENGTH(\0))	Length of input value.
PREFIX	CHAR	'x' \0	Prefix the character 'x'.
SUFFIX	CHAR	\0 'x'	Suffix the character 'x'.
ABS	NUMBER	ABS(TO_NUMBER(\0))	Absolute value.
CEIL	NUMBER	CEIL(TO_NUMBER(\0))	Smallest integer greater than or equal to input value.
FLOOR	NUMBER	FLOOR(TO_NUMBER(\0))	Largest integer equal to or less than input value.
EXP	NUMBER	EXP(TO_NUMBER(\0))	E (=2.71828183) raised to the power of the input value.
LN	NUMBER	LN(TO_NUMBER(\0))	Natural logarithm.
LOG10	NUMBER	LOG(10, TO_NUMBER(\0))	Logarithm, base 10.
SIGN	NUMBER	SIGN(TO_NUMBER(\0))	Sign: -1 if input value is less than 0; 0 if input value is 0; 1 if input value is greater than 0.
SQRT	NUMBER	SQRT(TO_NUMBER(\0))	Square root.
TRUNC_0	NUMBER	TRUNC(TO_NUMBER(\0))	Truncate to 0 decimal places.

Some Examples of SQL and SAS Macros

The following table lists and compares SQL macros, SAS macros, and SAS formats.

Macro Name	Question Data Type	SQL Macro	SAS Macro	SAS Format	Description
COMP_DT9	DATE	TO_DATE(Decode(LENGTH(\0), 6, \0 '01', 4, \0 '0701', 8, \0, NULL), 'YYYYMMDD')	-	DATE9.	Complete a partial date and display, using DD-MON-YY format. When no day is specified, the first day of the month is assumed. When no month is specified, July 1 is assumed.
DT9	DATE	SUBSTR(TO_CHAR(Decode(LENGTH(\0), 8, \0, NULL)), 1, 8)	INPUT(SUBSTR(\0, 1, 8), YMMDD8.)	DATE9.	SAS date format DATE9.

Macro Name	Question Data Type	SQL Macro	SAS Macro	SAS Format	Description
COMP_TM	TIME	SUBSTR (DECODE (SUBSTR (\0, 1, 2), NULL, '00', SUBSTR(\0,1,2)) DECODE (SUBSTR (\0, 3, 2), NULL, '00', SUBSTR (\0, 3, 2)) DECODE (SUBSTR (\0, 5, 2), NULL, '00', SUBSTR (\0, 5, 2)), 1, 6)	-	\$6.	Complete a partial time and display as HHMMSS.
TM8	TIME	SUBSTR (TO_CHAR (DECODE (LENGTH (\0), 4, \0, NULL)), 1, 4)	HMS (INPUT (SUBSTR (\0, 1, 2), 2.), INPUT (SUBSTR (\0, 3, 2), 2.), 0)	TIME8.	SAS time format TIME8.

Some Hints for SAS Users

In current SAS views produced by Oracle Clinical, Questions of data type TIME and DATE are viewed respectively as SAS variables of CHAR\$6 (character of length 6) and CHAR\$8 (character of length 8). In the SAS view, the TIME data is seen as HHMMSS, and the DATE data as YYYYMMDD.

SAS programmers and statisticians may prefer to have TIME and DATE Questions as true SAS TIME and DATE variables. Moreover, it may not be necessary, or possible, to complete a partial date or time according to a standardized algorithm. For example, data may require a sensitivity analysis with dates and times completed according to a variety of algorithms. The following approach supports these objectives.

1. Define two DATE Questions: one entered, called COLLD, and one derived, called COLLDT.
The derived Question has an extract macro, DT9, which formats the data as a SAS date variable, provided the date is already complete.
2. Enter data into COLLD, with the derivation expression for the derived Question, COLLDT, being the value of COLLD.

In the SAS extract view, COLLD is a CHAR8 variable that contains the entered data according to a YYYYMMDD character display, and COLLDT is a true SAS DATE variable. This translates into:

- If the value of COLLD is a complete date, then COLLDT contains the corresponding data.
- If the value of COLLD is a partial date, then the value of COLLDT is missing.

Defining Extract Macros Using the Maintain Extract Macros Window

Fields in the Maintain Extract Macros window are mandatory, unless stated otherwise.

If you want to edit an active extract macro, you must have selected the preference **Enable Edit of Active Extract Macros?**, which is displayed in the Data Extract Installation Configuration window (from the **Admin** menu, select **DX Installation Configuration**).

To define an extract macro:

1. Navigate to **Glib**, then **Data Extract View Builder**. The Maintain Extract Macros window opens.
2. Enter a name for the macro. This name must be unique in the Global Library.
3. Choose a status for the macro: Active (A), Provisional (P), or Retired (R). You can activate or delete Provisional macros, and retire Active macros.
4. Choose the expected data type of a response to a Question. Acceptable values are CHAR, DATE, NUMBER, or TIME.
5. Enter the SQL statement for this macro. Double-click in the **SQL Macro** field or click its LOV button to invoke the Editor.

SQL text can contain an embedded reference to database functions, keys, or variables. The variables may be patient data collected in the Response table--such as, Value Text, Exception Text, Data Comment Text, Data Change Reason Code, Audit Comment Text, Discrepancy Indicator, or Validation Status. You can choose components of a macro using the list of values for the SQL Macro item.

6. Enter a description of this extract macro. Entering descriptions for Global Library objects can be helpful to other users who want to use the objects you define. Use the following descriptions to complete the fields:

In the **Oracle Name** field, choose the Oracle name of the key columns in the view.

Length: Lists the maximum number of allowable characters for a response to the Question. This value cannot exceed: 200 for type CHAR; 40 for type NUMBER; 8 for type DATE; and 6 for type TIME. For a Question of type NUMBER, do not count a negative sign or a decimal point toward the length.

SAS Macro: Unvalidated SAS text referenced during generation of the SAS Pass Through view definition. For example, to apply the SAS function DATE1 to a Question value text, enter the expression DATE1(1) in the **SAS Macro** field. When associated with the Question ONSDATE, for example, the following appears in the .SAS file: DATE1 (ONSDATE).

A SAS macro references only one variable: the Oracle Extract View column. If you define a SAS macro as well as a SQL macro, the reference variable is always the same in both cases.

SAS Name: Mandatory when the macro references no response variables. Based on the Oracle name of the extract macro. Defined when you create a Question. The name cannot end with a number, and can be comprised only of uppercase letters, numbers, underscores, or hashes. It can be up to 8 characters in length, unless the Question has multiple occurrences in a single Question Group, in which case it can be only up to 7 characters.

SAS Label: Mandatory when the macro references no response variables. Free form text supplied to the **SAS Label** field in the SAS views and SAS datasets to describe the purpose of a Question.

SAS Format: Information for the format portion of the SAS Pass Through definition file.

A list of values is available, with values from the installation reference codelist SAS_FORMATS. You can use this codelist to store all the formats you might want to use for the creation of extract macros. Oracle formats are listed in the list of values with their equivalent SAS formats.

7. Save. Oracle Clinical validates the extract macro and adds it to the Global Library.

Changing a Macro Definition

When you change the status of a macro, either by activating or retiring it, you can enter more information about this change in the **Status Comment** field.

Listing Available Functions in the Database

Clicking the **List Function** button launches the Function Specification window, from which you can examine details about the functions to which you have access. These details include the function name, its owner, and the function's attribute names and data types. Use the Down and Up arrow keys to see different functions.

Creating Key Templates

In this section:

- [Maintain/Query Key Templates](#)
- [Customize/Query Key Template Columns](#)
- [Order Specification for Data Viewing](#)
- [Aggregation](#)

Maintain/Query Key Templates

You can select which keys to include in extract views by defining Key Templates. You define a Key Template by picking from previously defined key extract macros (from the **Glib** menu, select **Data Extract View Builder**, then **Extract Macros**) and then specifying how to name and display the keys in the template. You can control both the Oracle and SAS names for a column.

A predefined set of key extract macros is installed with Oracle Clinical. These extract macros form the basis of the standard Key Template provided with the application. You can define alternative Key Templates and instruct Oracle Clinical to use them by changing the name of the default Key Template in the Data Extract Installation Configuration form (from the **Admin** menu, select **DX Installation Configuration**) or by supplying a study-specific Key Template in the Clinical Study States form (from the **Conduct** menu, select **Security**, then choose **Clinical Study States**). If you want to be able to choose alternative Key Templates for different view definitions within a study, you can select **Enable Selection of Nonaggregate, Nondefault Key Template?** in the Data Extract Installation Configuration form.

Customize/Query Key Template Columns

By clicking the **Key Columns** button in the Key Template window, you can select the keys to include in the Key Template. You can use the **Up** and **Down** buttons to change the order in which the key columns are placed in views created with the Key Template. The keys are ordered in the view in the same order as they are displayed in the window.

Order Specification for Data Viewing

You can specify how to order data that is queried in the View Data window in the View Definitions form. Click the **Order By** button in the View Templates window. In the Data Order By Columns window, select from key columns that have already been added to the view, and specify the order in which the key values are used to order the query. The criteria you enter

for ordering in this window have no effect on data extract views: they affect the order in which data is returned in the View Data window.

To navigate with the arrows and buttons on the Data Order By Columns window, follow these instructions:

Table 14-2 Buttons and Actions in the Data Order By Columns window

Clicking...	Moves the Selected Column...
Up	Up one row
Down	Down one row
->	From Column Picklist to Selected Order By Data Columns
<-	From Selected Order By Data Columns to Column Picklist

Clicking the <<< button moves all columns in Selected Order By Data Columns to Column Picklist.

Aggregation

If you have added key columns to the Key Template, the **Aggregate?** box is automatically selected when the keys do not completely define a unique repeat of data. The set of keys that uniquely define one row of data in a view are as follows. If you choose the key DOCNUM, it includes PATIENT_ID and VISIT; if you choose the key ACTUAL_EVENT, it includes VISIT and SUBEVENT.

PATIENT_ID	VISIT	SUBEVENT	QUALIFYING_VALUE	REPEAT_SEQUENCE_NUMBER
DOCNUM				
	ACTUAL_EVENT			

Only nonaggregate Key Template can be used for view definitions intended to create data extract views.

Note:

If you are using Fast Views and an aggregate key template, you must enter the text "AGGREGATE" somewhere in the template's Status Comment field. Fast View structure is incompatible with aggregate key templates. If you enter the word "AGGREGATE" in the Status Comment, the system recognizes it and creates a regular view structure. See the *Oracle Clinical Administrator's Guide* chapter on Data Extract configuration for information on Fast Views.

Create Questions and View Templates

Creating Questions is of itself not a data extract task. However, customizing data extract can include creating complex Questions and mapping their child Questions to extract macro variables, or extending attributes on simple Questions and mapping selected attributes to extract macros. For more information, see these sections in this chapter: [About Extended Question Attributes](#), [About Extract Macros](#), [About Complex and Simple Questions](#), and [Example of Associating Extract Macros to Questions](#).

In this section:

- [Associating Questions with Views](#)

Associating Questions with Views

Navigate to the View Definitions window (from the **Definition** menu, select **Data Extract View Builder**, then choose **View Definitions**), then follow these steps:

1. Query for the view definition that you want to work on.
2. Select its template and either double-click on it in the **View Template** field or click the **Details** button. Oracle Clinical opens the View Template Details window, which lists Question names associated with the selected View Template, the response attributes for the Question, and whether they are complex.
3. The upper part of the View Template Details window serves as a tool with which you can restrict the possible Questions that you are going to map to the View Template.

In this upper part of the window, choose either the **Pick by Question** radio button or the **Pick by DCM Question** radio button. The Question referred to in **Pick by Question** is any library Question, simple or complex.

4. Go back to the View Definitions window to map the Questions of the complex Question to DCM Questions. Select a view definition and click the **Map** button. The Maintain View Question Mappings window appears.
5. The upper part of the View Mappings window serves as a tool with which you can restrict the possible DCMs and DCM Question Groups you are going to map to the DCMs and DCM Question Groups that are accessible to the study. If you select the radio button **DCM Filter ON**, a table is displayed with the fields where you can enter your restrictions. Selecting the radio button **Clear Filter** erases what you enter.

Maintaining View Definitions

The Maintain View Definitions window enables you to:

- Review and modify an existing view definition, created by defaulting or copying.
- Create a new view definition, using an existing study or library View Template.
- Create a new view definition and its View Template at the same time.
- View data from a new or existing view definition.

If you create a new view definition and reference an existing View Template, you can invoke the Map window to associate, or map, the columns that are specified in the View Template to specific DCM Questions in particular DCMs.

If you want to add columns to a view definition, or modify the specified Question attributes, you can invoke the View Template form by double-clicking on the **View Template** field. The View Template form behaves the same way as when you call it from the menu, with one important difference: Questions you add to the View Template by selecting from DCMs are automatically mapped to that DCM in the calling view definition.

You can create a View Template and view definition simultaneously by double-clicking an empty **View Template** field. Oracle Clinical automatically creates a new View Template with the same name as your current view definition and then places you in the Template Questions definitions window. As you add columns to the View Template by selecting from DCM Questions, the Questions are automatically mapped to the DCM in your view definition.

You can add Questions from either the DCM or the Global Library. You cannot add columns directly to the View Template, but when you add Questions to it, the system automatically maps the Questions to the DCM in your view definition. For example, if you add a DVG Question, Oracle Clinical automatically adds the columns for value_text or dvg_short_name for each Question. The automatic mapping of View Template Questions with the DCM Question works only if you add the Questions using the DCM Questions filter.

You can add more attribute columns by clicking the **Attribute** boxes for each View Template Question, provided they are defined in the Questions form (Global Library).

When you create or modify a view definition, you must also specify which pre-existing Key Template to use. You can also optionally use a previously created Where clause to restrict the view. When you batch load views, the display-only field **Source File**, in the View Definitions form, reflects the fully specified path of the operating system source file for the batch load.

For more information, see:

- [Linking a View Definition to a DCM](#)
- [Viewing the Data for a View Definition](#)
- [View Model](#)
- [Creating View Templates](#)
- [Mapping DCM Questions to View Template Questions](#)
- [Having Clauses](#)
- [Copying a View Definition](#)

Linking a View Definition to a DCM

Link modes govern the way that DCMs are tied to View Definitions for update purposes and how re-defaulting View Definitions affects these relationships. DEFAULT and LINKED Link Modes apply only to View Definitions that reference exactly one DCM. The link modes are:

LINKED - Changes to the DCM definition will also update the view definition.

NONE - Changes to the DCM will not affect or update the View Definition in any way. This mode can provide a useful method of preventing a cascade effect in a Cross-study data extract.

DEFAULT - Oracle Clinical refers to the default linking behavior defined in the DX Installation Configuration window accessed from the **Admin** menu. In this window, if the DCM Default Views Are Linked to Source DCM as Default Condition? setting is enabled, a view definition will be linked to the DCM; if this setting is clear, no linkage will exist. You cannot choose DEFAULT as the link mode once you save a view definition with either of the other two modes.

If a view definition's link mode is either DEFAULT or LINKED, and you add a Question to the DCM, you must either update the view definition manually (from the **Special** menu, select **Update View Definition**) or perform a View Maintenance. INCREMENTAL View Maintenance detects only the changes that are made since the last maintenance and synchronizes them with the DCM definition. FULL View Maintenance recreates the study access account.

Viewing the Data for a View Definition

When you click the **View Data** button in the Study View Definition window, you can see the data that the selected view definition would output. The window is divided into the keys that make up the view definition you selected and the Questions that make up the View Template of that view definition.

This button works only if you have chosen a Key Template and a View Template, and if you have mapped the Questions to the View Template. You see an alert specifying the problem if you have not completed the view definition. The button will also not work if you do not have a user account for the default study access account.

You can print from this window, and you can save the SQL view for the view definition in the USER_QUERIES table, provided you have this table set up for you by the system administrator.

View Model

When you click the **View Model** button in the View Definition window, you can see the components of the selected view definition: the key or Question name; the optional SQL macro; the attributes of Questions; sort order (for ad hoc data retrieval only); the names of the tables from which data is extracted; and any optional Having and Where clauses.

Creating View Templates

You can create View Templates in any of the following ways:

- Accept a default view definition by choosing **Default View Def** from the **Special** menu of the DCM form or by clicking the **Create Default View Definition?** button in the **Alert** box that comes up when you activate a DCM.
- Associate View Templates in the View Templates form, at the Library and study levels, with DCM Questions that you select.
- In the View Definitions window, query for the View Definition you want to work on, then select its template and either double-click on it in the **View Template** field or click the **Details** button.

In addition, to add a complex Question, you should select the Global Library Filter.

Mapping DCM Questions to View Template Questions

In the upper part of this window, when the button labeled **DCM Filter ON** is selected, list the DCM Question Groups that contain the Questions you want to include in the View Template.

When the button labeled **DCM Filter OFF** is selected, which is the default, you do not see the fields to the right of the buttons and there are no restrictions on which Questions can be associated with the View Template. You can choose the **Clear** button to erase what you enter. The **Clear** button is not visible when the filter is off. The lower part of the window lists the Questions available to a View Template in the selected study, given any choices you may have made in the filter area.

Having Clauses

Having clauses restrict data. You can create these clauses from the View Definitions window in the Global Library or at the study level by pressing the heading on the column Having, which brings up the Having window. Having clauses have no status and can be changed at will.

When there is more than one argument, each argument is enclosed in parentheses. Functions available include GREATEST, LEAST, UPPER, and LOWER. The Questions made available by the List function are Questions referenced in the View Template. The expression choices are LIKE, NOT LIKE, IN, NOT IN, =, <>, >, and <. You add arguments with the AND/OR option.

An example: (RACE+BLACK) OR (GREATEST AGE>65)

Copying a View Definition

When you copy a view definition, the elements of the view definition behave differently, depending on where they were created—at a study level or at the Global Library level—and the source/target relationship.

A view definition comprises a View Template, a Key Template, and optional Having and Where clauses. Having clauses are never copied with the view definition. Key Templates are always created in the Global Library, so you copy the reference to the Key Template, not the Key Template itself. View templates and Where clauses follow these rules:

- Copying a Library view definition into a study means that you reference the View Template and Where clause.
- Copying a study view definition into a study or the Global Library means that you copy the View Template and the Key Template as well as the view definition.

Copying a view definition would create two view definitions with the same name in the same study; to address this naming issue, Oracle Clinical automatically gives the copied view definition the name `COPY_OF_name`, where *name* is the name of the source view definition. The system alerts you to change this name in the View Definition window, which appears with the copied view definition when you click **OK** in the Copy View Definition window. For a Library view definition, the name must be unique in the first domain found in the user domain searchlist or in the default domain searchlist. When the name is not unique, the system alerts you.

Defining Study Sets

To create a study set, from the **Definition** menu, select **Data Extract View Builder**, then **Study Sets**. First give the study set a name, assign it an account type, and write in a description, then use the **Define** button to select the studies you want to include in the set. You must use the **Define** button if you choose **STUDY SET** or if you want to limit the studies available when you choose **PROGRAM** or **PROJECT**.

The order in which you select studies to form the study set has no effect on which study access account is associated with the Cross-study Union.

The name of the Cross-study View is associated with the name of the study you were working in when you opened the Maintain Study Sets form. The study name listed in the blue banner across the top of the window disappears once you give the study set a name.

Creating Union Views

A Union is a vertical grouping of multiple views. You follow a similar process in creating Union views whether they are within a study (In-study Union) or in more than one study (Cross-study Union). However, studies chosen for Cross-study Union must be in a study set, and views showing data across studies must share the same type of:

- View Template
- Key Template
- Study Account

Only views built with Global Library-defined View Templates can be unioned for a study set; whereas views built with either Global Library- or study-defined View Templates can be unioned within a study. You can interactively query view definitions both in the study and in the study set.



Note:

Union views cannot be represented as tables (see the "Data Extract" chapter of *Oracle Clinical Conducting a Study* for details on table views).

For more information, see:

- [Creating a Cross-study Union](#)
- [Troubleshooting Cross-study Union Views](#)
- [Creating In-Study Union Views](#)
- [Example of Union Views](#)

Creating a Cross-study Union

To define a Cross-study union:

1. Define a View Template with Questions in the STANDARD domain of the Global Library.
2. Create a DCM that includes all of the Questions from the first instruction, for each study.
3. Copy the same view template from the Global Library for each study.
4. Create a view definition of the same name in each study, from the template you created in the first step.
5. Perform Initial Log-in and data entry for each study.
6. Run Data Extract View for each study: Navigate to **Conduct**, then **Data Extract**, and finally **Data Extract Views**.
7. Create a study set by navigating to **Definition**, then **Data Extract View Builder**, and finally **Study Sets**, first giving the study set a name, assigning an account type, and

writing in a description, then using the **Define** button to select the studies you want to include in the set. You must use the **Define** button if you choose **STUDY SET** or if you want to limit the studies available when you choose **PROGRAM** or **PROJECT**.

The order in which you select studies to form the study set has no effect on which study access account is associated with the Cross-study Union.

8. Check the access account for the study set by navigating to **Conduct, Data Extract**, then **Study Set Access Accounts**, remembering to make any TABLE types of view into VIEW types.

See the "Data Extract" chapter of *Oracle Clinical Conducting a Study* for details on table views.
9. Define a cross-study view by navigating to **Definition**, then **DX View Builder**, and finally **Cross Study Union**.
10. Create the cross-study union view by navigating to **Conduct, Data Extract**, then **Data Extract View for Study Sets**, and submitting it.
11. Query the view in SQL*Plus.

Snapshot account naming becomes important for Cross-study Union views. For example, if you name a snapshot account BASEA in STUDY1 and you want to compare the same period that the snapshot covers in another study, STUDY2, then you need a snapshot account in STUDY2 with the same name as the snapshot account in STUDY1. The two snapshot accounts may have different batch validation timestamps, but because of the naming, in a Cross-study Union their data will be treated as if it were processed at the same time.

Troubleshooting Cross-study Union Views

An error may arise if you do not use the same Key Template and Global Library View Template, and in the same domain, for the study level views and the Union view. The error follows:

```
Update_view_account_statistics invoked
Acct: XXXVBS$CURRENT, View_id: 10101, status: INVALID
Inserting view_acct_stat
Update_view_account_statistics returned success
Invoking get_union_view_text with 10101,2001,82201
ERR level 1
get_union_view_text
Error: unable to get union text
ERR level 1
create_cross_study_views
Error when calling update_view_acct_statistics
ERR level 1
maintain_account
Error when calling create_cross_study_views
Connect prvl user.
Connected to ORACLE as user /.
Connected to ORACLE as user RXC_MAA.
RXCDXBVB EXIT WITH FAILURE
```

This error will appear in the .out file that Oracle Clinical creates when you run the Cross-study Unions Maintenance batch job. This batch job is available under the **Conduct** menu, by selecting **Data Extract**, and then **Data Extract Views for Study Sets**.

Creating In-Study Union Views

In-study Unions are vertical groupings of multiple views within a study. To create an In-study view properly, start by performing the following steps:

1. Create and activate the two DCMs. Do not create a Default View Definition for either one.
2. Create and activate a View Template for the Union view that contains all the required Questions. These Questions must exist in both source DCMs.
3. Create two or more view definitions using this newly created View Template. The data for all the view definitions using this common View Template is displayed in the in-study union view.
4. Map each view definition to the respective DCM and activate the view definitions.
5. Confirm that the Key and View Templates are the same for all the view definitions that are being unioned.
6. Log in data for both DCMs.
7. Run the Data Extract View.

When these steps are complete, create the In-study Union view by following these steps:

1. Navigate to **Definition, Data Extract View Builder, and In-study Unions**. If you have not selected a study during your session, Oracle Clinical prompts you to choose one now. The Maintain In-study Unions window opens.
2. Enter a name and description for this Union view.
Create an In-study Union by navigating to **Definition, Data Extract View Builder, In-study Unions**, and giving the Union a name and description.
3. Select a View Template that was used in the underlying view definitions of the Union view.
4. Select a Key Template that is nonaggregate and was used in the underlying view definitions of the Union view.
5. In the **Role Access** field, choose **RXCLIN_MOD** or **RXCLIN_READ**, which determines your role access.
6. Save.
7. Generate the data extract views.

You may want to run the Data Extract View Maintenance batch job at this point. See [Troubleshooting Cross-study Union Views](#).

Example of Union Views

A Global Library View Template, AE, has the columns AE, STARDT, STOPDT, and SEVERITY. Two studies, OCLIN1 and OCLIN2, share the View Template AE and the Key Template STANDARD.

In tabular form, these statements can be represented in more detail, as follows:

Study	DCM	View	View Template	Key Template
OCLIN1	SIGNSYMP	SIGNSYMP	AE	STANDARD

Study	DCM	View	View Template	Key Template
OCLIN1	AE	AE_OCLIN1	AE	STANDARD
OCLIN2	AE	AE_OCLIN2	AE	STANDARD

In-study Union for study OCLIN1, from a Union of views AE_OCLIN1 and SIGNSYMP is ALL_AES.

Cross-study Union of views AE_OCLIN1 and SIGNSYMP from study OCLIN1 and the view AE_OCLIN2 from OCLIN2 is ALL_OCLIN_AES.

Creating Where Clauses

Where clauses are optional methods by which you can restrict the data retrieved from DCMs/RDCMs by key values--that is, the key columns. These restrictions parallel those imposed by Having clauses, which allow you to restrict the data retrieved by response values to Questions.

Where clauses can be created at the study level or in the Global Library by navigating to either:

- **Glib**, then **Data Extract View Builder**, then **Where Clauses**
- **Definition**, then **Data Extract View Builder**, then **Where Clauses**

They are created with a status of Provisional and can be tested in a view definition with that status. In the Where clause window you can choose available key values via the List function.

The list of values for expressions includes LIKE, NOT LIKE, IN, NOT IN, =, <>, >, and <. In the **Value** field, you can enter an appropriate value or you can select a value from the list of values. If you have more than one argument, enclose each argument in parentheses. You add arguments by means of AND/OR.

An example of a Where clause:

```
(RDCM.PATIENT=5) AND (RDCM.VISIT_NUMBER=1)
```

You can modify and delete Where clauses, and you can change their status from Provisional (P) to Active (A) and from Active to Provisional or Retired (R).

For more information, see:

- [Copying Where Clauses](#)

Copying Where Clauses

When you copy a Where clause, you need to be sure, as always when copying from the Global Library, that the domain of the object you want to copy is in your user domain searchlist. If the domain is not in this searchlist, you can add it.

Adding DISTINCT terms

You can generate the extra "distinct" term in each column in the extract views. This change can decrease the performance of the views, so you should only use this approach if you experience the "sort key too long" error and you cannot increase your database blocksize.

To adjust the setting, navigate to **Definition**, then **Data Extract View Builder**, and finally **View Definitions** and include the text `DISTINCT` in the Status Comment for each view that requires a distinct clause. For instance, change `max(decode...` to `max(distinct decode...` When you have made your changes, rebuild existing extract views in FULL mode.

About Joining Fast Extract Views

When you join fast extract views to each other, it is important to include an optimizer hint specifying a hash join. Joins without optimizer hints may be up to ten times slower than joins with optimizer hints, and may even be slower than for joins between "slow" views. The following example shows a join between `PHYSICAL EXAM` and `VITAL SIGNS`, based on a common patient visit:

```
select /*+ ordered use_hash(v) no_merge(v) */
pe.pt, pe.actevent, pe.repeatsn, pe.exam, pe.result, v.blood_pressure_syst
from oraclin$current.pepe pe, oraclin$current.vit v
where v.pt=pe.pt and v.actevent=pe.actevent order by 1,2,3;
```

General Example

The goal of this example is to perform the following tasks:

1. create a patient-level view.
2. create one row per patient, with enrollment date and sex from `DEMOGRAPHICS`, and termination date from the Termination form.

The challenge is to create an appropriate Key Template. Oracle Clinical requires that a Key Template include the following columns to be considered nonaggregate:

Patient
Qualifying Value
Visit Number
Subevent Number
Repeat Number

An aggregate key can be used interactively when you click the **View Data** button in the View Definitions window, but not for a permanent view. Therefore, a Key Template to be used for a permanent view needs to include all these columns.

However, if a Key Template were created with these required columns, it is not possible to obtain one row per patient, since the `DEMOGRAPHICS` DCM and the `TERMINATION` DCM have different visit numbers.

Solution

Create an extract macro that uses each of these key columns, but collapses them to the same value. The extract macro is then added to the Key Template, which means that the extract macro contains the required columns and the Aggregate flag is therefore turned off. Now the Key Template can be used.

The steps might be as follows:

1. Create the following CHAR extract macro named `PATIENT_SUMMARY`:

```
decode(<QUALIFYING_VALUE>, '?', null, null) ||
decode(<REPEATSN>+ <VISIT_NUMBER>+ <SUBEVENT_NUMBER>, 0, 'N/A', 'N/A')
```

This always collapses to a value of N/A.

2. Create the Key Template to include PATIENT and PATIENT_SUMMARY, and make it active.
3. Create a View Template with the required Questions (ENROL_DATE, SEX, and TERM_DATE).
4. Create a view definition that links to this View Template and uses the newly created Key Template.
5. Map the Questions to the appropriate DCM Questions.
6. Make the view definition active.
7. Generate the extract views.

If there is more than one value for a DCM Question, the View Builder applies a MAX function. Therefore, if the view also needs to include minimum dosing date and maximum dosing date from a dosing DCM, there is no way to do this entirely within the View Builder (since there is no way to get the minimum dosing date). You need to create minimum dosing date and maximum dosing date as derived Questions in the DEMOGRAPHICS DCM, populate them in an appropriate Derivation Procedure, then include the derived Questions in the View Template.

15

Lab Ranges

Lab ranges allow you to check the validity of lab data in a validation procedure. You can set standard ("textbook") ranges of acceptable values for lab test Questions according to company standards. If a particular study needs to use a different range of values for a particular Question, you can define an overriding lab range for that Question. In either case, you also enter information into the system about the labs that are processing the tests, and associate the appropriate lab ranges with the labs.

If a lab test result is out of range and a validation procedure is defined to detect it, Oracle Clinical creates a discrepancy in the Discrepancy Database that must be examined by clinical personnel.

You also enter information about each acceptable measuring unit for each Question, which unit is preferred, and how to convert to that unit so that all the data for the Question can be easily compared.

Before you begin working with the functionality available in the Lab Ranges subsystem, you must have all the required lab test Questions defined in the Global Library (see [Questions and Question Groups](#)).

You can batch load lab-related definitions, which is described in the final section of this chapter.

For replication across a distributed study, some lab information (Lab Units and panels) must be defined at the Global Library location, while other lab information (labs, lab ranges, and lab assignment criteria (LACs)) must be defined at the study-owning location.

For more information, see:

- [Establishing Units of Measure](#)
- [Defining Standard Lab Ranges](#)
- [Lab Assignment Criteria](#)
- [Batch Loading Lab Ranges](#)

Establishing Units of Measure

For each lab test Question you must define each unit of measure acceptable for that Question, and how to convert Lab Units to a preferred unit. The following steps are required:

- [Defining Lab Units](#)
- [Defining lab test Question Units](#)
- [Defining Unit Conversions](#)
- [Defining Preferred Conversion Groups](#)

Defining Lab Units

You must define all acceptable units of measure for all lab test Questions in the Maintain Lab Units window.

If you collect data using different units whose value is exactly the same (for example, percentages and decimal values), you can group those units into synonym groups by giving them the same Representative Unit Code. To do this, you must define a unit with the same name you want to use as the Representative Unit Code. This Code must not be longer than ten characters, and should be meaningful because it is what you will see in other Oracle Clinical lab definition windows. When you save this unit definition, its value becomes available to other units in the **Representative Unit Code** field list of values.

You have the option of adding free form text to help in pronunciation. Lab unit pronunciation appears only on this form and on the Lab Units Report.

You must define Lab Units at the Global Library location. You can add a new unit at any time. You can change a unit only if it has not yet been saved, and delete one only if it is not referenced by a lab test Question, but you can choose to make a unit inactive at any time.

To define Lab Units:

1. Navigate to **Labs, Lab Units**, and then **Lab Units**. The Maintain Lab Units window opens.
2. In the **Lab Unit Code** field, define the name of the Lab Unit. For example, g/ml.
3. In the **Lab Unit Name** field, enter a descriptive name for the Lab Unit.
4. Enter a **Representative Lab Unit Code**. These codes enable you to define Lab Units as equivalent in the system. For example, because the Lab Units mg/ml and mg/cc are equivalent, you can define mg/ml as their common Representative Lab Unit Code. You can enter a code of up to 40 characters.
5. (Optional) Describe how this Lab Unit is pronounced in the **Lab Unit Pronunciation** field.
6. Select the **Active?** box to enable this Lab Unit to be used, or clear it to hide it from the system.
7. Save. Oracle Clinical commits this Lab Unit to the database.

Defining lab test Question Units

When you have defined all necessary Lab Test Units, associate each lab test Question you will be using with as many units of measure as apply, but at least one.

Lab units can be selected only by their representative Lab Unit code; however this code represents all the synonyms in the group—that is, all synonyms in the group are valid for the lab test Question.

In the Maintain lab test Question Units window you can insert representative Lab Units and delete those not referenced from Lab Unit conversions, ranges, Preferred Copy Groups, or preferred conversion groups. If the Lab Unit is referenced, you can choose to make it inactive. The **Active?** field controls whether the Lab Unit appears on lists

throughout the Lab Ranges subsystem when Lab Units are called for. If a unit is marked inactive, it does not appear on lists.

To define lab test Question units:

1. Navigate to **Labs, Lab Units, and LTQ Units**. The **Maintain lab test Question Units** window opens.
2. Query for the lab test Question to which you want to associate units.
3. Define each representative Lab Unit name that you want to use as a unit measure for this Question, and either select the **Active?** box to enable this Lab Test Unit for the Question or clear it to render this unit inactive for this Question.
4. Save. Oracle Clinical enables you to use the representative Lab Test Unit or Units when you enter lab data for this Question.

Defining Unit Conversions

You must define a unit conversion for each Question that has more than one acceptable unit of measure. You can write a Derivation Procedure that references this conversion formula (see [Validation and Derivation Procedures](#)).

If your study collects lab data for the same Question in multiple unit types (for example, kilos in Europe and pounds in the U.S.), you can specify conversion formulas to ensure that the data can be analyzed and reported using a single unit of measure. Use the Lab Unit Conversions window to set up standard conversion definitions.

The Lab Unit conversion must be unique in both source (the **From:** Lab Unit Name) and destination (the **To:** Lab Unit Name); the system checks for uniqueness when you leave the Lab Unit conversion record.

Always specify the conversion TO your preferred unit of measure, the one you will use to analyze the data. Later, in the **Maintain Preferred Conversion Groups** window (referenced in [Defining Preferred Conversion Groups](#)), you specify which unit is preferred for each Question.

The conversion must be expressed in terms of multiplication and/or addition. To divide, multiply by a fraction; to subtract, add a negative number. However, you cannot enter a multiplier of zero or a negative number; doing this results in an error message. The system issues warning messages in the following circumstances:

- The inverse of the reverse multiplier is not within 1/10000 of the new multiplier.
- Another multiplier for the same lab test Question is within 1/10000 of the new one.

To define a Lab Unit conversion:

1. Navigate to **Labs, Lab Units, and LU Conversions**. The Maintain Lab Unit Conversions window opens.
2. Query for the lab test Question for which you want to define conversions.
3. Query for a Lab Test Unit you want to use as the **From** value. You can only choose Lab Test Units that have been defined for this Question in the **Maintain lab test Question Units** window (referenced in [Defining Preferred Conversion Groups](#)).
4. Query for the Lab Test Unit you want to use as the **To** value. In addition to being defined as a Lab Test Unit for this Question, you must also choose a different unit than you selected as the **From** value.
5. Choose a **Conversion Source**.

6. Enter the **Multiply Number** and/or **Add Number** for this unit conversion.
7. (Optional) Enter a comment that describes this conversion further.
8. Save. Oracle Clinical saves this Lab Unit conversion for this Question.

For more information, see:

- [Testing a Lab Conversion Calculation](#)

Testing a Lab Conversion Calculation

Once you have completed the required fields and saved the definition, you can click the **Test Conversion** button to try it out.

Defining Preferred Conversion Groups

A preferred conversion group is a set of lab test Questions and the preferred Lab Unit of measure for each Question. The preferred conversion group, together with the relevant Lab Unit conversion formulas, can be used by a Derivation Procedure to convert a response to the lab test Question collected in one unit of measure to a derived value in the preferred standard unit of measure (see [Validation and Derivation Procedures](#)).

1. Navigate to **Labs, Lab Units, and Pref Conv Groups**. The **Maintain Preferred Conversion Groups** window opens.
2. Define the unique group ID in the **Group** field. This ID can be up to 10 characters long, and you cannot change it once you save the group.
3. Define a **Group Name**.
4. (Optional) Enter a comment to describe supplementary information about this preferred conversion group.
5. Enter each Preferred Lab Unit for this group:
 - a. Query for a lab test Question. When you choose one, Oracle Clinical populates its domain as well.
 - b. Query for a Lab Unit Name associated with that Question.
 - c. Enter the number of decimal places that you want reports displaying this value to use. Once you save, you cannot change this value.
 - d. (Optional) Enter a comment with supplementary information about using this preferred Lab Unit for this group.
6. Save. Oracle Clinical commits this preferred conversion group to the database.

Defining Standard Lab Ranges

Once Oracle Clinical has the Lab Unit information to draw from, you can define a standard ("textbook") lab range for each lab test Question and put them into Preferred Copy Groups to copy them for use in multiple studies. If you need to override your company's standard range for a particular Question in a particular study, you can do so by navigating to **Labs**, then **Labs, Labs, and Ranges**.

This section includes the following steps:

- [Defining Textbook Ranges](#)

- [Defining Preferred Copy Groups](#)
- [Defining Lab Panels](#)
- [Defining Labs](#)
- [Defining Specific Lab Ranges](#)
- [Defining Lab Subsets](#)
- [Using Textbook Ranges](#)
- [Copying Lab Ranges](#)

Defining Textbook Ranges

Textbook lab range definitions serve as the company standard range for lab test Questions. They provide a default definition for each lab test Question where no lab-specific ranges have been defined. You can also use these definitions as a foundation from which to create more specific lab range definitions for particular labs.

Oracle Clinical enforces uniqueness for textbook ranges in a similar way to lab ranges, but considers one more column: the Lab Unit Name. Thus, each textbook range must be a unique combination of lab test Question, Domain, Sex, Maximum Age, Minimum Age, Effective Start and End Date, and Lab Unit Name.

The Maintain Textbook Ranges window only displays the representative unit names. Representative Lab Test Units are defined when you define lab test Question units; see [Defining lab test Question Units](#) .

You can use the Maintain Textbook Ranges window to delete textbook ranges; however, you cannot delete a textbook range if a Preferred Copy Group uses it.

To define a textbook range:

1. Navigate to **Labs**, then **Labs**, and **Textbook Ranges**. The Maintain Textbook Ranges window opens.
2. Query for the Question for which you want to establish textbook ranges. The system displays only Questions that have a Question type of Lab Test.
3. Enter each textbook range for this Question:
 - a. Enter the gender to which this range applies. Valid choices are Male (M), Female (F), or Both (B).
 - b. Enter the minimum and maximum age ranges. You can leave either end of the age range blank, if no minimum or maximum age applies for this range.
 - c. Enter the low and high values for this textbook range, and the **Lab Unit Code** for the measurement you are specifying.
 - d. (Optional) Enter the dates during which the textbook range is in effect. You can leave one or both of these blank.
4. Save. Oracle Clinical commits these textbook ranges to the database for this lab test Question.

Defining Preferred Copy Groups

A Preferred Copy Group is a named set of lab test Questions and the units of measure that each question uses. Using Preferred Copy Groups enables you to create and copy groups of

textbook ranges; each time you add a lab test Question to a Preferred Copy Group and choose a preferred unit of measure, Oracle Clinical associates the textbook ranges you have defined for that Question with the copy group. When you copy textbook ranges into a lab definition, the Use Textbook Ranges window only allows you to use textbook ranges that are grouped into Preferred Copy Groups.

You can create Preferred Copy Groups to group Questions that often need to be copied together when applying ranges to a lab; or for lab test Questions that are identical except for, perhaps, the unit of measure measured.

Differences in units of measure arise when you have locations in Europe and also in the United States. To handle these differences you might have a Preferred Copy Group called US and one called EURO. For example, the Hematocrit test would likely be collected as a fraction in Europe and as a decimal in the U.S., which you would translate as two identical textbook range definitions for Hematocrit:

Hematocrit	Male	42-54	%	Female
38-46	%			
Hematocrit	Male	.42-.54	fraction	
Female	.38-.46	fraction		

Preferred Copy Groups and Preferred Conversion Groups (from the **Labs** menu, select **Lab Units**, then **Pref Conv Groups**) take almost the same information about lab test Questions and their preferred units of measure, but are used for different things. Conversion groups are used by Derivation Procedures and copy groups are used to copy lab Questions and their preferred unit and textbook range for use in a new lab; see [Defining Preferred Copy Groups](#) and [Defining Preferred Conversion Groups](#).

To define a Preferred Copy Group:

1. Navigate to **Labs**, then **Labs**, and **Pref Copy Group**. The Maintain Preferred Copy Group window opens.
2. Enter a unique ID for this group. This ID can be up to 10 characters long.
3. Enter the name for this group. This name is required, but you can change it even after you save the Preferred Copy Group.
4. (Optional) Enter a comment with supplementary information about the Preferred Copy Group.
5. Enter each lab test Question and its preferred Lab Unit combination that you want to include in this Preferred Copy Group:
 - a. Query for the lab test Question. When you choose one, the Maintain Preferred Copy Group window also populates its domain.
 - b. Query for the Lab Test Unit you want to use for this Question.
 - c. (Optional) Enter a supplementary comment about including this Question and unit in the Preferred Copy Group.
6. Save. Oracle Clinical commits this Preferred Copy Group to the database, making it available for use in the Use Textbook Ranges window.

Defining Lab Panels

To define a lab panel:

1. Navigate to **Labs**, then **Labs**, and **Lab Panels**. The Maintain Lab Panels window opens.
2. Define a unique identifier for the lab panel, not greater than 10 characters, in the **Panel** field.
3. Define a unique name for the lab panel, not greater than 60 characters, in the **Panel Name** field.
4. (Optional) Enter a comment to describe this lab panel. Descriptions in this field can be useful to other users browsing through lab panel records.
5. Enter the lab test Questions for this lab panel in the lower part of the window. For each lab test Question:
 - a. Click in an empty row. The Maintain Lab Panels window populates the sequence number (**Seq #**) field with the next unused number.
 - b. Enter a SAS Name for a Question defined and active in the Global Library with data type NUMBER and type LAB TEST.

When you choose a Question, Oracle Clinical automatically populates the **lab test Question** and **lab test Question Domain** fields with the corresponding information for that SAS Name.
 - c. Choose a lab test Question Type.
 - d. (Optional) Enter a comment about including this lab test Question in the Lab Panel.
6. Save.

You can modify existing lab panels in a similar way. Query for the lab panel you want to change using the fields in the upper part of the window, and add, remove or change lab test Questions.

Defining Labs

This section describes how to define the basic lab settings using the Maintain Labs window. A Lab Definition consists of the necessary contact and related information, and the lab ranges to be applied to data from that lab. You have to define the information in this section before you establish the lab's ranges.

Use the Maintain Labs window to launch these tasks: define lab ranges and lab range subsets, use textbook ranges in the lab definition, and copy lab ranges.

You can also delete labs using the Maintain Labs window; however, you can only delete a lab if it is not referenced by any received DCMs or lab assignment criteria.

To define a lab:

1. From the **Labs** menu, select **Labs**, then **Labs**. The Maintain Labs window opens.
2. Enter the **Lab ID** and its name. The ID must be no greater than 10 characters, all uppercase.
3. Select the **Active?** box to activate the lab, or clear it to render it inactive. Only active labs can supply lab ranges for the lab data entered or batch loaded for a study. You can change a lab's active status at any time.
4. **Ranges Approved:** If this field is checked, all ranges for this lab have a status of Approved. This is a display-only field.
5. **Format Mask:** Choose the default file format mask you want to use for batch loading data from this laboratory.

6. **Owning Location:** Name of the Oracle Clinical database that owns this lab data. This is a display-only field.
7. **Replicate?:** If checked, this lab must be replicated from an Oracle Clinical database other than your current database. You can change this flag from unchecked to checked at any time, but once checked, you cannot uncheck it.
8. **Last Replication:** Date and time this lab was last replicated from the Oracle Clinical owning location. This is a display-only field.
9. Enter the necessary information for the person who serves as the contact for this lab: the contact's first and last name, phone number and fax number.
10. Enter the lab's full mailing address.
11. Enter the start and end dates for this lab's accreditation.
12. Enter the certification agency and number for this lab, if applicable.
13. (Optional) Enter a supplementary comment to describe this lab.
14. Save. Oracle Clinical commits this lab definition to the database.

Defining Specific Lab Ranges

Use this window to specify ranges for this lab that are different from your company's textbook range for particular Questions.

As in textbook ranges, you can define a different range of valid values for each Question depending on the age or gender of the patient. Each range in the lab must have a unique combination of lab test Question, sex, maximum age, minimum age, and effective start and end date. If any range definition has a blank field for Lab Unit, low range, or high range, the range status is automatically set to MISSING, in which case the range is considered incomplete and not used for calculations.

The ranges you define here, and only the ranges you define here, will be available for Subsets definition (see [Defining Lab Subsets](#)).

For a special use of the Ranges window, see [Using Textbook Ranges](#).



Note:

If you change any part of a lab range definition that has been used against production data, you must immediately execute the PSUB job; from the **Labs** menu, select **Lab Procedures Re-Execution Pre-Process**. If you do not run this job, your changes will not be applied to existing data.

For more information, see:

- [Age](#)
- [Effective Date](#)
- [Lab Range Records without Effective Start or Effective End Dates](#)
- [Defining Lab Ranges Using the Maintain Lab Ranges Window](#)

Age

If you enter a value in the optional **Age** field, the value must be greater than zero. The minimum age is inclusive, and maximum age is exclusive. For example, an age span of 10 to 20 years would apply to patients from the day they turn 10 years old through the day before their twentieth birthday. When they turn 20 this lab range would no longer apply to their data for this Question. A set of age spans for ranges for a particular lab test might be 10-20, 20-30, 30-55. These spans do not overlap. Ages may be expressed in days, months, or years; the **Age Unit** field is to the right in the window.

Effective Date

Unless there is a specific date entered in either the **Start** or **End** field, the range is assumed to be continuous from the point it is defined. If you enter an end date, then you must define a range to start after that; otherwise the textbook range becomes the active range.

Two ranges can coexist if describing ranges for patients of different ages:

Table 15-1 Example of Non-conflicting Date Ranges

Lab Test Question	Sex	Minimum Age	Maximum Age	Minimum Age Unit	Maximum Age Unit
CHOLESTEROL_SERU	B	10	20	YEARS	YEARS
CHOLESTEROL_SERU	B	20	30	YEARS	YEARS
CHOLESTEROL_SERU	B	30	55	YEARS	YEARS

The **Effective Date** and **Sex** fields are all components of the unique key for the range. In the example below the two ranges conflict because the unique key components overlap—that is, sex (because Both includes Men) and the date ranges overlap by one day. The ages do *not* overlap because the age units are different. Also, note that the units of the minimum and maximum can be different from each other, as well as different from the previous row.

Table 15-2 Invalid Sex and Effective Date Values

Lab Test Question	Sex	Minimum Age	Maximum Age	Effective Start Date	Effective End Date	Minimum Age Unit	Maximum Age Unit
ALBUMIN_SERUM	B	30	40	-	19980101	DAYS	DAYS
ALBUMIN_SERUM	M	32	38	19980101	-	MONTHS	YEARS

Lab Range Records without Effective Start or Effective End Dates

If you enter a lab range record and leave either the **Effective Start** or **Effective End Date** field empty, the Lab Ranges window leaves that field blank when you save the record. However, Oracle Clinical does store a date value in the database for that record. The presentation of this date value in SQL*Plus can be counter-intuitive, so this section describes how Oracle Clinical stores this information.

If you leave the **Effective Start Date** field empty, Oracle Clinical stores the date January 1, -4712 in the effective start date column for that record. This date is 1 Julian, which is the beginning of time for an Oracle database. When you query for the record, SQL*Plus removes

the negative sign from the year, displaying the values as 01-JAN-4712 or just 01-JAN-12. Despite the presentation, Oracle Clinical logically handles lab range records in this format as having no effective start date.

Similar logic applies to the **Effective End Date**. When you leave this field empty for a lab range record, Oracle Clinical stores the end date as August 15, 3501. This date is 3,000,000 Julian, and represents the end of time for an Oracle database. Depending on the date formats you use for SQL*Plus, the system might display lab ranges without effective end dates as having an end date of "15-AUG-3501."

Defining Lab Ranges Using the Maintain Lab Ranges Window

To define a lab range:

1. Open the Maintain Labs window (from the **Labs** menu, select **Labs**, then **Labs**) and query for the lab you want to modify.
2. Click the **Ranges** button. The Maintain Lab Ranges window opens.
3. For each lab range you want to define:
 - a. Query for the lab test Question for this lab range. When you select a Question, the window displays its domain and the Range Type (which is the source of the range, and is either LAB or TEXTBOOK).
 - b. Choose the gender to which this range applies. Values are Male (M), Female (F), or Both (B).
 - c. Enter the minimum and maximum age ranges. You can leave either end of the age range blank, if no minimum or maximum age applies for this range.
 - d. Enter the low and high values for this textbook range, and the Lab Unit code for the measurement you are specifying.
 - e. (Optional) Enter the dates during which the textbook range is in effect. You can leave one or both of these blank.
4. Save. Oracle Clinical saves these lab ranges for this lab definition.

Defining Lab Subsets

Oracle Clinical enables you to create (optional) subsets of existing lab ranges, which you can use for particular lab Questions where the normal range is affected by a predictable factor. For example, diabetes consistently influences the range of data considered normal for certain Questions.

To prepare for subset creation, when you define a lab Oracle Clinical creates it with a single default subset. When you define ranges for a lab, Oracle Clinical automatically associates them with the default subset as well. These are the only ranges that are available to other subsets for the lab.

Before creating subsets you must enter the values you plan to use for subset names in the LAB RANGE SUBSET CODE installation reference codelist. If you define no explicit subsets then the lab will have only one, *lab_name\$default*. The name assigned to the default is also specified in the LAB RANGE SUBSET CODE reference codelist.

You also define these lab range subsets in the Lab Range Subsets window. When you create a subset, you are required to select the Questions from a list of the Questions included in the default subset. The default subset is created automatically when the

lab is created. Subsets have the syntax `lab_name$subset_name` when a lab is referenced elsewhere in the system.

For more information, see:

- [Defining the Subset Name](#)
- [Defining the Subset Range Values](#)

Defining the Subset Name

In this section, you establish the subset you want to use for this lab. To define a Lab Range Subset:

1. Open the Maintain Labs window (from the **Labs** menu, select **Labs**, then **Labs**) and query for the lab you want to modify.
2. Click the **Subsets** button. The Maintain Lab Range Subsets window opens.
3. Click in an empty row, and choose a Subset Code from the list of values. The window populates a default name for the subset, which you can modify at any time.
4. Select the **Active?** box to use this subset in the lab range, or clear it to prevent its use at this time. You can change this setting at any time.
5. Save. Oracle Clinical saves the subset name for this lab. Proceed to [Defining the Subset Range Values](#).

Defining the Subset Range Values

In this section, you specify the subset values for a particular lab range. There are two ways to do this:

- You can enter the subset ranges manually. From the Lab Range Subsets window, click **Ranges**, then enter the ranges as you would for the default subset.
- You can copy a set of ranges from the default subset, then edit them for the new subset.

To copy ranges from the default layout:

1. In the Lab Range Subsets window, confirm that the subset that you want to use is selected, then click **Ranges**.
2. In the Ranges window, click **Copy Ranges from Default Subset**. The Multiple Questions Selection from the Default Subset window opens.
3. Choose one or more Questions to include, and click **OK**. This window closes, and the ranges window adds all of the default lab ranges you selected to the subset.
4. Edit each subset range to include the values you want.
5. Save. Oracle Clinical saves this subset for the lab.

Using Textbook Ranges

The List function in the pop-up window displays available copy groups. The **New Dates Start/End** field in the Use Textbook Ranges window substitutes for the original effective dates of each of the textbook ranges. If the textbook ranges have changed over time, the **Text Range Date** field lets you copy older textbook ranges.

You can copy a whole set of lab Questions, their preferred units of measure, and predefined company-standard textbook ranges (see [Defining Textbook Ranges](#)) by clicking the **Textbook**

Ranges button and entering the name of a Preferred Copy Group (see [Defining Preferred Copy Groups](#)).

 **Note:**

If you want to associate a textbook range for a unit with a particular Question rather than a whole set of Questions, from the **Labs** menu, select **Labs**, then **Labs**. Select the **Ranges** button. In the Ranges window that appears, enter the Question with which you want to associate a textbook range. Oracle Clinical populates the LOVs of the enterable fields, such as **Min** or **Low** under **Age** or **Values**, with textbook range values.

To use textbook ranges in a lab:

1. From the Maintain Labs window (from the **Labs** menu, select **Labs**, then **Labs**), choose the lab for which you want to use textbook ranges.
2. Click the **Use Textbook Ranges** button. The Use Text Book Ranges window opens.
3. Select a Preferred Copy Group to use. Oracle Clinical will copy this group's test Questions, ranges and units into this lab.
4. If you want to use the current version of the textbook ranges, leave the remaining fields empty and click **OK**.

If you want to use earlier versions of the textbook ranges, enter a date when this textbook range was current in the **Text Range Date** field. You can only use previous versions of textbook ranges if these ranges are still active.

5. Enter the dates over which the copied ranges will be effective. Dates must be in the YYYYMMDD format.
6. Click **OK** to close the Use Text Book Ranges window, then save.

Copying Lab Ranges

If you are creating a new lab definition for a lab that will perform a set of lab tests very similar to an existing lab, you can copy a whole set of lab ranges from another lab definition by pressing the **Copy Lab Ranges** button and choosing a valid lab to copy from. You can then edit the ranges, if necessary. You cannot edit them once the new lab has been used.

To copy lab ranges from another lab into the lab you have selected:

1. In the Maintain Labs window (from the **Labs** menu, select **Labs**, then **Labs**), choose the lab into which you want to copy lab ranges.
2. Click the **Copy Lab Ranges** button. The Copy Lab Ranges window opens.
3. Choose the lab from which you want to copy ranges, and click **Copy**.

The operation generates Alert messages, which list the number of ranges copied, number rejected, and which Questions were not copied because of a conflict. Lab ranges that copy successfully will appear in the selected lab with a Range Type of LAB and a Comment of COPIED.

4. Save.

Lab Assignment Criteria

Lab Assignment Criteria (LAC) link studies to labs. When you set up a study you define the labs that will process the physical patient samples collected during the trial. The lab test Question results are then associated with the lab in the database and compared to the lab ranges defined for that lab, and when the lab test results are entered during data entry, the name of the lab that did the testing is displayed.

If necessary, you can link different labs to different RDCIs, RDCMs, subsets, layouts, study sites, Investigators, or even visits or patients.

You can update or create new lab assignment criteria if circumstances change or if new sets of assignment criteria are required while the study is in progress.

Lab assignment criteria include:

- Start Date/End Date
- Study Site
- Investigator
- DCI Name
- DCM Name, Subset and Layout
- Patient
- Clinical Planned Event Name, Visit Number

It is not necessary to complete all fields. If you leave a field blank, the lab is linked to all values for that field; for example, all sites or all Investigators. In matching lab assignment criteria to RDCMs, the system always uses the most specific set of criteria.

Each set of lab assignment criteria in the study must have a unique combination of criteria values.

A few logical restrictions apply in completing some LAC fields:

- If you enter a DCM Subset Name, the DCM name is required.
- If you enter the DCM Layout Sequence, the DCM name and subset are required.
- If you enter a DCI Name in the LAC, the list of values for DCMs will contain only DCMs that belong to this DCI; otherwise, all DCMs are valid.
- If you complete the **Actual Event** field, the clinical planned event is restricted to the correct event.
- Clinical planned events must be valid for the selected study.

For more information, see:

- [Example of Lab Assignment Criteria Evolution](#)
- [Applying Lab Assignment Criteria](#)
- [Resolving Problems](#)

Example of Lab Assignment Criteria Evolution

The following table illustrates how lab assignment criteria may evolve. Each line represents the creation of a new LAC in the same study. Each line is annotated below the table with the events that prompted the definition of a new LAC.

S. No	Lab	Start/End Date	Site	Investigator	DCI	DCM	Patient	CPE	AE
1.	L1	-	-	VI	-	-	-	-	-
2.	L2	-	S2	-	-	-	-	-	-
3.	L3	19971202 19980115	-	VI	-	-	-	-	-
4.	L1	19971202 19980115	-	VI	LE	-	P1	-	-

1. The original LAC, created when the study began: Investigator V1 managed all patients and Lab L1 processed all lab tests.
2. A new site, S2, is added to the study. Lab L2 processes its lab tests.
3. The original Investigator, V1, is now working with a different lab, L3. Any RDCMs that come in between December 2, 1997 and January 15, 1998 are assigned to L3. If no new LAC is defined for the period following January 15, then assignments default back to the LAC on line 1, which has no date.
4. During the period when V1 is using a different lab, one of his patients, P1, continues to use the original lab, L1.

Applying Lab Assignment Criteria

In addition to storing the lab to which each lab test RDCM is assigned, Oracle Clinical stores the method by which the lab was assigned. These methods are hierarchical and you can use this hierarchy to force Oracle Clinical to reassign certain RDCMs. The following table describes the five methods in order of descending precedence.

Method	Description
USER	Explicitly assigned by user by choosing from the list of values for Lab in the Received DCMs window of the Maintain Lab Assignment Criteria window.
SYSTEM	Not currently used.
LOGIN	Lab name explicitly entered during data entry.
CRITERION	Assigned by running the Apply LAC job.
UNKNOWN	Lab is missing. Default setting.

To apply Lab Assignment Criteria to a study:

1. Navigate to **Labs**, and **Lab Assignment Criteria**. The Maintain Lab Assignment Criteria window opens.
2. Query for, then select, the study to which you want to apply LACs.
3. Click **Apply LAC**. The Apply LAC to Study window opens.

4. Enter a **Lab Assignment Type**. This selection forces a batch process to examine each lab RDCM for the study. Any that have a lab assignment type equal to or lower in precedence than the method you entered is re-evaluated and reassigned according to the current lab assignment criteria.
5. Enter a **Change Reason**. A list of values is available.
6. (Optional) Enter an **Audit Comment** about applying this LAC to this study.
7. (Optional) Select the **Process Locked RDCMs** box to include locked RDCMs as well. This box may be grayed out if you only have update privileges, because processing locked RDCMs requires Privileged Update. See [Granting Privileged Update to a User](#) for instructions on being granted Privileged Update.
8. Click **Apply**. Oracle Clinical applies the LAC to this study.

For more information, see:

- [Granting Privileged Update to a User](#)

Granting Privileged Update to a User

Administrators can grant you Privileged Update status in the Data Entry Configuration settings if you need to perform activities such as processing locked RDCMs.

To grant a user Privileged Update status:

1. From the **Admin** menu, select **Users**, then choose **Oracle Accounts**. The Maintain Oracle Accounts window opens.
2. Choose the user's account name, and select **Special**, and **DE Configs**. The Maintain User Configuration window opens.
3. For **Privileged Update**, choose **Enabled** from the list of values.
4. Save. Upon your next visit to the Maintain Lab Assignment Criteria window, you will be able to select the **Process Locked RDCMs** box.

Resolving Problems

Lab data may not link to ranges because an RDCM is not linked to a lab or because necessary patient information, such as age or gender, is missing. These problems—[Missing Lab References in the RDCMs](#) and [Missing Data in Patient Positions](#)—can be identified in windows reached through the buttons in the Maintain Lab Assignment Criteria window.

For more information, see:

- [Missing Lab References in the RDCMs](#)
- [Missing Data in Patient Positions](#)

Missing Lab References in the RDCMs

To identify missing lab references in the RDCMs:

1. Navigate to **Labs**, and **Lab Assignment Criteria**. The Maintain Lab Assignment Criteria window opens.
2. Choose the study that you want to investigate.
3. Click the **Received DCMS** button.

The Received DCMs for Study window opens; this window enables you to identify received DCMs (documents) that have either not been assigned to a lab or have been assigned incorrectly. Only RDCMs that have lab test Questions may be queried.

The **Lab Assignment Type** field reflects how the lab was assigned to the RDCM. If you assign a lab using this form, the system updates the Lab Assignment Type to USER. USER is hierarchically the highest method Oracle Clinical recognizes, so this lab assignment can only be changed in the future by explicit user intervention. If you make the **Lab** field blank here, the Lab Assignment Type is set to UNKNOWN. Each document can be associated with one and only one lab.

To query all RDCMs where the lab is blank:

1. Enter Query mode.
2. Enter LAB in the **Lab** field.
3. Execute the query.
4. Enter LAB IS NULL in the window.
5. Save.

Missing Data in Patient Positions

To examine patient positions where data is missing:

1. Navigate to **Labs**, and **Lab Assignment Criteria**. The Maintain Lab Assignment Criteria window opens.
2. Query for and select the study that you want to investigate.
3. Click the **View Patient Positions** button.

Oracle Clinical cannot compare lab test results to lab ranges if a patient's age and gender are not entered in the system. To identify patient positions with missing birth dates or gender, press the **View Patient Positions** button from the Maintain Lab Assignment Criteria form.

You can only query patient positions in this window; you cannot insert, modify, or delete patients.

Batch Loading Lab Ranges

Oracle Clinical accepts lab range data in batch mode. Batch loading data is more efficient and less prone to introducing new errors for large quantities of data than manually entering data. Loading of lab ranges is for new lab ranges only; you cannot perform updates of this data using the batch loader.

Put the input file into the directory specified in the LAB_ROOT setting of the OCL_STATE local reference codelist. If LAB_USERDIRS in the same codelist is set to Y, copy the file into the user-specific subdirectory for the user who will run the job. See the *Oracle Clinical Administrator's Guide* for more information.

This section explains the processes:

- [Loading Data](#)
- [Validating Data](#)
- [Transferring Data](#)

- [Executing the Process](#)
- [Running the Lab Procedures Re-execution Pre-Process Batch Job](#)

Loading Data

The goal of the data load stage is to verify that a value is present for each mandatory key column. There are no checks for the validity of the data at this stage; only that it exists.

The system loads the data into a temporary table, TEMP_RANGES. If there are existing rows for the lab being loaded, then it aborts and displays a message, "Another Batch Data Load in Progress for this lab <Lab Name>." The input to the process is the lab name, the full pathname to the data file, the domain name, and the delimiters used for the data, which are entered using a form described in detail in [Executing the Process](#).

After all validations are complete, the entire data file is loaded into TEMP_RANGES. If there are errors, the status code in the LAB_DATA_FILES table is updated to BAD LOAD, but rows without errors are inserted in TEMP_RANGES.

For more information, see:

- [File Format](#)
- [Example](#)

File Format

The data file you load must be in a standard format, including the following columns in this order: Lab Name, Lab Range Subset Code, lab test Question, Sex, Minimum Age, Maximum Age, Minimum Value, Maximum Value, Lab Unit Name, Effective Start Date, Effective End Date, Minimum Age Unit, and Maximum Age Unit.

You can use whatever delimiter you want in the file and specify it in the execution window.

Example

The delimiter here is a comma (,):

```
CENTRAL,,SODIUM,F,10,20,-5,10,u/ml,,,DAYS,DAYS
```

The two commas after the Lab Name and two commas after the Lab Unit Name mean that the Lab Range Subset, Effective Start Date and Effective End Date are not provided. Leading and trailing blanks must be stripped before insertion into the TEMP_RANGES table.

Validating Data

During preparation stage the system performs the following validations on the data loaded into the temporary table:

- Checks input data against existing database definitions for duplicate rows for:
 - Lab Name
 - Question Name
 - Lab Unit
- Performs lab range validations to check:

- Relative values of the age fields—maximum_age should be greater than the minimum_age
- Overlap in gender fields
- Date range; with the last row having an open Effective End Date
- minimum_value and maximum_value
 - * These parameters can be zero, a positive value, or a negative value.
 - * maximum_value should be greater than minimum_value, if both the values are not null.

After the error checking, the system produces a printable report of the results of the load and prepare stages, indicating the number of erroneous records and providing the PSUB job ID, which can be queried in the Batch Jobs window.

Transferring Data

Following checks of the TEMP_RANGES table for validation errors and duplicate rows, the permanent lab ranges database tables are populated from the rows in the TEMP_RANGES table. If no duplicate rows exist then all the rows from the TEMP_RANGES are inserted into the RANGES table. Duplicate rows are handled as follows:

- If you have the same lab range (combinations of key values, i.e. Lab, Question, Gender, minimum_age, maximum_age) and different minimum value or maximum value, update the record with the new minimum or maximum value, or both.
- If the audit has to be recorded in the range_comment column, then the values of previous values and the updated values can be shown in the comment.
- If the audit is not being recorded in the range_comment column, then the range_comment can also have a different value and can be updated.
- If you have the same key combinations (Lab, Question, Gender, minimum_age, maximum_age) and different Effective Start and End Dates, rearrange data according to the rules to accommodate the new effective dates.
- If your date ranges overlap in a record, rearrange all rows for that key set.

For more information, see:

- [Example](#)

Example

If the existing rows are as follows:

Question, Gender, Min.Age, Max.Age, Effective Start ,Effective End

```
QUESTION ,B ,10 ,20 ,19980110 ,19980209
QUESTION ,B ,10 ,20 ,19980310 ,19980409
```

adding the following new row, which comes in with the same combination of keys except for the minimum age and maximum age,

Question ,Gender, Min.Age, Max.Age, Effective Start, Effective End

```
QUESTION, B ,10 ,20, 19980210 ,19980309
```

results in the following final set of rows:

Question ,Gender ,Min.Age ,Max.Age ,Effective Start ,Effective End

QUESTION, B, 10 ,20, 19980110, 19980209

QUESTION ,B ,10, 19980210 ,19980309

QUESTION ,B ,10, 20, 19980310, 19980409

Executing the Process

You set parameters and execute batch data loads of lab range data from within Oracle Clinical. To execute a batch data load:

1. From the **Labs** menu, select **Lab Batch Data Load**, and **Lab Ranges Batch Data Load**.
2. Complete the fields as follows:

Run Error Check Only (Y/N)? specifies whether to run a preliminary error check on the file (Y) or execute the full batch load (N).

Lab Data File Name provides the full pathname to the data file. The file must be in the directory specified in the LAB_ROOT setting of the OCL_STATE local reference codelist. If LAB_USERDIRS in the same codelist is set to Y, copy the file into the user-specific subdirectory for the user is submitting the job. See the *Oracle Clinical Administrator's Guide* for more information.

Column Delimiter used in the Data File identifies the delimiter used in the file to separate data elements. There are no restrictions in choosing this delimiter.

Domain Name identifies the domain of the lab test Questions.

Default Effective Dates (Y/N)? specifies whether the program should set the effective dates (Y) or if you plan to specify them manually in Oracle Clinical (N).

3. Click **Submit Job**.

The program produces a detailed output file, providing a complete listing of the events that occurred during execution. If the process completes successfully, it places a new row in the RANGES table.

Running the Lab Procedures Re-execution Pre-Process Batch Job

If you change any part of a lab range definition that has been used against production data, you must immediately execute the PSUB job by selecting **Lab Procedures Re-Execution Pre-Process** from the **Lab** menu. If you do not run this job, your changes will not be applied to existing data.

Validation and Derivation Procedures

Validation and Derivation Procedures serve two different purposes. Validation Procedures compare multiple Question responses for the same patient for the purpose of ensuring that patient data is valid. Derivation Procedures use calculations to derive values from collected data. However, both types of Procedures are defined the same way, have the same internal structure, and are executed during batch validation.

Procedures operate on data for only one patient at a time. You cannot write a Procedure to compare the data of different patients. When a Procedure is executed during batch validation, the system processes only those patients whose data has changed (including having new data entered) since the last batch validation. The system runs every Procedure over all the data specified in each Procedure for the patient, not just the new or changed data.

 **Note:**

If you want to run a Procedure over all patients, you can run the Procedure manually; see [Executing a Single Procedure](#).

In addition, batch validation automatically runs a Procedure over specified data for all patients if you change the Procedure definition itself.

Batch validation is usually scheduled to run at regular Intervals, such as nightly. During batch validation, all Derivation Procedures are executed before all Validation Procedures, so that you can use the derived values in Validation Procedures. For further information on batch validation, see the Batch Validation section of the chapter "Using the Discrepancy Database" in *Oracle Clinical Conducting a Study*.

You can create an entirely new Procedure or, if your company has standard DCMs and Procedures, copy one or more to a new study and make any necessary modifications. You can revise a Procedure in an ongoing study and save its current discrepancies by creating a new version of it (see [Where to Start](#)). You can test a Procedure before making it active and running it against production data (see [Testing a Procedure](#)).

If you want to use the Patient Status feature, you must create a Procedure in each study to populate the Patient Statuses table. Packaged functions with the necessary logic are included with Oracle Clinical. See [Maintaining Patient Statuses](#).

For information about the Oracle Clinical tables and columns you may want to reference in Procedures, see the *Oracle Clinical Stable Interface Technical Reference Manual*. Contact Oracle Support to receive a copy of this PDF-format manual. There is no charge.

 **Note:**

In order to reference tables that are not already accessible to the Procedure compilation (including basic tables such as RESPONSES and PATIENT_POSITIONS) you must explicitly grant select privileges on the table to the RXC_PD account.

For more information, see:

- [About Validation Procedures](#)
- [About Derivation Procedures](#)
- [Overview of Defining a Procedure](#)
- [How Procedures Work Internally](#)
- [Special Processing](#)
- [Procedure Variables](#)
- [Using Lab Ranges](#)
- [Defining Validation and Derivation Procedures](#)
- [Testing and Activating Procedures](#)
- [Executing Procedures](#)
- [V3.1-Style and Pre-V3.1-Style Procedures](#)
- [Maintaining Patient Statuses](#)

About Validation Procedures

You can create Validation Procedures to clean and compare patient data, and to check for inconsistencies and other discrepancies among Question responses for a single patient. For example, you can compare a patient's dosage amount over several visits, or compare a patient's diastolic blood pressure to his systolic blood pressure at the same visit. You must write an expression that, if true, indicates that the collected data is discrepant (inconsistent). Oracle Clinical then creates a *discrepancy* in its Discrepancy Database, associated with the discrepant response(s), a comment, and any other response values you specify. Clinical personnel must then determine whether the discrepant response is medically significant or the result of a data entry or other error, and handle it accordingly (see "Using the Discrepancy Database" in *Oracle Clinical Conducting a Study*).

Because Validation Procedures compare multiple responses, the discrepancies they generate are called *multivariate* discrepancies. The Discrepancy Database also holds *univariate* discrepancies, which are automatically generated by the system when a Question response does not conform to the limits defined for the Question, such as length, data type, upperbound, or lowerbound (see "Using the Discrepancy Database" in *Oracle Clinical Conducting a Study*). When you create a Procedure, you can choose whether or not to run the Procedure on data that the system has already found to have a univariate discrepancy.

Validation Procedures can compare multiple responses from a single Received DCM or make complex comparisons of data from multiple RDCMs and visits. Validation

Procedures can include arithmetic functions and data manipulations, and can compare both collected and derived data. Several types of special processing are available, including aggregation, lag functions, and limiting the data processed by correlating data by response value, event, or qualifying value (see [Special Processing](#)). The system makes many key, response, and related values available to Procedures as variables (see [Procedure Variables](#)).

For more information, see:

- [Examples of Validation Procedures](#)

Examples of Validation Procedures

This section includes examples of Validation Procedures:

- [Simple Validation Procedure](#)
- [Validation Procedure with an Arithmetic Function](#)
- [More Complex Validation Procedure](#)

Simple Validation Procedure

Is a patient's diastolic blood pressure greater than or equal to his systolic blood pressure?

```
A.BP_DIASTOLIC >= A.BP_SYSTOLIC
```

The system tests whether the diastolic is greater than or equal to the systolic blood pressure, and creates a discrepancy if it is.

This Procedure references responses to a Question Group that includes the blood pressure-related Questions BP_DIASTOLIC and BP_SYSTOLIC. Within the Procedure, that Question Group has an alias of "A." The statement above is the expression in a Procedure detail.

Validation Procedure with an Arithmetic Function

Is the difference between a patient's systolic blood pressure and his diastolic blood pressure less than 20?

```
A.BP_SYSTOLIC - A.BP_DIASTOLIC < 20
```

The system subtracts the diastolic blood pressure from the systolic and then checks if the difference is less than 20; if it is, a discrepancy is created.

Because this Procedure references the same Questions as the example Procedure above, you may want to include both tests in a single Procedure as separate details. If both tests evaluate to True, the system creates two discrepancies against the same data.

More Complex Validation Procedure

This Validation Procedure compares the reported onset date and stop date for an adverse event. If the reported onset date of the adverse event is later than the reported stop date, the Validation Procedure creates a discrepancy.

In this example, the Procedure operates on a single event and a single DCM. The Questions are both in the same Procedure Question Group, which is the Question Group ADVERSE_EVENTS. The first Question is AE_ONSET_DATE, and the second is AE_STOP_DATE.

Instead of directly comparing the values of the date Questions, you can create two user variables to hold those values after you have converted them to Oracle date type so they can simply be subtracted. For example, create user variable *i* to define the variable `ONSET_DATE`, and *j* to define the variable `STOP_DATE`. You must also define pre-detail custom code to convert the date character values stored in Oracle Clinical to Oracle variables of type `DATE` (for example, convert June 30, 2005 to 20050630). The custom code can also check for invalid dates (such as June 31, 2005) and take care of a type `DATE` of length 6 (such as June 2005, or 200506, to which a mid-month `TO_DATE` of 15 might be consistently appended for a length 8-type `DATE` of 20050615).

Finally, the Validation Procedure tests `ONSET_DATE > STOP_DATE` (for example, 20050630 > 20050603) and creates a discrepancy if True. The user variables are used in the actual comparison, not the underlying raw data values. The purpose of the user variables and the pre-detail custom code is to create variables of type `DATE` so that this simple numerical comparison can be effected.

No system variables are used by this Procedure definition.

About Derivation Procedures

Derivation Procedures perform calculations on collected data to derive related values; for example, convert values to alternate units (such as pounds to kilos), or compute a value from other data (such as patient age from birth date and visit date). You specify the Question Groups, Questions and clinical planned events (CPEs; usually visits) to use as input and write a calculation that results in the response to a derived Question.

You must create a derived Question to receive the derived value. You define derived Questions in the Global Library the same way as other Questions, except that you must select their **Derived?** box. You then add them to a Question Group in a study DCM. Alternatively, you can mark a Question as derived in a study DCM Question Group, even if it is not defined as derived in the Global Library. You can also choose whether or not to display the derived Question in the Data Entry window (though it cannot be enterable).

You can include validation tests in a Derivation Procedure to test data before you perform calculations on it. Oracle Clinical reports any discrepancies it finds to the Discrepancy Database. See "Using the Discrepancy Database" in *Oracle Clinical Conducting a Study*.



Note:

You can test data values before derivation in the same procedure, but to test the derived value you must create a separate procedure.

You can specify the order in which Derivation Procedures are executed during batch validation in the same study, so that you can use the derived value in other Derivation Procedures.

For more information, see:

- [Restrictions on Derivation Procedures](#)
- [Examples of Derivation Procedures](#)

Restrictions on Derivation Procedures

The following restrictions apply to Derivation Procedures:

- Each derived Question is derived by one and only one Derivation Procedure, and each Derivation Procedure calculation detail derives one and only one derived Question. It is possible to have multiple calculation details in a single procedure, each one deriving a value for a different derived Question.
- You must denote the Question Group that contains the derived Question as the *primary reference* Question Group.

 **Note:**

The cursor for the primary reference Question Group is the outermost Procedure Question Group Cursor (see [How Procedures Work Internally](#)). The response values to the primary Question Group are therefore fetched before any other responses in the Procedure, and the derived value is derived once for each instance of that outermost Procedure Question Group Cursor, with the final value derived at the end of the Procedure (only the final derived value is stored). In other words, if the derivation involves multiple Question Groups/cursors, the value of the derived response can change until all of the inner cursors are through looping for a given derived outer cursor.

Since validation tests are performed in the innermost loop, use caution when testing the derived value in such procedures. You may prefer to use a separate validation procedure to validate derived values.

- You cannot create a DCM based entirely on derived Questions or populate a DCM entirely with derived data.
- You must create the derived Question before you create the Procedure that will derive its response (see [Questions and Question Groups](#)).

Examples of Derivation Procedures

Following are two examples of simple Derivation Procedure calculation expressions.

Temperature Unit Conversion Given a temperature measured in centigrade, calculate the equivalent Fahrenheit value:

```
(9/5*TEMPERATURE_C)+32
```

Weight Unit Conversion Given a weight measured in pounds, calculate the equivalent metric (kg) value:

```
A.WEIGHT_LBS/2.21
```

Overview of Defining a Procedure

To create a Procedure, you must do the following:

- Name and describe the Procedure and specify whether data from single or multiple events (visits) and DCMs are involved; see [Naming and Describing a Procedure](#) . You can also copy and modify an existing Procedure; see [Where to Start](#).
- Specify the Question Group(s) and Questions whose responses the Procedure compares; see [Defining Procedure Question Groups](#) and [Defining Procedure Questions](#).
- Define any variables necessary for the test logic; see [Procedure Variables](#).
- (Optional) Set up special processing such as correlation, aggregation, lags, and/or Where clause extensions or qualifying expressions; see [Special Processing](#).
- Write one or more validation tests or, in the case of Derivation Procedures, calculations; see [Defining Procedure Details](#).
- (Optional) Write custom code to be inserted at one or more fixed places in the generated Procedure PL/SQL code; see [Custom Code](#).
- Run Procedure generation. The system generates PL/SQL code based on your definitions and custom code. You can see the generated code and edit it if you have the necessary privileges, but all edits of generated code are lost if you run generation again. Custom code is not lost when you regenerate. [Generating and Compiling a Procedure](#) and [Editing Generated PL/SQL Code](#).

It usually makes sense to include all the tests you need to do for the same set of data in the same Validation Procedure. See [Structure of Procedure PL/SQL Programs](#) for further information.

How Procedures Work Internally

When you invoke generation, Oracle Clinical takes the Procedure definitions you create and generates PL/SQL code that retrieves patient records (Question responses and related data) for evaluation or computation one at a time. To create complex Procedures it is important to understand:

- Which records are retrieved, and the order of retrieval (see [Structure of Procedure PL/SQL Programs](#))
- How to modify the way records are retrieved to best suit your needs and to enhance performance (see [Special Processing](#))
- Given a certain order of record retrieval, how to best save any data that may be needed later in the Procedure—for example, to compare values across visits (see [User Variables](#), [Aggregate Functions](#), [Aggregate Lag Functions](#), and [Lag Checks](#)).
- Do not use special characters in your code, including messages. Special characters include: ! & () * + , - . / : < = > [] ^ |. For further information see the *Oracle Database PL/SQL Language Reference 11g Release 2 (11.2)*. Or, for the latest information, you can generate a list of all keywords and reserved words with the V\$RESERVED_WORDS view, described in the *Oracle Database Reference 11g Release 2 (11.2)*; see **User Documentation** in [Preface](#).

For more information, see:

- [Cursors](#)
- [Structure of Procedure PL/SQL Programs](#)
- [Viewing Generated PL/SQL Code](#)

Cursors

A *cursor* is a PL/SQL data structure that retrieves one record at a time, as specified in a SELECT statement. The system places each column value retrieved by the cursor into a variable. When you generate a Procedure, the system creates two kinds of cursors:

- [Patient Information Cursors](#)
- [Procedure Question Group Cursors](#)

Patient Information Cursors

The Patient Information Cursor becomes the outermost loop in the Procedure. The system generates a standard Select statement for the Patient Information Cursor so that it always fetches values for the same columns. The cursor retrieves patient records sequentially, putting the information into standard variables such as Clinical Study ID and Patient Position ID. See [Standard Variables](#) for a complete list.

Procedure Question Group Cursors

Procedure Question Group (Procedure QG) Cursors are nested inside the Patient Cursor. The system creates one Procedure QG Cursor for each Procedure Question Group, and generates a Select statement for it based on your Procedure Question Group definition (see [Defining Procedure Question Groups](#)). The Procedure QG Cursors retrieve key values, Question responses, and response-related information (entry timestamp, response ID, and exception value text).

The system puts the values of the key fields and response-related information into Question Group variables (see [Question Group Variables](#)) and puts Question response values into Question variables (see [Question Variables](#)).

Structure of Procedure PL/SQL Programs

For each patient, a Procedure fetches records for each nested cursor, running each expression for every set of records.

Each Procedure includes the following logical parts:

- Declaration of variables and cursors. The system converts your Procedure definitions to PL/SQL code to declare each of the following:
 - Variables. All the variables—user, Question, and system variables—that you have included in the Procedure. See [Procedure Variables](#).
 - Patient Cursor. The Patient Cursor is not based on your definitions, but is standard in every Procedure. See [Patient Information Cursors](#).
 - Procedure Question Group Cursors. Each alias (Procedure Question Group) defined in the Procedure is declared as a cursor, with aggregate Question Groups (if any) first, then the primary reference Question Group, and then all other Question Groups in the order they are displayed in the Question Groups window. See [Procedure Question Group Cursors](#).
- Patient Cursor. The patient cursor is the outermost loop in the Procedure. It fetches information from the Patient Positions table about the next patient to be processed, and populates standard system variables with the information.

- First...*n* Procedure QG Cursors. The system creates a cursor for each Question Group, and orders them as follows:
 1. Aggregate Question Groups (if any)
 2. The primary reference Question Group
 3. All other Question Groups, in the order they are displayed in the Question Groups window

The first Procedure QG Cursor is the loop immediately inside the patient loop, and each successive cursor forms the next inner loop.

For each Question you specify in the Procedure Question Group, each Procedure QG Cursor fetches response values collected for the current patient at all CPEs (unless otherwise specified in the Question Group Event Range or correlation by event) in the order you specify in the **Sort Order** fields for the Question Group (see [Sort Order](#)).

The cursor also fetches values for any other variables you specify in the Procedure (see [Procedure Question Group Cursors](#)).

- Detail. The system picks up your definitions from the Detail and Detail Variable windows to determine whether to run the expression on the current record.
- Expression. The system runs each expression, referencing values for Questions and other variables as specified in the expression.

For Derivation Procedures, the system populates derived Question with the derived value (see [About Derivation Procedures](#)). For Derivation and Validation Procedures, the system reports any discrepancies found, and associates each discrepancy with the variables and comments specified in the Procedure.

- The system closes each cursor in turn and starts processing the next patient.

For more information, see:

- [Example: Procedure with More Than One Procedure Question Group](#)
- [Example: Procedure Containing One Procedure Question Group](#)
- [Example: Procedure with Multiple DCMs](#)
- [Sort Order](#)
- [Sample Procedure Code, with Custom Code Locations](#)

Example: Procedure with More Than One Procedure Question Group

The structure can be represented as follows:

```

/* cursor for getting VITALS Production */
cursor V_CUR(
ORDER BY RDCM.VISIT_NUMBER ASC,RDCM.SUBEVENT_NUMBER ASC, RES.REPEAT_SN ASC;
V V_CUR%rowtype;
V$BEGIN_SEQNUM number := -99999;
V$END_SEQNUM number := 99999;
/* cursor for getting AE Production */
cursor A_CUR(
...
ORDER BY RDCM.VISIT_NUMBER ASC,RDCM.SUBEVENT_NUMBER ASC, RES.REPEAT_SN ASC;
A A_CUR%rowtype;
A$BEGIN_SEQNUM number := -99999;

```

```

A$END_SEQNUM number := 99999;
  /***** Main Begin *****/
loop <<fetch_next_patient>>
  /***** Post Patient *****/
  loop << fetch_V_cur>>
    /***** Post QG V *****/
  loop << fetch_A_cur>>
    /***** Post QG A *****/
  <<detail_1>>
  <<detail_2>>
  /***** Post Details *****/
    end loop; /* A cursor loop */
  end loop; /* V cursor loop */
end loop; /* patients cursor loop */
  /***** Main End *****/
  /* Populate discrepancy check table */
  /* Populate response check table with derived values*/
end;

```

Example: Procedure Containing One Procedure Question Group

A Procedure fetches information as follows when it includes only one Procedure Question Group, and therefore only one Procedure QG Cursor. In this example, the Procedure Question Group is called Physical Exam, and two visits have occurred so far. The sort order is set to the default: by Event and Subevent in ascending order (see [Sort Order](#)).

Retrieve Patient 1 information

Retrieve responses and key values for Visit 1 Physical Exam for Patient 1

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Physical Exam for Patient 1

For each detail, execute expression(s)

Retrieve Patient 2 information

Retrieve responses and key values for Visit 1 Physical Exam for Patient 2

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Physical Exam for Patient 2

For each detail, execute expression(s)

Example: Procedure with Multiple DCMs

In this example, the Procedure retrieves information from two DCM Question Groups, Physical Exam and Lab Tests, that are collected at the same visit. The sort order is set to the default: by Event and Subevent in ascending order (see [Sort Order](#)). Each Question Group has its own Procedure QG Cursor:

Retrieve Patient 1 information

Retrieve responses and key values for Visit 1 Physical Exam for Patient 1

Retrieve responses and key values for Visit 1 Lab Tests for Patient 1

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Lab Tests for Patient 1

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Physical Exam for Patient 1

Retrieve responses and key values for Visit 1 Lab Tests for Patient 1

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Lab Tests for Patient 1

For each detail, execute expression(s)

Retrieve Patient 2 information

Retrieve responses and key values for Visit 1 Physical Exam for Patient 2

Retrieve responses and key values for Visit 1 Lab Tests for Patient 2

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Lab Tests for Patient 2

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Physical Exam for Patient 2

Retrieve responses and key values for Visit 1 Lab Tests for Patient 2

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Lab Tests for Patient 2

For each detail, execute expression(s)

Sort Order

In the Question Groups window you can specify the order in which you want the system to retrieve records within the Procedure Question Group Cursor for each Question Group. Order matters primarily when you use lag variables or perform a similar function yourself by using user variables to hold information from record instance to instance. You may want to compare each visit with the previous visit by Visit and Subevent Order or by DCM Date and Time (which, particularly for subevents, could differ from the Subevent Order). There are two fields for this purpose: **Event** and **Extension**.

In the **Event** field, you can choose to order by either:

- Visit Number and Subevent Number
- DCM Date and Time

The default behavior is: Visit Number ascending order, Subevent Number ascending order.

Within the context you choose in the **Event** field, in the **Extension** field you can choose to order by either:

- Repeat Sequence Number (SN) alone (applicable only to repeating Question Groups; the setting has no effect if repeats are not collected)

- Qualifying Value followed by Repeat Sequence Number (applicable only to DCMs with qualifying values; not available in pre-V3.1-style Procedures)

The default value is Response Repeat Sequence Number ascending.

In both fields, you can choose any combination of ascending or descending order.

Sample Procedure Code, with Custom Code Locations

Following is more detailed sample Procedure code. The places where you can insert custom code are highlighted. See [Custom Code](#) for further information.

```
PROCEDURE main /*-----*/
(i_clinical_study_id      IN patient_positions.clinical_study_id%TYPE,
 i_clinical_study_version_id IN patient_positions.clinical_study_version_id%TYPE,
 i_data_modified_flag     IN patient_positions.data_modified_flag%TYPE,
 ...
 i_current_location      IN patient_positions.owning_location%TYPE,
 i_timer                 IN BOOLEAN := FALSE ) IS
 v_duplicate_discrepancy  BOOLEAN;

/* << Oracle Clinical puts DECLARATION custom code here >> */

BEGIN
  IF i_timer THEN timer.capture; END IF;

BEGIN
  ...
  std.v_code_location := 'Main Begin';

/* << Oracle Clinical puts MAIN-BEGIN custom code here >> */

  OPEN std.patients_cur;
  LOOP <<fetch_next_patient>>
    FETCH std.patients_cur INTO std.patients_rec;
    std.v_code_location := 'Post Patient';
    EXIT WHEN std.patients_cur%NOTFOUND;

/* <<Oracle Clinical puts POST-PATIENT custom code here >> */

    IF std.v_debug = 'Y'
      THEN DBMS_OUTPUT.PUT_LINE (' ');
      DBMS_OUTPUT.PUT_LINE ('--> Patient: '||std.patients_rec.patient);
    END IF;
  BEGIN
    OPEN b_cur(std.patients_rec.patient_position_id,b$begin_seqnum,b$end_ seqnum);
    std.v_code_location := 'Fetching B data';
    FETCH b_cur INTO b;
    IF b_cur%FOUND
      THEN b$has_data := 'Y';
      ELSE IF b$has_data = 'Y'
        THEN EXIT;
        ELSE b := null_b_record;
      END IF;
    END IF;
    std.v_code_location := 'Post QG B';

/* << Oracle Clinical puts POST-QG-B custom code here >> */
```

```

...   std.v_code_location := 'Pre Details';

/* << Oracle Clinical puts PRE-DETAILS custom code here >> */

<<detail_1>>
std.v_code_location := 'First prototype detail - 1 = 1 test';
IF (1 = 1)
    ...
END IF;
GOTO detail_2; /* NextDetail=detail_2 */
<<detail_2>>
std.v_code_location := 'Second prototype detail - 2 = 2 test';
IF (2 = 2)
    ...
END IF;
GOTO detail_3; /* NextDetail=detail_3 */
<<detail_3>>
std.v_code_location := 'Third prototype detail - 3 = 3 test';
IF (NOT(c.diasbp IS NULL OR c.sysbp IS NULL OR c$sysbp$11 IS NULL))
    AND (3 = 3)
    ...
END IF;
GOTO no_more_details; /* NextDetail=no_more_details, could omit */
<<no_more_details>>
NULL;
...
std.v_code_location := 'Post Details';

/* << Oracle Clinical puts POST-DETAILS custom code here >> */

...
    END LOOP; /* b_cur */
END;
END LOOP; /* Patients cursor */
CLOSE std.patients_cur;
IF i_timer THEN timer.show_elapsed; END IF;

std.v_code_location := 'Main End';

/* << Oracle Clinical puts MAIN-END custom code here >> */

EXCEPTION WHEN OTHERS THEN exception_handling(SQLERRM);

END main;

```

Viewing Generated PL/SQL Code

After you have created the necessary elements of a Procedure, from the Procedures w Edit window (which is visible only if you have the privileges necessary to edit the code) you can generate the code and view the result (both from the **Special** menu).

Note that V3.1-style Procedures are implemented using a static package, and the Patient Cursor declaration is contained in the package spec, so that you do not see it in the generated code. You do see the Question Group Cursor declarations, which are specific to the Procedure.

In addition, the generated code includes a test and production version of each cursor. The test version has a T appended to its name (A_CURT instead of A_CUR for Alias A) and is not shown in the examples in this documentation.

Special Processing

Oracle Clinical provides a user interface (UI) for defining several types of algorithms for processing your data. You enter information in the UI as specified in each section below. When you invoke Procedure generation, the system creates PL/SQL code based on your definitions, and inserts the code in the appropriate place in the generated Procedure. Special processing types include:

- [Correlation](#)
- [Aggregate Functions](#)
- [Aggregate Lag Functions](#)
- [Lag Checks](#)
- [Where Clause Extension](#)
- [Qualifying Expression](#)
- [Custom Code](#)

Correlation

By default, a given Procedure fetches data from all Questions at all visits for each Procedure Question Group you specify. You can use correlation to limit the data processed by inner Procedure Question Group Cursors to data that matches values in outer Procedure Question Group Cursors.

Correlation processing is driven by the Question Group in the outer loop of the two Question Groups (which is the one displayed further up in the Question Groups window) because the outer loop is processed before the inner loop, but you must define correlation in the definition of the inner loop (the Question Group displayed lower in the window). See [Procedure Question Group Cursors](#) and [Defining Procedure Question Groups](#).

As the Procedure processes the data in the outer loop, it processes only data in the inner loop that satisfy the correlation condition(s) you define. The *current record*, or record currently being processed, is a Question response of the Question Group displayed higher in the window, fetched in the outer loop. The Question Group whose data is limited by correlation is the one displayed lower in the window, for which you define correlation conditions, and whose cursor becomes the inner loop of the Procedure.

You can specify the same Question Group as the correlating Question Group (outer loop) for any number of Question Groups displayed lower in the Question Groups window.



Note:

You can use the **Create Placeholder** setting in combination with correlation to keep track of missing data. See [About the Question Groups Window](#), "Create Placeholder?".

You can use as many combinations of correlation types as you need; for example, correlate the same two Question Groups on both Event and Question in the Time Since Dose example.

You can define three types of correlation, as follows. In each example, Question Group A is displayed higher in the window (forming the outer loop) and Question Group B is displayed lower (forming the inner loop of the two).

For more information, see:

- [Correlation on Event](#)
- [Correlation by Qualifying Value](#)
- [Correlation by Question Value](#)
- [Restrictions](#)
- [Setting Up Correlation](#)

Correlation on Event

Use correlation on an event, or visit, to process data collected only for certain events relative to the Event Number of the current record (from Procedure Question Group A). Choose **Previous**, **Actual**, or **Next Event**.

For example, you commonly need to compare responses from different Question Groups only with other responses collected at the same visit. In that case, correlate on the **Actual Event**. Then, when Question Group B is defined as correlated with Question Group A on the **Actual Event**, the Procedure runs an expression that compares responses from the two Question Groups only when the response from Question Group B was collected at the same visit as the response from Question Group A. If correlation by Event was not specified, the Procedure would run the expression comparing Question Group A's response to Question Group B's response from every visit.

If you specify correlation on the **Previous** or **Next Event**, you specify it in the definition of Procedure Question Group B, but it is from the perspective of Question Group A. For example, if Question Group B is defined as correlating with Question Group A on Event (Next), the Procedure runs the expression only when the response from Question Group B was collected at the visit after the visit of the current response from Question Group A. In other words, if the response from Question Group A was collected during Visit 5, the Procedure runs the expression only for the response collected from Question Group B at Visit 6.

 **Note:**

To compare responses to the same Question for several visits, use a lag check. See [Lag Checks](#).

 **Note:**

In pre-V3.1-style Procedures you can correlate only with the Actual Event.

Example: Correlation on Event

This example shows the same situation as in [Example: Procedure with Multiple DCMs](#), except that the two Question Groups are correlated by Event. Therefore the Procedure processes the record from the Lab Tests Procedure Question Group only when it was collected at the same visit as the current Physical Exam record.

Retrieve Patient 1 information

Retrieve responses and key values for Visit 1 Physical Exam for Patient 1

Retrieve responses and key values for Visit 1 Lab Tests for Patient 1

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Lab Tests for Patient 1

(No expressions are executed because the visit number does not match.)

Retrieve responses and key values for Visit 2 Physical Exam for Patient 1

Retrieve responses and key values for Visit 1 Lab Tests for Patient 1

(No expressions are executed because the visit number does not match.)

Retrieve responses and key values for Visit 2 Lab Tests for Patient 1

For each detail, execute expression(s)

Retrieve Patient 2 information

Retrieve responses and key values for Visit 1 Physical Exam for Patient 2

Retrieve responses and key values for Visit 1 Lab Tests for Patient 2

For each detail, execute expression(s)

Retrieve responses and key values for Visit 2 Lab Tests for Patient 2

(No expressions are executed because the visit number does not match.)

Retrieve responses and key values for Visit 2 Physical Exam for Patient 2

Retrieve responses and key values for Visit 1 Lab Tests for Patient 2

(No expressions are executed because the visit number does not match.)

Retrieve responses and key values for Visit 2 Lab Tests for Patient 2

For each detail, execute expression(s)

Correlation by Qualifying Value

Use correlation by qualifying value to compare responses to Questions that belong to DCMs with Qualifying Questions, only when the value of the DCM Qualifying Question is the same. The actual qualifying value is not specified, so the system will run the expression over the specified responses whenever their RDCMs' qualifying values are the same.

For example, you might have two DCMs with a qualifying value of either Left or Right, one DCM to collect general information about either the left eye or the right eye, and the other to collect information about abnormalities in one eye or the other. Correlation by qualifying value ensures that left eye general information at each visit is compared only with left eye abnormality information, and right eye general information is compared only with right eye abnormality information.

For information on DCMs with qualifying values, see [Configuring Qualifying Questions](#).

 **Note:**

If you want to run the expression only when the RDCM has a particular qualifying value, you can use the Where Clause Extension to specify the value in the DCM key variable `qualifying_value` (see DCM Key Variables in [Standard Variables](#)).

Example: Correlation by Qualifying Value

This example shows a similar situation to [Example: Procedure with Multiple DCMs](#), except that both Question Groups belong to a DCM with a qualifying value, and are correlated by DCM qualifying value. Therefore the Procedure processes the record from the Lab Tests Procedure Question Group only when its DCM qualifying value is the same as the current Physical Exam record's DCM qualifying value.

Retrieve Patient 1 information

Retrieve responses and key values for V1 Physical Exam for Pat. 1 (DCMQV=Y)

Retrieve responses and key values for V1 Lab Tests for Pat. 1 (DCMQV=Y)

For each detail, execute expression(s)

Retrieve responses and key values for V1 Lab Tests for Pat. 1 (DCMQV=N)

(No expressions are executed because qualifying value does not match.)

Retrieve responses and key values for V2 Lab Tests for Pat. 1 (DCMQV=Y)

For each detail, execute expression(s)

Retrieve responses and key values for V2 Lab Tests for Pat. 1 (DCMQV=N)

(No expressions are executed because qualifying value does not match.)

Retrieve responses and key values for V2 Physical Exam for Pat. 1 (DCMQV=N)

Retrieve responses and key values for V1 Lab Tests for Pat. 1 (DCMQV=Y)

(No expressions are executed because qualifying value does not match.)

Retrieve responses and key values for V1 Lab Tests for Pat. 1 (DCMQV=N)

For each detail, execute expression(s)

Retrieve responses and key values for V2 Lab Tests for Pat. 1 (DCMQV=Y)

(No expressions are executed because qualifying value does not match.)

Retrieve responses and key values for V2 Lab Tests for Pat. 1 (DCMQV=N)

For each detail, execute expression(s)

Retrieve Patient 2 information--same processing as for Pat. 1 above

Correlation by Question Value

When the Procedure compares values from paired sets of Question responses (a Question response common to both the current Question Group and the correlated Question Group), use correlating Questions. If two Question Groups collect data for a common series of nominal times (such as Time Since Dose), you can use the response value of the Time Since Dose Question to allow comparison testing with other Question responses collected at the same nominal time in the correlated Question Group.

Example: Correlation by Question Value

The processing of Question Groups correlated on Question Value looks the same as in [Correlation by Qualifying Value](#). The difference is that the system compares a Question response instead of the DCM qualifying value before running the detail.

Restrictions

The following restrictions apply to correlation:

- You can correlate a Question Group only to Question Groups listed above it in the Procedure Question Groups window.
- Correlation and aggregation cannot coexist in a single Procedure.

Setting Up Correlation

To use correlation, do the following:

1. In the Procedure Question Groups window, in the **Correlate With Alias** field, enter the alias of the Question Group with which you want to correlate the highlighted Question Group. The alias must be for a Question Group listed higher in the window than the highlighted Question Group.
2. Enter a value for one or more of the following types of correlation:
 - Event. From the list of values, choose **Previous**, **Actual**, or **Next**. The system will run an expression on a Question response only against the correlated Question Group's response(s) that were collected at the previous, same, or next event, compared to the event of the current Question response.
 - Qualifying Question. Applicable only for Question Groups that belong to DCMs with qualifying values. If selected, the system will run an expression that compares Question responses from Question Groups correlated on Qualifying Value only when the value of both Question Groups' DCM's Qualifying Question is the same.
 - Correlating Question. Click the **Correlating Questions** button and enter the name of a Question from each of the correlated Question Groups (use the list of values). You can enter one or more pairs of Questions, one from each Question Group. The system will run the Procedure only when the values of each pair of Question responses are the same.

Aggregate Functions

Aggregate functions compute an aggregate value for response values for a particular Question for a patient. Aggregate functions available are:

- Maximum. The highest value.

- Minimum. The lowest value.
- Sum. A running total.
- Average. The average of all specified responses.
- Count. The number of specified responses collected.

Oracle Clinical creates a system variable for each function you specify and automatically populates the variable to hold the information generated by the function. By default, the system includes all values for the same Question for the same patient, but you can use a Where Clause Extension to filter the responses included in the aggregate function by the value of any of the keys available in that field (see [Where Clause Extension](#)).

Oracle Clinical aggregate functions are based on standard Oracle aggregate functions. Except for the Count function, none of them include null values in their calculations.

The system fetches one row per patient for an aggregate Question Group for all visits. You cannot use an aggregate function, for example, to compute an average of repeated Question values at a single visit (you can use an aggregate lag function for that purpose; see [Aggregate Lag Functions](#)).

You define a Question Group as aggregate by selecting its **Aggregate?** box, then clicking the **Questions** button, selecting one or more Questions, and specifying the aggregate function for each selected Question.

The system displays aggregate Question Groups before all other Question Groups, regardless of the order in which you define them. It fetches their values before other Question Groups' values, and executes details with aggregate functions before other details in a Procedure.

For more information, see:

- [Variable Names](#)
- [Restrictions](#)
- [Setting up Aggregate Functions](#)

Variable Names

The naming convention for the system variables is:

alias.question_name[occurrence#]\$suffix. The suffixes are: mn, mx, sm, av, and ct for the five functions available. The occurrence number is present only when there is more than one occurrence of the Question in the DCM Question Group.

Restrictions

The following restrictions apply to aggregate functions:

- A single Procedure cannot include both aggregate functions and correlation.
- Qualifying expressions cannot be applied to an Aggregation Question Group (a Question Group with its **Aggregate?** box selected).
- All Questions in an Aggregation Question Group must use at least one aggregate function.
- Questions that use aggregate functions cannot use aggregate lag functions (see [Aggregate Lag Functions](#)).

Setting up Aggregate Functions

To use aggregate functions, you must:

1. In the Procedure Question Group window, select the **Aggregate?** box for the Question Group that contains the Question(s) on whose responses you want to run an aggregate function.
2. In the Questions window for that Question Group, for each Question on whose responses you want to run an aggregate function, in the **Aggregate Functions** field enter each aggregate function you want to have performed. Use the list of values or enter more than one value by entering a value followed by a comma, followed by another value, and so on.
3. From the main Procedures window, click the **Details** button. The Details window opens.
4. Click the **Variables** button. The Variables window opens.
5. In a new row in the Variables column, press F9 to open the System Variables window.
6. Select all the aggregate variable(s) you need to use in the Procedure and click **OK**.
7. Select **Report?** for each variable whose value you want to be able to see in the Discrepancy Database and in DCFs.

You can use the variables in detail expressions and custom code by pressing F9 from the **Expression** or **Custom Code** field and selecting them from the list of values.

Aggregate Lag Functions

To compare responses to the same Question for the same patient over a set number of visits, and flag those that are inconsistent, use an aggregate lag function.

Aggregate lag functions operate over the last n responses to a particular Question collected for a particular patient. You set the Lag Depth n to the number of immediate past responses you want to include, and select the aggregate function you want to calculate for those responses.

For example, to obtain the average of a patient's heart rates at his last three visits, mark the Procedure Question HEART_RATE with a Lag Depth of 3 and enter AVG into the **Aggregate Lag Function** field for that Procedure Question.

In aggregate lag functions you have two choices that are unavailable in aggregate functions:

- You can choose to include the current value in the aggregate lag function or not. If you include the current value, the system calculates the aggregate lag function on the current value plus the most recent n values.
- You can choose to include null values in the aggregate lag function calculation or not.
 - If you do not include null values, the system calculates the aggregate lag function on the n most recent non-null values.
 - If you do include null values, the system calculates the aggregate lag function on the n most recent values, whether the values are null or non-null.

You can use aggregate lag functions on any numeric Question in any Question Group that does not have its **Aggregate?** box selected. The **Aggregate?** box applies only to aggregate functions (see [Aggregate Functions](#)).

Oracle Clinical creates a system variable for each function you specify and automatically populates the variable to hold the information generated by the function, as follows:

- **Maximum.** The highest of the last n values entered as a response to the Question for the patient.
- **Minimum.** The lowest of the last n values entered as a response to the Question for the patient.
- **Sum.** A total of the last n responses to the Question for the patient.
- **Average.** The average of the last n responses to the Question for the patient.
- **Count.** The number of the last n responses collected for the Question for the patient. In aggregate lag functions, Count will equal n except at the beginning of the study, when fewer than n responses have been collected for a Question for a patient.

For more information, see:

- [Variable Names](#)
- [Restrictions on Aggregate Lag Functions](#)
- [Using Aggregate Lag Functions](#)

Variable Names

Oracle Clinical creates two types of system variables for aggregate lag functions:

- **Lag Variable.** The system creates a lag variable to hold each value needed for the Lag Depth you specify. If you specify a Lag Depth of 3, the system creates three lag variables to hold the values of the three most recent Question responses. These variables follow the naming convention `alias.question_name[occurrence#]$.Ln`. The occurrence number is present only when there is more than one occurrence of the Question in the DCM Question Group.
- **Aggregate Variable.** The system creates a variable for each aggregate function you specify for the Question. The naming convention for the system variables is the same as for the aggregate function: `alias.question_name[occurrence#]$.suffix`. The suffixes are: `mn`, `mx`, `sm`, `av`, and `ct` for the five functions available. The occurrence number is present only when there is more than one occurrence of the Question in the DCM Question Group.

Restrictions on Aggregate Lag Functions

A Question Group cannot support both aggregate functions and aggregate lag functions. If you want to use an aggregate lag function, do not define the Question Group as an aggregate one (do not select its **Aggregate?** box).

Using Aggregate Lag Functions

Do not select the **Aggregate?** box for the Question Group. That flag is for the aggregate function. See [Aggregate Functions](#).

To use aggregate lag functions, you must:

1. In the Questions window, for each Question on whose responses you want to run an aggregate function, enter values in the following fields:
 - **Lag Depth.** Enter the number of responses whose values you want to include in the lag function's calculation. For example, if you enter "5" the system will include the last five responses collected, within the limits you specify in the fields **Include Null in Lag?** and **Include Current Event in Aggregate Lag?**.
 - **Include Null in Lag?** Select this box if you want to include null values in the lag function's calculations. The default value is deselected.
 - **Aggregate Lag Function.** Enter each aggregate function you want to use. Use the list of values or enter more than one value by entering a value followed by a comma, followed by another value, and so on.
 - **Include Current Event in Aggregate Lag?** If selected, the aggregate lag function includes the value of the current event in its calculations.
2. In the Details window, click the **Variables** button.
3. Press F9 to open the System Variables window.
4. Select the aggregate lag variables and click **OK**.
5. Select **Report?** for each variable whose value you want to be able to see in the Discrepancy Database and in DCFs.

You can use the variables in expressions and custom code.

Lag Checks

In a lag check, you write the logic to compare the most recent n response values to the same Question for the same patient, where n is the Lag Depth you specify. Oracle Clinical creates a system variable for each of the last n responses. You can use these variables in writing expressions and custom code.

For example, if you want a Procedure to check whether a value is increasing over time, you can compare the current value with the single most recent previous value of a Question by using a Lag Depth of 1.

You can choose to include null values in the lag check or not.

In the code you write, you can include the current value in any way you choose.

For more information, see:

- [Variable Names](#)
- [Setting Up Lag Checks](#)
- [Lag Check Examples](#)

Variable Names

The naming convention for the system variables is: *alias.question_nameoccurrence#\$Ln*.

The occurrence number is present only when there is more than one occurrence of the Question in the DCM Question Group.

Setting Up Lag Checks

Do not select the **Aggregate?** box for the Question Group that contains a Question on whose responses you want to run a lag check.

To create a lag check, you must:

1. In the Questions window, for each Question on whose responses you want to run a lag check, in the **Lag Depth** field enter the number of responses whose values you want to include in the lag check. For example, if you enter "5" the system will create system variables for the values of the last five responses to the Question for the patient.
2. In the Details window, click the **Variables** button.
3. Click F9 to open the System Variables window.
4. Select the lag variables and click **OK**.
5. Use the variables in custom code.



Note:

Lags are not reset to null after each visit, so you need to design Procedures to take that into account when testing the first n Questions in repeating Question Groups.

Lag Check Examples

If you have a Procedure Question Group called MEDS, alias M, and in it a Question named DOSE, when you specify a Lag Depth of 3, the system creates three lag variables: $M\$dose\$L1$, $M\$dose\$L2$, and $M\$dose\$L3$. In addition, you still have access to the current value of the Question M.DOSE. You could create a discrepancy if the dose did not increase from one visit/repeat combination to the next by making this statement in the test detail: $M\$DOSE\$L1 >= M.DOSE$. This evaluates to TRUE if the value from the previous visit/repeat combination is greater than or equal to the current visit/repeat, and therefore creates a discrepancy.

The first lag ($\$L1$) of the first repeat of a visit is null. In general, the n th lag of the first n values seen by the system are null.

Other lag check examples:

Populating Lag Variables

Suppose the data for a non-repeating Question Group on a given Question, DOSE, is as follows (note that here the sort order of the Question Group is ascending):

Visit	D.Dose
1	10
2	20
3	30
4	40

Visit	D.Dose
5	100

As the data is fetched in the Procedure, the variables are populated as follows:

Visit	D.DOSE	D\$DOSE\$L1	D\$DOSE\$L2	D\$DOSE\$L3
1	10	null	null	null
2	20	10	null	null
3	30	20	10	null
4	40	30	20	10
5	100	40	30	20

Lag Values for Data from a Repeating Question Group

The Lag Depth applies to previous repeats in the same visit and then previous visits, in descending order. For example, for the Question Group alias AE with three repeats, Question AMOUNT, the lag variables would be populated as follows with successive values 10, 20, 50, 40, and 60:

Visit	Repeat	AE.AMOUNT	AE\$AMOUNT\$L1	AE\$AMOUNT\$L2
1	1	10	null	null
1	2	20	10	null
1	3	50	20	10
2	1	40	50	20
2	2	60	40	50

Including Null in Lag

When **Include Null in Lag?** is selected, variables are populated with the null value when no value is collected:

Visit	AE.AMOUNT	AE\$AMOUNT\$L1	AE\$AMOUNT\$L2
1	10	null	null
2	20	10	null
3	null	20	10
4	40	null	20

Not Including Null in Lag

When **Include Null in Lag?** is deselected, the system populates the lag value with the previous non-null value:

Visit	AE.AMOUNT	AE\$AMOUNT\$L1	AE\$AMOUNT\$L2
1	10	null	null
2	20	10	null

Visit	AE.AMOUNT	AE\$AMOUNT\$L1	AE\$AMOUNT\$L2
3	null	20	10
4	40	20	10

Qualifying Expressions and Lags

Qualifying expressions and lags interact in that lag variables are allowed in qualifying expressions; but the expression is evaluated before the lag is calculated. For example, if the qualifying expression is Q1=10, and the data is originally:

Qualifying Expression	Value 1	Value 2	Value 3	Value 4
Q1	10	20	10	10
Q2	a	b	c	d

then the values seen in the test detail and qualifying expressions would be:

Pass	Test Details	Q1 Value	Q2 Value
1st Pass	Q1=10, Q2=a	Q1\$L1 is null	Q2\$L1 is null
2nd Pass	Q1=10, Q2=c	Q1\$L1 = 10 (not 20)	Q2\$L1 = a (not b)
3rd Pass	Q1=10, Q2=d	Q1\$L1 = 10	Q2\$L1 = c

Where Clause Extension

For each Procedure Question Group, you can write a PL/SQL Where clause statement (up to 200 characters) specifying DCM key value(s) and/or a Response Repeat Sequence Number as criteria to filter the records retrieved for processing by the Procedure. See DCM Key Variables in [Standard Variables](#) for a list of DCM keys. To filter on other criteria, use the **Qualifying Expression** field (see [Qualifying Expression](#)).

The Where clause extension is very efficient because the filter occurs before each Question Group fetch.



Note:

Do not use the Where clause extension in Derivation Procedures.

To define a Where clause extension:

1. In the Procedure Definitions window, highlight the Procedure and click the **Question Groups** button. The Question Groups window opens.
2. In the **Where Clause Extension** field, enter the Where clause.
 - Do not include the word "where." The system supplies it in the generated code.
 - Use the list of values to select variables to use as criteria in the statement.

3. Save.

Qualifying Expression

For each Procedure Question Group, you can write a PL/SQL statement (up to 2,000 characters) specifying variable values as criteria to filter the records processed by the Procedure.

The qualifying expression approach is less efficient than the Where clause because the filter occurs after data has been fetched. However, because the cursor evaluates each Question Group after the fetch, you can filter on any criteria.

In V3.1-style Procedures, the Where clause extension is more efficient because it filters before the fetch, but the criteria are limited to DCM key fields.



Note:

Do not use the **Qualifying Expression** field in Derivation Procedures.

To define a qualifying expression:

1. In the Procedure Definitions window, highlight the Procedure and click the **Question Groups** button. The Question Groups window opens.
2. In the **Qualifying Expression** field for the Question Group, enter the expression. Use the list of values to select Question variables to use as criteria in the statement. You can use other variables as well; for lists of all the different types of variables, see [Procedure Variables](#).



Note:

The list of values includes all the Questions defined for the Procedure Question Group, even if they have not been selected for use in this Procedure. If you select a Question here, you must also define it as part of the Procedure by selecting **Questions** under **Question Groups**.

3. Save.

Custom Code

You can add your own PL/SQL code at certain points in the generated PL/SQL Procedure code. An extended list of values for variables is available by pressing F9. From the list of values, discrete value group information is available. You can enter up to 4000 characters.

For example, use pre-Details custom code to perform procedural logic prior to the Detail, accumulate information for previously retrieved DCM or patient records, or retrieve information not available in the generated cursors. Or, in post-Details code you can make calculations in preparation for the next cursor fetch, or maintain indexes for array variables. You must define variables before referencing them in code (see [Defining User Variables](#) and [Using Variables in V3.1-Style Procedures](#)).

Oracle Clinical places your custom code in the generated PL/SQL code at the location you specify in the **Custom Code Location** menu. See [Sample Procedure Code, with Custom Code Locations](#) for the exact location of the custom code types.

In pre-V3.1-style Procedures your custom code choices are limited to:

- PRE-DETAILS, immediately before the Details block; previously called Pre-Code
- POST-DETAILS, immediately after the Details block; previously called Post-Code

In V3.1-style Procedures your choices include:

- PRE-DETAILS, immediately before the Details block; previously called Pre-Code
- POST-DETAILS, immediately after the Details block; previously called Post-Code
- DECLARATION — at the main Procedure declaration section
- MAIN-BEGIN — at the beginning of the main Procedure
- POST-PATIENT — immediately after the patient fetch
- POST-QG — after Question Group fetch and any placeholder logic
- MAIN-END — at the end of the main Procedure

**Note:**

In 3.1-style Procedures, you must explicitly allow for the possibility of pointing to either test or production database tables. See [Converting to the V3.1 Style](#).

Procedure Variables

There are several different kinds of variables available for use in Procedures. Some are automatically created by the system (see [Question Variables](#) and [System Variables](#)). You can create others as necessary (see [User Variables](#)).

To use a variable in a Procedure, you must first add it to the Procedure. See [Defining User Variables](#) and [Using Variables in V3.1-Style Procedures](#). You then add it to a Procedure Detail Expression or custom code by selecting it from the list of values in the appropriate field. To see the list of values, press F9.

When you use a variable in a detail expression, you must also define various settings for the variable. Click the **Variables** button and specify whether or not you want to report its value to the Discrepancy Database. For Question variables, you can also specify whether or not the system should run the Procedure if the response value for the Question is null or discrepant (see [Using Variables in V3.1-Style Procedures](#)).

For more information, see:

- [Question Variables](#)
- [Aggregate and Lab Attribute Variables](#)
- [System Variables](#)
- [User Variables](#)

Question Variables

Use Question variables to compare Question response values in detail expressions, custom code, or qualifying expressions.

The system automatically creates a Question variable for each Question you include for each Question Group in the Procedure, and populates it with the value for the current patient record during Procedure execution.

The cursor retrieves NULL if the conversion fails, or if no response for the Question can be found for the current DCM.

The system creates Question variables using the following naming convention:

alias.question_name

Aggregate and Lab Attribute Variables

When you define a Procedure as including aggregation processing, the system automatically creates the variable necessary to hold the aggregate value. Similarly, when you define a Question as lab-related, the system creates several lab attribute variables.

All these variables are displayed with Question variables in the list of values for detail variables.

For more information, see:

- [Aggregate Variables](#)
- [Lab Attribute Variables](#)

Aggregate Variables

The system creates aggregate variables using the following naming convention:

alias.question_name[occurrence#]\$suffix

The suffixes are: *mn*, *mx*, *sm*, *av*, and *ct* for the five functions available. The occurrence number is present only when there is more than one occurrence of the Question in the DCM Question Group.

Lab Attribute Variables

The system creates lab attribute variables using the following naming convention:

question_group_alias\$question_name[occurrence#]\$suffix

The occurrence number is present only when there is more than one occurrence of the Question in the DCM Question Group.

The four two-letter suffixes are:

- *hi* (high value) Contains the value at the high end of the range for the Question defined in Lab Ranges. The high value variable is data type Number (13,5).
- *lo* (low value) Contains the value at the low end of the range for the Question defined in Lab Ranges. The low value variable is data type Number (13,5).

- `un` (unit) Contains the name of the unit used to define the range in Lab Ranges. The unit variable is data type `Varchar2` (10).
- `nf` (Not Found) The system sets this flag to Y if no lab range is found for the age and sex of a patient. If the lab range is not gender- or age-specific, the patient's sex or age is not taken into account. The "not found" variable is data type `Varchar2` (1).

System Variables

There are several types of system variables: [Standard Variables](#), [Question Group Variables](#), and [DVG Variables](#). To see them, you must press F9 in either the **Detail Expression** field or the **Custom Code** field, which opens the System Variable window. This window displays any user or Question variables you have already created for use in the Procedure. There are buttons for DVG, QG, and Std variables below. When you click one of these buttons, you see the system variables of that type related to the highlighted user or Question variable. You can then select the system variables you want to use in that detail expression or custom code location.

For more information, see:

- [Standard Variables](#)
- [Question Group Variables](#)
- [DVG Variables](#)

Standard Variables

The system generates a fixed set of variables for each Procedure. The patient cursor fetches the values for the following patient information fields in the outermost loop of the generated Procedure:

- `patient_position_id` Internal ID of the patient.
- `clinical_study_id` Internal ID of the study.
- `reported_sex` Sex (M or F), from enrollment.
- `reported_birth_date` Birthdate, from enrollment.
- `patient` Patient number.
- `Site_id` Internal ID of site.
- `Investigator_id` Internal ID of Investigator.
- `early_termination_flag` *If checked, patient has left study ahead of schedule.
- `patient_enrollment_date` *Date the patient was allocated this patient position.
- `clinical_subject_id` Internal ID of clinical subject.
- `inclusion_exclusion_date` Date of decision to include or exclude the patient from the final analysis.
- `reported_death_date` !*Date of patient's death.
- `reported_patient_reference` *Code the patient was known by in a previous study; according to patient (not validated).
- `reported_initials` * Patient's initials, from enrollment.

- `reported_date_last_pregnancy` *Date of patients's last pregnancy, from enrollment.
- `first_screening_date` Patient's first screening date.
- `termination_date` *Date patient position was terminated.

An asterisk (*) indicates that the variable is available in 3.1-style Procedures only.

The naming convention for standard patient-related variables is `rxcpdstd.patients_rec.question name`; for example, the Question variable name for the Question `REPORTED_BIRTH_DATE` would be `rxcpdstd.patients_rec.reported_birth_date`.



Note:

Pre-V3.1-style Procedures use the prefix `patients_rec`, so that the variable name for the same Question would be `patients_rec.reported_birth_date`.

Question Group Variables

Three types of automatically generated variables appear when you click the **QG Vars** button from the System Variables window that you display by pressing F9 from the **Expressions** field in the Details window, or in the Custom Code window:

DCM Key Variables

The Procedure QG Cursor fetches DCM key (login) information for each Question Group as follows:

- `received_dcm_id` Internal ID of Received DCM
- `received_dcm_entry_ts` Timestamp of Received DCM
- `Investigator_id` Internal ID of Investigator
- `Site_id` Internal ID of site
- `dcm_id` Internal ID of DCM Definition
- `dcm_subset_sn` DCM Subset Number
- `dcm_date` DCM Date (YYYYMMDD)
- `dcm_time` DCM Time (if Time Collected)
- `actual_event_id` Internal ID of Actual Event
- `lab_id` Internal ID of Lab (if Lab Data)
- `qualifying_value` Value for Qualifying Question
- `subevent_number` Subevent (0 for Planned Visits)
- `clin_plan_eve_id` Internal ID for Planned Event
- `visit_number` Visit Number for Planned Event
- `repeat_sn` Repeat Sequence Number
- `lab.*` Name of the Lab

- `clin_planned_eve_name.*` Event (CPE) Name
- `lab_range_subset_num.*` Lab Range Subset Number

*V3.1-style Procedures only

Question Group Standard Variables

The system generates a fixed set of variables for each Procedure Question Group. They may not all be applicable to a particular Question Group. For example, there are lab-related variables regardless of whether the Question is lab-related or not.

Response-Related Variables

The DCM cursor creates Question variables to hold the actual response value for each Question you specify. However, it also fetches information about the response as Question Group variables, as follows:

- `resp_id` Internal Unique ID for the Response.
- `ent_ts` Response Timestamp.
- `exc_val` Response Exception Value Text; contains text entered during data entry, usually if there was a problem with the actual response, including alpha DVG codes.

Other Question Group Variables

In addition, the system generates the variable `alias$has_data` if you select the Procedure Question Group's **Create Placeholder?** box. The system sets the variable's value to Y if any data is collected for that Question Group. The default is N. You can use this variable to help track missing DCMs.

DVG Variables

Discrete Value Groups (DVGs) are assigned to Questions to define the acceptable responses to the Question. There are three types: Internal, Alpha, and Thesaurus. The system creates variables based on the values of internal and thesaurus DVGs. See [Creating and Using DVGs](#) for further information.

To see DVG variables, open the System Variables window by pressing F9 from the **Expressions** field in the Details window, or in the Custom Code window, select a Question variable based on a Question with a DVG, and click the **DVG** button. The **DVG** button is grayed out when the selected Question variable is based on a Question that does not have a DVG.

Internal DVG Variables - Internal DVG variables are not true variables, but a set of fixed values determined by the DVG, not the Question response. The system creates a variable for each value in the DVG. The variable is simply the short DVG value text surrounded by single quotation marks; for example, if the DVG associated with a Question is YES_NO and the short value for YES is Y, the corresponding variable is 'Y'.

Thesaurus DVG Variables - If you select the **Include Thesaurus Terms** box in the Procedure Questions window for a Question with a thesaurus DVG, the system creates a variable for each of the (up to) three columns specified in the DVG to provide supplementary information about the term (response) from the external decode table specified in the DVG definition. The naming convention is:

`alias$question_nameoccurrence#$suffix` where the suffix corresponds to the Short

Label defined for each column. The occurrence number is present only when there is more than one occurrence of the Question in the DCM Question Group.

See [Thesaurus DVGs](#) for further information.

User Variables

You can create variables as necessary for use in internal processing and for reporting values you want to see associated with the discrepancy in the Discrepancy Database and in discrepancy reports.

You reach the User Variables window by clicking the **User Variables** button in the main Procedure Definitions window.

You can use user variables in **Detail Expressions** and **Custom Code**. From either field, press F9 to see the list of values and select a variable.

When you use a user variable in a detail expression, click the **Variables** button and specify whether or not you want to report its value to the Discrepancy Database.

Using Lab Ranges

You set lab ranges to define medically appropriate values for Question responses. A Question's lab range may vary according to a patient's age or sex. You can create a Procedure to check if a response falls into the range defined for its Question, creating a discrepancy if it does not. To reference lab ranges in a Procedure:

- In the Procedure Definitions window, select the **Lab Range Dependent?** box. If you select this box and then run the PSUB job **Lab Procedures Re-execution Pre-process** (under the **Labs** navigation path) after changes are made to lab ranges or lab range subsets, any Procedures that reference Questions with changed lab ranges will be re-executed for all patients during the next batch validation. This functionality is available in V3.1-style Procedures only. See **Lab Procedures Re-execution Pre-process Job Limitations** below.
 - If responses are no longer discrepant, according to the new range definition, associated discrepancies are obsoleted.
 - If responses are now discrepant, the system generates new discrepancies.
- In the Procedure Questions window, select the **Include Lab Range?** box for a Question that is defined with a Question type of Lab Test .

When a lab-dependent Procedure is executed, the system executes an internal function that brings lab range information into four system variables (High, Low, Unit, and Not Found) for each record fetched. See [Aggregate and Lab Attribute Variables](#).

 **Note:**

The function requires a value for DCM_DATE. If the lab range is date- or age-dependent, the function uses the entered DCM date to identify the correct range for the patient's data. Even if the lab range is not date- or age-dependent the DCM date is required.

You can ensure that a DCM_DATE value is available by enabling the Data Entry Configuration setting **DCI and DCM Date Required**, and either checking **Use DCI Date?** for the DCI Module containing the lab Questions or entering the DCM date during data entry.

- In a Procedure detail expression, reference one or more of the lab-related system variables.

For example:

You could define a nonaggregate Procedure Question Group PHARMACOKINETIC, alias PK, with a Procedure Question named HEMOGLOBIN. You can check the high and low ranges for hemoglobin, given the patient's age and sex, using the variables PK\$hemoglobin\$hi and PK\$hemoglobin\$lo. These variables are populated from the values in the Lab Reference Range system.

Different users may enter responses to lab range Questions in different units, and the units associated with the high and low values in the Lab Ranges system may not match the user-entered units. For this reason, it is important that Procedure designers not write Procedures that depend on values being expressed in the same units. Oracle Clinical creates a system variable to contain the unit of the ranges given by the Lab Reference Range system, for example:

PK\$hemoglobin\$un

No unit conversion (normalization) facility in Procedures exists, but you can convert and store values with custom code and user variables. You can write Procedures that compare values only when the units match. For example:

```
PK.hemoglobin > PK$hemoglobin$hi * 1.1 and PK.hemo_unit = PK$hemoglobin$un
```

The expression above creates a discrepancy when the hemoglobin count is higher than the upper lab range value by 10%, only if it is expressed in the same units.

In this example, you must define the Question Hemo_unit in the PK Question Group, to collect the unit in which the response is entered. You must also select the Question variable based on Hemo_unit in the detail expression.

Lab Procedures Re-execution Pre-process Job Limitations

The Lab Range Re-execution Pre-process job does not obsolete discrepancies and create new ones in the following circumstances:

- When changes are made to RDCM keys, except for the keys: Patient, Clinical Planned Event, and Subevent Number.
- When the Lab Range is changed in such a way that it no longer applies to a patient record. For example, if the Effective End Date is changed to a date before the patient data was collected, the job does not process the record and does not make the discrepancy obsolete.
- When the Question tested by the Procedure is removed from the lab definition.

Defining Validation and Derivation Procedures

This section gives explicit instructions on how to define Validation and Derivation Procedures. It includes the following topics:

- [Where to Start](#)
- [Procedure Definition Sequence](#)
- [Naming and Describing a Procedure](#)
- [Defining Procedure Question Groups](#)
- [Defining Procedure Questions](#)
- [Defining Correlating Questions for Procedure Question Groups](#)
- [Defining User Variables](#)
- [Defining Procedure Details](#)
- [Using System Variables in Pre-v3.1-Style Procedures](#)
- [Using Variables in V3.1-Style Procedures](#)
- [Adding Custom Code](#)

Where to Start

You define a Procedure following one of three basic approaches:

- [Creating an Entirely New Procedure](#)
- [Creating a New Version of a Procedure](#)
- [Modifying an Existing Procedure](#)

Creating an Entirely New Procedure

Define a new Procedure if your company is a new Oracle Clinical user, or if you are creating a Procedure for a new DCM. See [Naming and Describing a Procedure](#) .

Always create new Procedures in the study, not the Global Library. You cannot generate or test procedures in the Global Library. To add a procedure to the Global Library, you must define and test it within a study (via the Definition subsystem), activate it, and then copy it into the Global Library.

From the **Definition** menu, select **Validation Procedures**, and then **Procedures**, or from **Definition**, choose **Derivation Procedures**, and **Procedures**.

Creating a New Version of a Procedure

Use a new version for modest changes in functionality with the option to save discrepancies created by the original version. The new version replaces the current Procedure, which is saved with a status of Retired. Select **Create New Vers** from the **Special** menu in the Procedure Definitions window.

Modifying an Existing Procedure

From either the Global Library or another study, copy, then modify, an existing Procedure that references the same or functionally equivalent Questions and DCMs. Select either **Copy Proc-Study** or **Copy Proc-GLIB** from the **Special** menu in the Procedure Definitions window, then modify as necessary. To view existing Procedures in detail, from the **Glib** menu, select **DCIs DCMs Procedures**, and choose **Qry Validation/Derivation Procedures**, or from **Definition**, select **Validation (or Derivation) Procedures**, then **Procedures**, then choose a study. The following restrictions apply:

- You can copy only Active Procedures.
- The Data Collection Modules, Questions Groups, and Questions referenced in the Procedure (or their functional equivalents) must exist in the study to which you copy the Procedure; otherwise the copy action is terminated with an error message.
- You can copy only Procedures with domains in the current study domain searchlist.

Procedure Definition Sequence

To define a Procedure you must perform the following tasks, the first four in the order given:

1. [Naming and Describing a Procedure](#)
2. [Defining Procedure Question Groups](#)
3. [Defining Procedure Questions](#)
4. [Defining Correlating Questions for Procedure Question Groups](#) (optional)

The following tasks have no required definition sequence, though you must define variables before you can reference them in Details:

- [Defining User Variables](#) (optional)
- [Defining Procedure Details](#)
- [Using System Variables in Pre-v3.1-Style Procedures](#) (optional)
- [Using Variables in V3.1-Style Procedures](#)
- [Adding Custom Code](#) (optional)

Naming and Describing a Procedure

To reach the Procedure Definitions window, from the **Definition** menu, select **Validation (or Derivation) Procedures**, then choose one of the options listed in the table below, according to your access privileges.

For more information, see:

- [Access Privileges](#)
- [Defining a Procedure](#)

Access Privileges

The navigation option(s) you see under **Definition** are determined by your access privileges. You can perform the following tasks through these menu options:

Menu Option	View and Define Procedures	Set Procedures to Active	Edit Generated PL/SQL Code
Procedures	Allowed	Allowed	Not allowed
Prov Procedures	Allowed	Not allowed	Not allowed
Procedures w Edit	Allowed	Allowed	Allowed

Defining a Procedure

To define the Procedure, enter the information requested, including:

Name

Each Procedure Name must be unique within the clinical study.

Domain

Display only. The system enters the name of the current Oracle Clinical domain.

Version

A display-only number automatically assigned to each new version of a Procedure after its creation, which is version 0. The status of a new version of a Procedure defaults to Provisional. (You can revise a Procedure in an ongoing study and save its current discrepancies by creating a new version of it—select **Create New Vers** from the **Special** menu in the Procedure Definitions window.)

Status

There are three possible statuses: Provisional (P), Active (A), and Retired (R). You can modify and test Procedures only when they are set to Provisional. See [Assigning Procedure Status](#).

Note:

If you set an Active Procedure's status back to Provisional, all the discrepancies already created by the Procedure are hard-deleted during the next batch validation run. To avoid this, create a new version of the Procedure from the Special menu.

Sort Order

(Derivation Procedures only) Determines the order in which batch validation executes this Derivation Procedure relative to other Derivation Procedures in the same study.

Tip: If the execution order makes no difference, give the Procedure a Sort Order of 100, for example. Any number of Procedures can have the same Sort Order Number. Give a lower Sort Order Number to Procedures that generate a derived value that is used by other Validation or Derivation Procedures.

 **Note:**

During batch validation, the system executes all Derivation Procedures before all Validation Procedures for a particular study. Validation Procedures are executed in alphabetical order by Procedure Name. Derivation Procedures are executed by Sort Order Number. If multiple Derivation Procedures have the same Sort Order Number, the system executes those Procedures in alphabetical order by Procedure Name.

Execution Context

The default setting, Off-line, is appropriate for Procedures you want to execute within the context of Oracle Clinical, either as part of batch validation or individually.

The other two settings are for use with the Remote Data Capture (RDC) option for Oracle Clinical, or with your own DCAPI application. They enable you to run a Procedure on data soon after it is entered into the system, without waiting for batch validation to run:

- On-line/DCM. Procedures with this setting are run when the RDC user saves work for a particular patient, if the Procedure's primary reference Question Group belongs to the DCM the user has just entered. Any discrepancies generated by the Procedure become visible to the user. Use this setting only if the Procedure runs on responses from only one DCM.
- On-line. Use this setting if the Procedure compares more than one DCM, or if you prefer not to have the Procedure run automatically on save in RDC.

The **Validate Patient** and **Validate Site** options in RDC's **Validate** menu execute both On-line and On-line/DCM Procedures for a patient or a site, respectively. For information on DCAPI implementation, see the *Oracle Clinical Application Programming Interface Guide* manual.

Oracle Clinical batch validation runs all Procedures, regardless of their Execution Context setting.

 **Note:**

If you make a change to a Procedure, you must run batch validation to run the Procedure against all patients, including patients with no data changes. When On-line and On-line/DCM Procedures are executed from outside Oracle Clinical, Procedures are run only against patients with changed data, even if the Procedure itself has been modified.

Category

Choose the combination of one or multiple events (visits) and one or multiple DCMs that describes the data compared in this Procedure.

The `Thes_Derivation` option is for use with thesaurus DVGs. It has nothing to do with the current Thesaurus Management System (TMS).

Ver 3.1 Style?

Select the **Ver 3.1 Style?** box to get the best performance and functionality. See [V3.1-Style and Pre-V3.1-Style Procedures](#).

Lab Range Dependent?

Select this box if this Procedure references a Question that is associated with a lab range. Oracle Clinical then creates lab range-related system variables for use in the Procedure. In addition, if you change the definition of a lab range and then run the Lab Procedures Re-execution Pre-Process PSUB job (under **Labs**), the next batch validation executes all Procedures that both

- have this box selected
- reference Questions with changed lab ranges

for all patients, even if the patients' data has not changed.

Defining Procedure Question Groups

To reach the Question Groups window, click the **Q-Grps** button in the Procedure Definitions window. In this window you list the DCMs and Question Groups that the Procedure will reference and define how the Procedure will process them.

For more information, see:

- [Performance Issues](#)
- [About the Question Groups Window](#)

Performance Issues

The choices you make in this window have important implications for the efficiency of the Procedures you create. This section gives tips on how to design your Procedure for optimal performance. For more detailed information, see [How Procedures Work Internally](#).

Each cursor, or Question Group Alias, is nested inside those declared before it. So for each A record fetched, the B cursor must be opened and closed. To minimize need for cursors and maximize performance, use these design strategies:

- Use correlation whenever possible (See Correlate With, [About the Question Groups Window](#)).

Note:

You can correlate a Question Group only to Question Groups listed above it in the Question Groups window, so order Question Groups accordingly.

- Use qualifying expressions and Where clause extensions whenever possible in Validation Procedures. You cannot use them in Derivation Procedures. See [Qualifying Expression](#) and [Where Clause Extension](#).
- If you need two types of aggregate functions—for example, sum and count—from the same Question Group, use only one cursor. You can specify as many aggregates as you need using one alias for all the Questions in that Procedure Question Group.
- Use **Event Range** and **First/Last Event Only** whenever possible.

- Bear in mind the fact that the Procedure QG Cursor fetches aggregate Question Groups first, followed by the primary reference Question Group. The cursor then fetches the remaining Question Groups in the order they appear in the window, which is determined by the order in which you define them.

About the Question Groups Window

Be sure to read [Performance Issues](#) before proceeding.

In order to use a Question response value in a Validation or Derivation Procedure, you must define the Question and its DCM Question Group as part of the Procedure. In the Question Group window you can also define special processing for the Question Group.

When you include a DCM Question Group in a Procedure, you create a reference to it. A single DCM Question Group can be referenced by several Procedure Question Groups in the same Procedure. For example, to compute an average baseline value for a group of DCM Questions and compare baseline values to the latest values at each subsequent visit, you need to create two Procedure Question Groups for that DCM Question Group: one an aggregate Question Group to compute average baseline value; one to compare each subsequent value to the average baseline value.

The window includes the following fields:

Alias

A short name that uniquely identifies the Question Group and its Questions in Procedure details, correlated Question Groups, and some system variables. The name may be up to 4 characters and may not begin or end with a number. You can change this field only for Provisional Procedures. Changes automatically cascade to all uses of the alias throughout the Procedure.

Aggregate?

Select this box to use the Aggregate function to keep track of the minimum, maximum, sum, count, or average of a given Question response over all patients and visits; see [Aggregate Functions](#). If you want to track an aggregate value for only the last n visits, do not select this box and define an aggregate lag function for the Question in the Questions window.

Primary Reference

If selected, this Procedure Question Group is the primary reference Question Group. Any discrepancies found by the Procedure are associated with this Procedure Question Group in the Discrepancy Database. If this is a Derivation Procedure, the target derived Question must be part of the primary reference Question Group. The primary reference Question Group is fetched after any aggregate Question Groups but before all others.

First in the Event Range

The first visit (clinical planned event) you want the Procedure to process.

Last in the Event Range

The final visit (clinical planned event) you want the Procedure to process.

Sort Order Event

Controls the sort order for responses to the Questions in this Procedure Question Group, based on Event and Subevent or DCM Date and Time. You must choose an ascending order you plan to enter a value in the **First/Last Event Only** field. Use the list of values to select the sort order type. See [Sort Order](#).

Sort Order Extension

From the list of values, specify the innermost sort criteria for responses to the Questions in this Procedure Question Group. Default: REPEAT_SN ASC, or, where the corresponding DCM has a Qualifying Question, QUALIFYING_VALUE ASC, REPEAT_SN ASC.

First/Last Event Only

Specifies that the Procedure will process only the first or only the last event it finds for each patient.



Note:

Do not use this field in Derivation Procedures.

Single Repeat Only?

In V3.1-style Procedures, you can specify that you want to fetch only the first occurrence of a repeating Question Group.

Create Place Holder?

The Procedure creates a null DCM record when data the Procedure needs to run an expression is missing (equivalent to a SQL outer join). Since a particular Question response might be null because the data entry operator did not fill it in, rather than because the entire RDCM is missing, refine your code by referencing the Has_Data variable that Oracle Clinical creates for each Procedure Question Group with **Create Place Holder?** selected. Its default value is N. If data is entered for any Question in the RDCM, the system changes the value of the Has_Data variable to Y. Reference it as *Alias\$Has_Data*.

You can use this setting to create a discrepancy if a Question response is missing, and to detect missing RDCMs.



Note:

Do not use this field in Derivation Procedures.

Correlate With

By default, Oracle Clinical fetches all data for patients with changed data, and runs each Procedure on all Question responses at all visits for the RDCM Question Group Questions specified in the Procedure definition. You can avoid unnecessary processing by setting up correlation where appropriate. See [Correlation](#).



Note:

You cannot have both correlation and aggregation defined in the same Procedure.

To use correlation, you must specify an Alias—a Question Group with which to correlate this Question Group. The correlating alias must be displayed higher in this window than the current Question Group. You can then choose **Event**, **Qualifying Question**, or **Correlating Question**, or any combination of the three. See [Correlation](#).

Where Clause Extension

A PL/SQL Where clause statement (up to 200 characters) that limits the retrieval of records for processing by the Procedure by filtering on the value of DCM key fields before each Question Group fetch. Use the List function to obtain a list of DCM key fields to use as criteria. To filter on other criteria, use the **Qualifying Expression** field. See [Where Clause Extension](#).



Note:

Do not use this field in Derivation Procedures.

Qualifying Expression

A PL/SQL statement that limits the retrieval of records for processing by the Procedure. Use the List function to obtain a list of available QG Questions. You can filter on any criteria because the cursor evaluates each Question Group after the fetch. See [Qualifying Expression](#).



Note:

- Do not use this field in Derivation Procedures.
- If you use a Question here, you must also define it as part of the Procedure by selecting **Questions** under **Question Groups**.

Custom Code

You can add code to be executed after the system fetches the Procedure Question Group you are defining by clicking the **Custom Code** button and entering the code. See [Adding Custom Code](#).

Defining Procedure Questions

Reach this window by selecting **Question Groups** under **Procedure Definition**. In the Procedure Question Groups window, highlight a Question Group and click **Questions**.

List the DCM Questions you want to reference in the current Procedure Question Group and specify special processing, such as referencing lab ranges, as needed. When you define the Detail, or test, for the Procedure, you can use Procedure Questions' values as variables of type QUESTION using the naming convention *alias.question_name*.

Information on each field in the window follows:

Question Name - Use the list to define the DCM Questions that you want to reference in this Procedure Question Group.

Derived Questions that are the target of the Procedure have the following constraints:

- They must be in the primary reference Procedure Question Group.
- They cannot occur as the target of any other Detail in any other Procedure.
- You must include at least one non-derived Question in a Procedure Question Group.

Occurrence Number - Select the occurrence sequence number of the Question in the DCM Question Group. The list displays valid numbers for each available Question.

DCM Derived Question? - If the Question has been defined as a derived Question, this field is selected.

Report? - For pre-V3.1-style Procedures only. Select this box if you want this Question response included in discrepancy reports. (For V3.1-style Procedures see [Using Variables in V3.1-Style Procedures](#).)

Aggregate Function - The Aggregate function allows you to keep track of the minimum, maximum, sum, count, or average of a given Question response over all patients and visits. (You cannot define a Question as aggregate unless its Question Group's **Aggregate?** box is selected in the Procedure Question Groups window.) See [Aggregate Functions](#).

Include Lab Range? - If this Question is defined as a lab test Question, select this field. See [Using Lab Ranges](#).

 **Note:**

When you change a lab range (from the **Labs** menu, choose **Labs**, then **Labs**, then choose **Ranges**) you must run the Lab Procedures Re-execution Pre-Process PSUB job (under **Labs**). Oracle Clinical will then re-execute all Procedures that reference that changed lab range for all patients during the next batch validation.

Include Thesaurus Terms? - Select if the Question is defined as having a thesaurus DVG. The system generates up to three system variables to receive the values defined in the thesaurus DVG definition. See [DVG Variables](#), "Thesaurus DVG Variables".

Lag Depth - Specifies the number of values of a single Procedure Question for a single patient (immediately before the current value) to make available for use in detail expressions and custom code. Oracle Clinical creates a system variable to hold each of the last *n* values for a Question, where *n* is the Lag Depth you specify. Used in aggregate lag functions and in lag checks. See [Aggregate Lag Functions](#) and [Lag Checks](#).

Include Null in Lag? - If this field is not selected, null values are not included for lag variables, so a previous non-null value is retained until a new non-null value is available. Controls whether re-initialization occurs for each process loop, or only LAG_FLAG when the lag variable is not null. See [Aggregate Lag Functions](#) and [Lag Checks](#).

Aggregate Lag Function? - If you want to calculate an aggregate value over only the last n values, use the aggregate lag function. See [Aggregate Lag Functions](#).

Include Current Event in Aggregate Lag? - If selected, the aggregate lag function includes the value from the current visit. See [Aggregate Lag Functions](#).

Defining Correlating Questions for Procedure Question Groups

To reach this window, select a Question Group in the Question Groups window that has **Correlate w/ Alias?** selected, and click **Correlating Questions**. In this window, you specify the name and occurrence number of the two Questions you are correlating. The system will run the Procedure on the response to the Question in the Question Group displayed higher in the Question Groups window only when the response value of the Question from the lower group is the same. See [Correlation](#).

Before you can carry out the task, you must do the following:

- Select the **Correlate w/ Alias?** field for the current Question Group.
- Define the Questions you want to correlate in the Questions window for the current Question Group and the correlated Question Group.

Defining User Variables

To reach the User Variables window, click **User Vars** in the Procedure Definitions window.

User variables are variables you define and use to store values for testing and to provide temporary placeholders for generated PL/SQL code. You can use them for internal processing and/or to report values to the Discrepancy Database and in discrepancy reports. They can represent a single value or an array of values, and can be of data types character, numeric, or date/time.

You must define user variables here before you can use them in Procedure Detail expressions or custom code.

Oracle Clinical also creates system variables for use in Procedures (see [Using System Variables in Pre-v3.1-Style Procedures](#)).

Variable Name - Name of a user variable, unique within a Procedure; up to 30 characters.

Init Alias - Specify when to initialize a variable; if the field is left blank, Oracle Clinical initializes the variable for each new patient; if you specify an alias, the variable is initialized after each fetch of that alias (Question Group) cursor.

Data Type - Choose the appropriate value from the list.

Length - Specify the maximum length of the value of the variable. If you want to limit the length of a text string, or to cause number rounding, enter the limit here.

Decimal Places - Set the precision of the value of a user variable with data type NUMBER.

Initial Value - Specify the value to which the user variable is initialized. If you leave this field blank, the user variable initializes to a null value.

Report - For pre-V3.1-style Procedures only, select if you want discrepancies found on this variable to be reported. In V3.1-style Procedures you set this in the Detail Variable Settings window. See [Using Variables in V3.1-Style Procedures](#).

Description - Description of the variable, up to 70 characters.

Defining Procedure Details

To reach the Details window, click **Details** in the Procedure Definitions window.

A *detail* is the logical test or calculation performed on data during a Procedure. It consists of a single Boolean statement that, if true, results in an entry in the Discrepancy Database. A given Procedure may contain many details, each testing a particular condition. You must define user variables before referencing them in detail expressions. See [Defining User Variables](#) and [Alternative Approaches to Error Reporting](#).

Define a detail as follows:

- [Order Number](#)
- [Continue If Discrepancy](#)
- [Test Not Null Only](#)
- [Detail Types](#)
- [Externally Derived](#)
- [Failure Type](#)
- [Description](#)
- [Expression](#)
- [Message](#)
- [Discrepancy Code](#)

Order Number

Enter the order in which you want this detail to be executed in relation to the other details in the Procedure. You can change this number in Provisional Procedures only.

Continue If Discrepancy

Select this box if you want the Procedure to continue executing other details if this one finds a discrepancy. If so, specify the number of the detail you want to execute next in the **Cont at Detail#** field. (This functionality is available only in V3.1-style Procedures.)

Test Not Null Only

Select this box if you want the Procedure to execute this detail only if the Question response is not Null.

Detail Types

This field is found in the Derivation Procedure window only, since Calculation-type details are used only by Derivation Procedures to calculate derived Question responses. All other details (all details in Validation Procedures and details in Derivation Procedures that do not directly

derive data) are of type Test, even if they include arithmetic functions and data manipulations. You can use Test details in a Derivation Procedure to ensure that your data is valid before you calculate the derived response, or to check the calculation after deriving it.

Externally Derived

Select this box if the response to this Question is derived by a system outside Oracle Clinical, for example, the Thesaurus Management System (TMS).

Failure Type

Specify the severity of a discrepancy generated by the Detail expression in this detail line. The failure type is associated with the discrepancy in the Discrepancy Database. You can change this field only for Provisional Procedures. Select from the list. The choices **Thesaurus_PT** and **Thesaurus_SCT** are used by the outdated Thesaurus Management Option (TMO) of Oracle Clinical; TMS does not use them. (These values are set in the installation reference codelist VALIDATION FAILURE TYPE CODE; you may want to set the TMO values to Inactive so they do not appear in this list of values.)

Note: Calculation-type details in Derivation Procedures do not generate discrepancies, and therefore, do not use this field.

Description

Free form text, up to 200 characters of mixed case, describing the Detail expression in this detail line of the Procedure. The system enters the text you enter here as the default text in the **Message** field. You can edit and expand it there.

Expression

In Validation Procedures and in Test-type details in Derivation Procedures, the expression is a single Boolean statement that, if true, results in a discrepancy being created. For example, in the test detail `A.BP_SYSTOLIC - A.BP_DIASTOLIC < 20`, the system subtracts the diastolic blood pressure from the systolic and creates a discrepancy if it is less than 20.

You can see all the variables available for use in this expression (Question, system, and user variables) by pressing F9 from this field. See [Procedure Variables](#) for further information.

In details of type Calculation in Derivation Procedures, the expression generates the derived response to a derived Question. Constraints for derivation details of type Calculation:

- You must give the Calculation Detail a lower order number than any other details that use the response.
- You can include user-entered (non-derived) Questions as input to a calculation or test detail, but not as the subject or target of the calculation detail.
- The Derivation Procedure must generate a value of the same data type as the derived Question.

Message

Free form text, up to 500 characters, that appears as a comment associated with any discrepancies this detail generates. The system enters the Description text as the default message.

Discrepancy Code

Select the code to be associated with the discrepancy in the Discrepancy Database according to your company's policy.

Using System Variables in Pre-v3.1-Style Procedures

Reach this window by the **Sys Vars** button in the Procedure Definitions window. In V3.1-style Procedures, this button is grayed out; you mark system variables for reporting with the **Variables** button in the Details window. See [Using Variables in V3.1-Style Procedures](#).

System variables are created by the system when you create Procedure Questions that require them—for example, if you define lags as part of the Procedure, a system variable is created for each lag of the Lag Depth. View the system variables to decide whether to report related test failures as discrepancies.

Using Variables in V3.1-Style Procedures

To reach the Variables Settings window, click **Variables** in the Details window (not available for Calculation-type Details in Derivation Procedures or in pre-V3.1-style Procedures). You can specify the interaction of each detail and each variable as follows:

- [Variable](#)
- [Perform Detail Only If...](#)
- [Discrepancy Test](#)
- [Report](#)
- [Report Order](#)

Variable

The **Variable** field lists all variables (user, system, and Question) currently selected for the detail. You can add variables to the detail by pressing F9 and making selections from the multipick list of values. It includes all remaining variables available to the Procedure. Question variables are Questions you have selected from the **Procedures** menu, by selecting **Question Groups**, and then **Questions**.

Perform Detail Only If...

You can select any combination of the three fields included in this section, each of which is designed to avoid processing inappropriate data.

Not Null

If selected, Oracle Clinical will run this detail on a variable only if the variable actually contains a response value. If the variable is null (because no data was entered for the Question for a patient at a particular visit) Oracle Clinical will skip this detail.

Note that the Procedure treats the value as null if an alpha DVG value has been entered, because alpha DVG values are stored in a different field: the **Exception Value Text** field. Treating alpha DVG values as null is appropriate because the presence of an alpha DVG value indicates a problem with the response: either the response was missing or an alphabetical value was entered for a numeric-type field (such as TRACE for lab data). Variables can have alpha DVG values only if they are Question variables and the Question on which they are based is defined as having an alpha DVG. See [Alpha DVG Subset Examples](#).

No Univariate Error(s)

Applicable to Question variables only. If selected, Oracle Clinical will run this detail on this variable (Question response) only if the response does not have a univariate error of the type(s) you specify in the list of values. If the response has a univariate error of the type(s) you specify, the system will skip this detail. You can choose more than one.

A univariate error is a Question response that is invalid because it does not satisfy conditions specified in the Question definition. The system creates a discrepancy for each univariate error automatically. You may not want to create additional discrepancies based on known discrepancies. The univariate error types are:

- Data Type. The response does not match the defined data type; for example, an alphabetical response for a numeric field, or an incorrect date format.
- DCM Level. Not used.
- DVG. The response entered is not included in the DVG associated with the Question.
- DVG Subset. The response entered is not included in the DVG Subset associated with the Question.
- Length. The response exceeds the length defined for the Question.
- Lowerbound. The response value is lower than the lowerbound limit defined for the Question.
- Mandatory. A mandatory response is missing.
- Upperbound. The response value is higher than the upperbound limit defined for the Question.
- Missing PT. This choice is a vestige of the Thesaurus Management Option, which is no longer supported. It has nothing to do with the current Thesaurus Management System.
- Missing SCT. This choice is a vestige of the Thesaurus Management Option, which is no longer supported. It has nothing to do with the current Thesaurus Management System.

Unresolved

If selected, Oracle Clinical will skip this detail if this variable has an irresolvable univariate error.

Discrepancy Test

If selected, Oracle Clinical will test any discrepancies found on this variable to determine whether they are new or the same as an existing one. If the same, the system does not create a new discrepancy. This field must be selected for at least one variable in each detail. If selected, the system automatically selects **Report** also.

Batch validation processes only changed data, so it does not create duplicate discrepancies. Flag variables whose values you may want to compare between batch validation runs. Do not flag variables—such as Patient Name—that might trigger a discrepancy due to a simple misspelling.

Report

If selected, the value of this variable is visible in the Maintain Discrepancy Database window, associated with the discrepancy. In that window, users can choose whether or not to include the value in the Data Clarification Form as well. Be sure to report the value of the Question being tested so that, if its response value triggers the creation of a discrepancy through this Procedure, it will be possible to see the discrepant value in the Discrepancy Database window and in the Data Clarification Form. This field must be selected for at least one variable in each detail.

Report Order

If you want the variable to be reported, specify the order in which you want it to appear, in relation to other variables from this detail.

Adding Custom Code

If you need to refine Procedure tests, you can add your own PL/SQL statements of up to 4000 characters at certain points in the generated PL/SQL Procedure.

For example, use pre-details custom code to perform procedural logic prior to the detail, accumulate information for previously retrieved DCM or patient records, or retrieve information not available in the generated cursors. Or, in post-details code you can make calculations in preparation for the next cursor fetch, or maintain indexes for array variables. For more information, see [Custom Code](#).

Oracle Clinical places your custom code in the generated PL/SQL code at the location you specify in the **Custom Code Location** box. See [Sample Procedure Code, with Custom Code Locations](#) for the exact location of the custom code types.

An extended list of values for variables is available by pressing F9. See [Procedure Variables](#) for information on the automatically generated variables available for you to use in custom code. See [Defining User Variables](#) for information about creating variables.

To reach the Custom Code window, click **Custom Code** from the Procedures Definitions window. To add custom code of type Post-QG only, you can also click **Custom Code** from the Question Groups window.

To add custom code you must:

1. From the **Custom Code Location** list, choose the place you want to put your code.

2. In the main field, enter the PL/SQL code you want to insert into the generated Procedure code at that location. Press F9 for a list of available variables. You can enter up to 4000 characters.
3. Save.
4. Repeat for each custom code location you need.
5. Generate the Procedure. From the Procedure Definition window, select **Generate** from the **Special** menu.

To view the full, generated PL/SQL code of the Procedure, go to **Edit Proc** in the **Special** menu. You must use the **Procedure with Edit** navigation path to reach the Procedures window. If you do not see that path, you need additional privileges to see the code.

If you have the necessary privileges, you can edit the generated code. However, if you ever generate the code again, your edits are lost. Custom code is not lost when you regenerate the Procedure.

Testing and Activating Procedures

After you define a Procedure, you must generate it and set its status to Active before you can execute it. Oracle Clinical does not enforce Procedure testing, but provides for it (see [Testing a Procedure](#)).

For more information, see:

- [Generating and Compiling a Procedure](#)
- [Editing Generated PL/SQL Code](#)
- [Testing a Procedure](#)
- [Assigning Procedure Status](#)
- [Assigning Standards Affiliation](#)

Generating and Compiling a Procedure

When you invoke generation, Oracle Clinical takes the definitions you have specified and converts them to the PL/SQL code that will perform the Procedure.

1. In the Procedure Definitions window, highlight the Procedure you want to generate and compile.
2. Select **Generate Proc** from the **Special** menu. Oracle Clinical generates the PL/SQL code to execute the Procedure in the background and updates the **Last Generated** field with the current date, or displays an error message. A message confirms when the generation is complete.
3. To view error messages resulting from a failed generation, choose **View Parse Errors** from the **Special** menu.

 **Note:**

If you receive an error on generation that refers to a naming conflict with the Question QGalias.LAB, you have a study Question named Lab that conflicts with the standard procedures lab Question named Lab. To resolve the problem, you can change the standard lab Question alias in the OCL_STATE local reference codelist (under **Admin**).

Editing Generated PL/SQL Code

If you have the required security privileges, you can edit the actual generated PL/SQL code. However, you should take extreme care and use custom code whenever possible (see [Adding Custom Code](#)).

 **Note:**

Changes that you make directly to the generated code (not through the Custom Code window) will be lost the next time the Procedure is generated.

You can also use the editing window to view the generated code and better understand what effect your Procedure definitions and settings have on the generated code.

If you have a navigation path choice called **Procedures w Edit** after either Validation or Derivation Procedures, you have the required privileges to edit generated Procedure code.

1. In the Procedure Definitions window, highlight the Procedure with the generated PL/SQL code you want to edit.
2. From the **Special** menu, select **Edit Gen Proc** to display the generated PL/SQL text.
3. Modify the text.
4. Save. Oracle Clinical returns to the Procedure Definitions window, sets the value in the **Edited?** field to selected, and re-parses the code.
5. Enter a comment about why you edited the text.
6. Click **Back** to return to the original window.
7. From the **Special** menu, select **View Parse Errors**.

Testing a Procedure

Before making a Provisional Procedure active and running it against production data, you should test it (see [Test Mode](#) for information on Oracle Clinical's Test mode). You can then modify the Procedure components as necessary while the Procedure is still Provisional.

You can test the Procedure from either Procedure definition or study testing. You can view the results of testing Validation Procedures in the Test Discrepancy Database; from the **Definition** menu, select **Test a Study**, then choose **Discrepancy Database**. Unlike Production mode, in Test mode each time you run a test, the previous discrepancy data for that test is erased.

For more information, see:

- [Testing a Procedure from Definition](#)
- [Testing a Procedure from Test a Study](#)

Testing a Procedure from Definition

To test a Procedure from Definition, do the following:

1. In the Procedure Definition window, highlight the Procedure and select **Execute in Test** from the **Special** menu. A dialog box appears.
2. If you want to run the Procedure in Debug mode, which lets you view the individual values as they are validated in the .out file, choose Yes. A message appears, giving you the batch job ID number.
3. To check on the job's progress and see the .out and .log file names, select **Batch Jobs** from the **Action** menu.
4. View the results:
 - See the .log and .out files in the procedure output files on the server.
 - See the discrepancies created by the Procedure in the Discrepancy Database by selecting **Show Test Discrep** from the **Special** menu in the Procedure Definition window. Alternatively, from the **Conduct** menu, select **Data Validation**, then **Discrepancy Database**.

Testing a Procedure from Test a Study

To test a Procedure from **Test a Study**, from the **Definition** menu, select **Test a Study**, then **Execute Single Procedure**, and follow standard procedure for submitting batch jobs. To view the results:

- See the .log and .out files in the procedure output files on the server.
- See the discrepancies created by the Procedure in the Discrepancy Database by selecting **Show Test Discrep** from the **Special** menu in the Procedure Definition window. Alternatively, from the **Conduct** menu, select **Data Validation**, then **Discrepancy Database**.

Assigning Procedure Status

Oracle Clinical automatically assigns a status of Provisional (P) to new or copied procedures. You can delete, modify, and test only Provisional Procedures. You can copy and run only Active (A) Procedures. Procedures you no longer want to run during batch validation should be set to Retired (R).

If you have the appropriate security privileges, you can:

- Change a Provisional Procedure to Active.
- Change a Retired Procedure back to Active.
- Change an Active Procedure to Provisional (although this action deletes all discrepancies associated with the Procedure).

 **Note:**

When a Procedure's status is changed from Active to Provisional, all discrepancies associated with it are deleted during the next batch validation run. To avoid this, create a new version of the Procedure from the Special menu."

To change the status, enter the appropriate letter (P, A, or R) in the **Status** field in the Procedure Definitions window.

Oracle Clinical runs all Active Procedures during each batch validation.

Assigning Standards Affiliation

After you have defined a Procedure in a study, generated and tested it, work with the Global Librarian to copy it into the Global Library so that you can use it in other studies. In the Global Library you can also assign it one or more standards affiliation types to identify its purpose, to mark the Procedure as Required (or not) in a study with a particular standards affiliation, and to make the Procedure easier to find for future reference. You cannot define Procedures in the Global Library. (See the *Oracle Clinical Getting Started* manual for information on the Global Library.)

Executing Procedures

Procedures are executed as part of every batch validation, usually scheduled to run at regular intervals. You can also execute a single procedure or batch validation at any time:

- [Executing a Single Procedure](#)
- [Executing Batch Validation](#)

Executing a Single Procedure

You can explicitly execute a single Active Procedure at any time. When you do, the system runs the Procedure over all patients in the current study, not just those patients for whom data has been added or modified (as in batch validation).

The output file displays the number of discrepancies created during this run, the number that remain current and the number that have become obsolete since the last run. The number remaining current reflects the total number of open discrepancies in the study that have been generated by this Procedure.

View discrepancies created by the Procedure from the **Special** menu (**Show Prod Discrep**), or from the **Definition** menu, select **Data Validation**, then **Discrepancy Database**.

You can execute a single Procedure from two locations in Oracle Clinical: the Procedures window and the Execute Single Procedure PSUB window. To execute a Procedure against a Test database, you must execute it from the Procedures window.

For more information, see:

- [Execute a Single Procedure from the PSUB Window](#)
- [Execute a Single Procedure from the Procedures Window](#)

Execute a Single Procedure from the PSUB Window

From the **Conduct** menu, select **Data Validation**, then **Execute Single Procedure**, and do the following:

1. From the list of values, choose the Procedure you want to run.
2. From the list of values, choose the version sequence number of the Procedure you want to run.
3. From the list of values, choose **Y** to run the Procedure in Debug mode, which lets you view the individual values as they are validated in the .out file. The default value is **N**.
4. Click **Submit Job**.

To check on the job's progress and see the .out and .log files, click **Job Status**.

Execute a Single Procedure from the Procedures Window

From the **Definition** menu, select **Validation Procedures** or **Derivation Procedures**, then **Procedures** (or **Prov Procedures** or **Procedures w Edit**; the menu path you see depends on your privileges) and do the following:

1. In the Procedure Definitions window, select the Procedure you want to execute. You can also execute a Procedure from any of the child windows.
2. From the **Special** menu, choose **Execute in Prod** (or **Execute in Test** to run against a test data database). A dialog box appears.
3. If you want to run the Procedure in Debug mode, which lets you view the individual values as they are validated in the .out file, choose **Yes**. A message appears, giving you the batch job ID number.

To check on the job's progress and see the .out and .log files, open the PSUB form (from the **Conduct** menu, select **Data Validation**, then **Execute Single Procedure**) and click **Job Status**.

Executing Batch Validation

You can schedule batch validation to run at regular Intervals (such as nightly) or once at a particular time, or you can manually invoke a batch validation at any time. By selecting **Data Validation** from the **Conduct** menu, and then **Batch Validation Session**, you open a PSUB window. You do not need to enter information in this window. All Procedures in the current study are automatically run every time you run a batch validation.

For further information on batch validation, see Chapter 4, "Using the Discrepancy Database" in *Oracle Clinical Conducting a Study*. For information on submitting batch validation and viewing its output file, see Chapter 3, "Submitting batch jobs and reports" in *Oracle Clinical Getting Started*.

In this section:

- [Facts About Procedure Execution During Batch Validation](#)
- [Batch Validating TMS Derivation Procedures](#)
- [Procedures and Frozen-status Patients](#)

Facts About Procedure Execution During Batch Validation

- During batch validation, Oracle Clinical runs all Derivation Procedures before any Validation Procedures.
- Batch validation always executes incrementally; that is, it processes only data that has changed since the last batch validation run.
- Oracle Clinical detects changes made in the definition of the length of a DCM Question and automatically regenerates any Active Procedure that references that Question before executing the Procedure.

That is, the procedure is regenerated if and when it needs to be executed in a subsequent batch validation run because data has been modified for one or more patients in fields referenced by the Procedure. The Procedure is then run on patients with modified data.

- Whenever a derivation calculation is re-executed, Oracle Clinical compares the resulting value with any existing value for the response. If the new value differs, Oracle Clinical creates a new version of the response and records DATA CHANGE or DERIVATION CHANGE, depending on whether the recalculation resulted from a change to data or to the derivation calculation.
- A study-specific parameter (the study's **Second Pass Required** option) controls if data that has only completed first-pass entry participates in derivation.

Batch Validating TMS Derivation Procedures

Oracle Clinical executes derivation procedures before TMS processing in batch validation runs. This feature enables you to perform the following processes on TMS derivation procedures:

- You can derive a value from TMS into an Oracle Clinical derived Question in the first TMS portion of batch validation, and derive related TMS values for the derived Oracle Clinical Question during the second TMS portion of the same batch validation.
- You can use an Oracle Clinical derivation procedure to derive a value in Oracle Clinical, and send that value to TMS for processing during the second TMS portion of the same batch validation.

These are examples of how this feature can be useful:

- you would like to derive values from two different dictionaries for a single verbatim term.
- You would like to substitute the original collected term with another term, leaving the original response intact; for example, in a historic study or a study conducted by a company you have acquired.

Note that you cannot have more than two levels of derivation for TMS Questions. The TMS code only handles the following scenario:

- TMS parent Question
- Derived Question (This Question is in turn used as a parent Question.)
- Derived Question

You cannot create a fourth derived Question.

See the *Oracle Thesaurus Management System User's Guide* for more information.

Procedures and Frozen-status Patients

Oracle Clinical runs Derivation Procedures even for patients whose status is Frozen; in these cases, recalculation can take place when:

- an existing Procedure is changed
- a new Procedure is created
- an existing, unchanged Procedure is explicitly executed over all patients, and calculation is based on external data that has changed
- a patient was frozen before having new or modified data processed by batch validation

Derivation Procedures containing validation expressions may produce discrepancies for patients with a status of Frozen.

V3.1-Style and Pre-V3.1-Style Procedures

Oracle Clinical V3.1 introduced a new style of Procedure, with functional enhancements. V3.1-style Procedures consist of static packages in the database rather than anonymous PL/SQL blocks, eliminating the need to parse a Procedure each time it is executed, for greater efficiency and speed.

You can continue to run existing pre-V3.1-style Procedures in V3.2 and later releases with their current functionality, or you can explicitly convert them to the V3.1 style for enhanced performance. After converting a Procedure to the V3.1 style, you can modify it to take advantage of the additional functionality. You can create new Procedures in either the newer V3.1 style (the default) or the old (pre-V3.1) style.

For more information, see:

- [V3.1-Style Procedure Enhancements](#)
- [Converting to the V3.1 Style](#)
- [Creating a New Pre-V3.1-Style Procedure](#)

V3.1-Style Procedure Enhancements

- Increased speed of execution
- Possible to add custom code in more locations in the code without editing generated PL/SQL
- Separate **Report?**, **Test Not Null?**, **Continue If Discrepancy?** boxes and other settings for each variable as it is associated with each detail, making it possible to include more details in a single Procedure
- Ability to prevent processing Question responses with current univariate discrepancies, and a multipick list of values to select which univariate discrepancies prevent processing in each detail
- Option to choose which detail to process next, if Oracle Clinical finds a discrepancy and the **Continue If Discrepancy?** box is selected
- Embedded data values in discrepancy messages to improve context sensitivity
- Additional event order processing choices based on:

- the leading fields **DCM Date** and **Time**, plus **Visit** and **Subevent Numbers**
- acceptance of manually coded processing order based on trailing fields, usually **Qualifying_Value** and/or **Repeat_SN**
- Single repeat only option for repeating Question Groups
- Capacity to correlate Questions with the previous or next event, as well as the actual event
- A new standard variable, *question\$exc_val*, which provides the EXCEPTION_VALUE_TEXT field value for all Question Group Questions
- An extended multipick list of values for variables, including Question Group Cursor key fields, Patient Position Cursor fields, internal Procedure variables, and DVG information, customized according to the field from which it is invoked
- Optional re-execution of Procedures for patient records affected by lab range and lab range subset changes made since the last batch validation

Converting to the V3.1 Style

When converting a pre-V3.1-style Procedure to the V3.1 style, it is helpful to understand what Oracle Clinical does internally to accomplish the conversion:

1. Populates the data structure, Variable Settings, for each detail, as appropriate.
2. Generates the Procedure in V3.1 style, as a packaged object. This step occurs only if the Procedure being converted has been successfully generated; if not, you are prompted to generate the Procedure after conversion. Oracle Clinical will not generate Global Library Procedures.
3. Switches the **Ver 3.1 Style?** box to selected if the conversion was successful.

 **Note:**

Oracle Clinical saves the pre-V3.1 version of the Procedure, and reverts to it if problems are encountered in conversion.

Oracle Clinical preserves all current discrepancies associated with a Procedure when you convert the Procedure.

To convert a single existing pre-V3.1-style Procedure to the V3.1 style:

1. Before beginning the automatic Procedure conversion, if the following three internal variable names appear in existing Pre-Details or Post-Details custom code, manually change them from the pre-V3.1-style to their V3.1-style names.

Table 16-1 Style Variables Comparison

Pre-V3.1-style Variable	V3.1-style Variable Name
patients_rec	rxcpdstd.patients_rec
test_or_prod	rxcpdstd.v_mode
batch_start_ts	rxcpdstd.v_current_batch_ts

2. Start the automatic Procedure conversion process by choosing **Convert to 3.1 Style** from the **Special** menu in the Procedure Definitions window.

 **Note:**

You can convert only Procedures with a status of Active or Provisional (not Retired).

3. If the original Procedure had Pre- or Post-Details custom code (Pre-Code or Post-Code), and that code accessed database tables (such as Patient_Positions, Responses, or Received_DCMs), you must edit the code to explicitly account for Production and Test environments. (In pre-V3.1-style Procedures, Oracle Clinical pointed to test or production tables through synonyms, which is no longer possible in the Packaged Object Model.)

 **Note:**

Convert Procedures in bulk with the GEN_PROCS utility. See Chapter 8, "Utilities" in the *Administrator's Guide*.

For more information , see:

- [Conversion Example](#)

Conversion Example

Original, pre-V3.1-style Procedure Pre-Details custom code:

```
...
SELECT DATA_COMMENT_TEXT
INTO COMMENT_VAR
FROM RESPONSES
WHERE RESPONSE_ID = A.QUESTION1$RESP_ID
AND RESPONSE_TS = A.QUESTION1$ENT_TS;
...
```

The code after conversion by Oracle Clinical (the same, in this case; if the code had included the variables PATIENTS_REC., TEST_OR_PROD, or BATCH_START_TS, they would have been renamed RXCPDSTD.PATIENTS_REC., RXCPDSTD.V_MODE, or RXCPDSTD.V_CURRENT_BATCH_TS, respectively):

```
...
SELECT DATA_COMMENT_TEXT
INTO COMMENT_VAR
FROM RESPONSES
WHERE RESPONSE_ID = A.QUESTION1$RESP_ID
AND RESPONSE_TS = A.QUESTION1$ENT_TS;
...
```

The code after you have made the required edits to account for test and production tables' references:

```
IF RXCPDSTD.V_MODE = 'P'
THEN /* PRODUCTION RUN*/
```

```
...
SELECT DATA_COMMENT_TEXT
INTO COMMENT_VAR
FROM RESPONSES
WHERE RESPONSE_ID = A.QUESTION1$RESP_ID
AND RESPONSE_TS = A.QUESTION1$ENT_TS;
...
ELSE /* TEST RUN*/
...
SELECT DATA_COMMENT_TEXT
INTO COMMENT_VAR
FROM RESPONSEST
WHERE RESPONSE_ID = A.QUESTION1$RESP_ID
AND RESPONSE_TS = A.QUESTION1$ENT_TS;
...
END IF;
```

 **Note:**

For a test run, add a "T" to the name of the table to indicate the test table with the same type of data as the production table of the same name. For example, ResponsesT, not Responses.

Creating a New Pre-V3.1-Style Procedure

The default style for new Procedures from 3.1 on is the 3.1 style. However, you can create a new, pre-V3.1-style Procedure by deselecting the **Ver 3.1 Enabled?** box for the Procedure in the Procedure Definitions window.

You can do this only when you first create the Procedure definition, before saving it.

Maintaining Patient Statuses

You must create a Validation Procedure to populate the Patient Statuses table for each study using two packaged procedures provided with Oracle Clinical. Data in the Patient Statuses table is used by the Oracle Clinical Enrollment form and in several reports, including the Patient Enrollment Report.

To populate the table initially, use the RXC_PAT_STAT package. To ensure that the information in the Patient Statuses table reflects any change in status entered in a CRF, use PAT_SYNCH, a function in RXC_PAT_STAT; see [Patient Synchronization](#).

For more information, see:

- [Populating the Patient Statuses Table](#)
- [Patient Synchronization](#)
- [Example](#)
- [Pre-Details Custom Code](#)
- [Alternative Approaches to Error Reporting](#)
- [Type Coercion and Comparison of Derived Values](#)

Populating the Patient Statuses Table

As patients progress through a study, they achieve various statuses, such as ENROLLED and TERMINATED. The status ENROLLED is used by the Enrollment form and reports to define when a patient is considered to be enrolled in a study. All studies must use the status of ENROLLED, and can use as many or as few other statuses as are relevant for tracking patients in that study. Other statuses are user-defined in the installation reference codelist PATIENT STATUS CODE.



Note:

The ENROLLED status has no effect on the unenrolled patient alert, lab criteria assignment, batch loading enrollment data, or access to studies in Oracle Clinical Remote Data Capture.

Each patient status has associated with it a Status Date—the date the patient achieved the status. So, the Status Date associated with a patient status of ENROLLED is the Enrollment Date; the Status Date associated with the patient status of TERMINATED is the Termination Date. A patient can therefore have several statuses at the same time, one for each status achieved so far:

Patient	Status	Status Date
100	ENROLLED	01092003
100	TERMINATED	12092003
101	ENROLLED	03092003
102	ENROLLED	04092003
102	TERMINATED	10092003
103	ENROLLED	07092003
104	ENROLLED	19092003

You must create a special Validation Procedure in each study to populate the Oracle Clinical Patient Statuses table. In the Pre-Details custom code within the Procedure, use a package, RXC_PAT_STAT provided with Oracle Clinical to insert and delete status codes for each patient. The package includes two functions:

- The CLEAR_STAT function takes as input a patient_position_id, and deletes all existing patient status records for a particular patient: Procedure CLEAR_STAT(PATIENT_ID IN NUMBER)
- The SET_STAT function takes as input several keys for a patient, as well as a **Status** (code), a **Status Date**, and a **Comment**. These status fields are inserted into the patient statuses table as follows:

```
Procedure set_stat(clinical_study_id in number,
site_id in number,
patient_id in number,
status in varchar2,
status_date in date,
status_comment varchar2)
```


You must ensure that the CLEAR_STAT function is executed only once per patient, and that a given status is inserted only once. You can use any of the following techniques:

- Access DCMs that can only have a single occurrence, such as Demography and Termination. This is the approach used in the example below.
- Use only aggregate Procedure Question Groups with the MAX function. Each one returns a single row, so the Pre-Code executes only once. However, you can not access the DCM_Date in an aggregate Procedure Question Group.
- Define variables to set flags indicating that the CLEAR_STAT Procedure has been called for the patient, to ensure it is called only once. Write the status calculation logic so that a given status can be recorded only once.

For more information , see:

- [Example](#)

Example

Following is an example of a Procedure to track ENROLLED and TERMINATED statuses along with Enrollment Date and Termination Date, referencing DCMs with a single occurrence. The Enrollment Date is the DCM_DATE of the Demog DCM, and the Termination Date is the Question TERM_DATE asked on the Term DCM. Create the Validation Procedure for synchronizing the statuses and dates between the responses and the Patient Statuses table in the following manner:

Procedures

Define the Validation Procedure itself (see [Naming and Describing a Procedure](#) .

Procedure Question Groups

Select two Questions Groups, one from each DCM that contains the Question that affects the patient status: DEMOG and TERM. Select **Create Place Holder?** for each group. Do not correlate the groups. (See [Defining Procedure Question Groups](#).

Procedure Questions

In the example code below, the alias of the DEMOG Procedure Question Group is D. Select DCM_DATE as its Procedure Question. The alias of the TERM Procedure Question Group is T. Select TERM_DATE as its Procedure Question. (See [Defining Procedure Questions](#).

Procedure Details

Define one Procedure detail with a FALSE Boolean expression to prevent the Procedure from creating discrepancies (for example, 2<1). Leave the other fields in the window alone; no other field is relevant to the purpose of this Procedure.

Variable Settings

Procedure definition rules require that you mark at least one variable of this detail for Report and for Discrepancy Test. Because the expression is false, the Procedure never creates a discrepancy and it does not matter which variable you set.

Custom Code

In the Custom Code window, choose Pre-Detail and enter the following:

```
/* clear existing patient statuses */  
/* current patient key is RXCPDSTD.patients_rec.patient_position_id */  
rxc_pat_stat.clear_stat(RXCPDSTD.patients_rec.patient_position_id);
```

```

        /* Set enrollment date into patient_status */
        if D$has_data = 'Y' then
            rxc_pat_stat.set_stat(i_clinical_study_id      /* clinical_study_id
*/,
                                RXCPDSTD.patients_rec.site_id      /* site_id *//,
                                RXCPDSTD.patients_rec.patient_position_id      /* patient_id
*/,
                                'ENROLLED'      /* status *//,
                                rxc_date.convert(D.DCM_DATE)      /* date must be converted to
native Oracle date format for storage *//,
                                " "      /* status_comment *//);
        end if;
        /* Set termination date into patient_status */
        if T$has_data='Y' then
            rxc_pat_stat.set_stat(i_clinical_study_id      /* clinical_study_id
*/,
                                RXCPDSTD.patients_rec.site_id      /* site_id *//,
                                RXCPDSTD.patients_rec.patient_position_id      /* patient_id *//,
                                'TERMINATED'      /* status *//,
                                rxc_date.convert(T.TERM_DATE)      /* date - must be converted to
native Oracle date format for storage *//,
                                " "      /* status_comment *//);
        end if;

```

This logic guarantees that the Termination and Enrollment Dates for each patient in the study will be obtained if they exist. The algorithm is:

```

loop over each patient{
  get the termination date (if any of these CRFs exist)
  get the enrollment date (if any of these CRFs exist)
  delete all status information for the current patient
  if a termination form is present, store the termination status with the termination date
  if an enrollment form is present, store the enrollment status with the enrollment date
}

```

Each Procedure Question Group has **Create Placeholder?** selected, so that if a patient has no Termination or Enrollment data, the Pre-Code still executes and the CLEAR_STAT function removes any previous statuses. If an Enrollment record is removed from the system because it was mistakenly logged in, the Enrollment status code is cleared by CLEAR_STAT and not reinserted by SET_STAT.

The test on alias\$has_data is present to force the insertion of a status only in the case where a relevant Question contains response data.

Patient Synchronization

The Patient Positions table includes information that may be collected as part of clinical trials data. To ensure that the information in the Patient Positions table reflects updates to a patient's data entered in Data Entry, you can use the packaged function RXC_PAT_STAT.PAT_SYNCH in any Derivation or Validation Procedure.

Since Validation and Derivation Procedures are automatically executed during batch validation whenever a patient's data changes or when a Patient Enrollment Record is

modified, the PAT_SYNCH function is also re-executed after any event that might change the consistency between the patient enrollment data and the CRF data. If you access any patient information in other Derivation or Validation Procedures (including through access to lab ranges) you should call the PAT_SYNCH function from a Derivation Procedure executed before any Procedure that uses the information.

The function compares the current values of the Patient Enrollment information with the values passed in and returns results as follows:

- If the values are identical, the function changes nothing and returns SUCCESS.
- If any value in the Patient Positions table is null and the corresponding value passed in is non-null, the function updates the enrollment values in the Patient Positions table to the passed values, sets the Modification TS to the Batch Start TS of the batch validation run, and returns SUCCESS.
- If any value in the Patient Positions table for a given patient is non-null and differs from the corresponding value passed in, and the value passed in is non-null, then the function returns FAILURE. You should set the **Report?** option for the variable to create a discrepancy.

For more information, see:

- [Procedure Definition](#)

Procedure Definition

Incorporate the PAT_SYNCH function into your Procedure in the same way as the other two packaged functions (see [Example](#)).

- Reference the Question Groups and Questions that collect patient status data. The names of the Questions (and therefore Procedure variables) need not be the same as the arguments to the function or the fields in the Patient Positions table (for instance BIRTH_DATE and DOB).
- Call the function from the Pre-details custom code of the Procedure.
- If necessary, the collected data values should be converted to the data type (for instance, DATE) of the same values in the Patient Positions table.
- Declare user variables to receive the return status and the returned values from the Patient Positions table, and also to act as placeholders in the Procedure call for any input arguments you do not want to update.
- Mark as **Report?** only those user variables you want to report as part of the discrepancy.
- The input arguments can include both Procedure variables (for instance D.INITIALS) and user variables (for instance TEMP_SEX).
- Do not select **Continue If Discrepancy?** in the Details window. See [Type Coercion and Comparison of Derived Values](#).

Arguments passed to PAT_SYNCH as nulls are ignored. This means you need not synchronize a particular patient's data if it is not collected. It also means that PAT_SYNCH can be called from several Procedures as long as each Procedure passes in different variables.

For instance, one Procedure could update SEX, BIRTH_DATE, and INITIALS, while another updates FIRST_SCREENING_DATE, ENROLLMENT_DATE, and TERMINATION_DATE (perhaps the same Procedure that records patient status information using the RXC_PAT_STAT package).

The return status of the Procedure can be tested in Procedure Details to enable reports on conflicts. Optionally, a new validation failure severity and/or Procedure type could be used to distinguish this new type of validation failure. See [Alternative Approaches to Error Reporting](#).

Table 16-2 lists the variables you must define in order to pass them as arguments to the functional call, with D_ arguments as the input data values and the P_ arguments as the return patient enrollment values.

Table 16-2 List of Arguments for Procedure Definition

Argument Name	In/Out	Data Type	Comment
test_or_prod	in	varchar2	Indicates whether the Procedure was called in test (T) or production (P) mode
batch_start_ts	in	date	Date/time of beginning of batch run
d_patient_position_id	in	number	Internal ID of Patient
d_reported_first_name	in	varchar2	-
d_reported_last_name	in	varchar2	-
d_reported_sex	in	varchar2	Value must be M or F
d_reported_patient_reference	in	varchar2	-
d_reported_initials	in	varchar2	-
d_reported_birth_date	in	date	-
d_reported_date_last_pregnancy	in	date	-
d_first_screening_date	in	date	-
d_patient_enrollment_date	in	date	-
d_termination_date	in	date	-
d_early_termination_flag	in	varchar2	Value must be Y or N
p_reported_first_name	out	varchar2	-
p_reported_last_name	out	varchar2	-
p_reported_sex	out	varchar2	-
p_reported_patient_reference	out	varchar2	-
p_reported_initials	out	varchar2	-
p_reported_birth_date	out	date	-
p_reported_date_last_pregnancy	out	date	-
p_first_screening_date	out	date	-
p_patient_enrollment_date	out	date	-
p_termination_date	out	date	-
p_early_termination_flag	out	varchar2	-

Example

This Procedure handles synchronization of SEX, BIRTH_DATE, INITIALS, and TERMINATION_DATE.

Validation Procedure: PATIENT_DATA_SYNC

Procedure Question Group: DEMOGRAPHICS, Alias: D

Procedure variable: SEX
 Procedure variable: DOB
 Procedure variable: INITIALS

Procedure Question Group: TERMINATION, Alias: T, No correlation, Create Placeholder:

Procedure variable: TERMINATION_DATE

This structure results in a single execution of the Pre-details custom code for each patient. The execution occurs even if the termination data has not been received.

User Variable Name	Report?	Data Type	Length
d_date_last_pregnancy	n	date	-
d_early_termination_flag	n	varchar2	1
d_first_name	n	varchar2	15
d_first_screening_date	n	date	-
d_last_name	n	varchar2	20
d_patient_enrollment_date	n	date	-
d_patient_reference	n	varchar2	25
enr_birth_date	{1} y	date	-
enr_date_last_pregnancy	n	date	-
enr_early_termination_flag	n	varchar2	1
enr_first_name	n	varchar2	15
enr_first_screening_date	n	date	-
enr_initials	y	varchar2	4
enr_last_name	n	varchar220	-
enr_patient_enrollment_date	n	date	-
enr_patient_reference	n	varchar2	25
enr_sex	y	varchar2	1
enr_termination_date	y	date	-
pat_synch_return	n	varchar2	7
temp_dob	n	date	-
temp_sex	n	varchar2	1
temp_term_date	n	date	-

Pre-Details Custom Code

```
{2}
if (D.SEX = 'MALE') then TEMP_SEX = 'M';
elsif (D.SEX = 'FEMALE') then TEMP_SEX := 'F';
end if;
{3}
TEMP_TERM_DATE := to_date(T.TERMINATION_DATE, 'YYYYMMDD');
TEMP_DOB := to_date(D.DOB, 'YYYYMMDD');
PAT_SYNCH_RETURN := RXC_PAT_STAT.PAT_SYNCH(
  RXCPDSTD.V_MODE,
  RXCPDSTD.V_CURRENT_BATCH_TS,
```

```
RXCPDSTD.patients_rec.PATIENT_POSITION_ID,  
D_FIRST_NAME,  
D_LAST_NAME,  
{4}      TEMP_SEX,  
D_PATIENT_REFERENCE,  
{5}      D.INITIALS,  
TEMP_DOB,  
D_DATE_LAST_PREGNANCY,  
D_FIRST_SCREENING_DATE,  
D_PATIENT_ENROLLMENT_DATE,  
TEMP_TERM_DATE,  
D_EARLY_TERMINATION_FLAG,  
ENR_FIRST_NAME,  
ENR_LAST_NAME,  
ENR_SEX,  
ENR_PATIENT_REFERENCE,  
ENR_INITIALS,  
ENR_BIRTH_DATE,  
ENR_DATE_LAST_PREGNANCY,  
ENR_FIRST_SCREENING_DATE,  
ENR_PATIENT_ENROLLMENT_DATE,  
ENR_TERMINATION_DATE,  
ENR_EARLY_TERMINATION_FLAG);
```

In lines 9, 10, and 11 the internal variable names are: TEST_OR_PROD, BATCH_START_TS, and PATIENTS_REC.Site_ID.

Alternative Approaches to Error Reporting

You can choose to create one or potentially multiple failures using one of these approaches:

- [Single Failure](#)
- [Multiple Failure](#)

Single Failure

A single Validation Detail can test the return status variable and issue a single failure covering all inconsistencies. For instance:

Detail 1:

Test: PAT_SYNCH_RETURN = 'FAILURE'

Message: Patient Enrollment information differs from CRF data.

Since the error is a multivariate error, all the potentially differing values are associated with the resulting discrepancy.

Multiple Failure

Multiple Details can test for each potential mismatch and report them individually. For instance, assuming that input values come from the Procedure Question Group with alias D (Demography):

Detail 1:

Test: PAT_SYNCH_RETURN = 'FAILURE' AND D.SEX != ENR_SEX

Message: Patient's Sex differs between Patient Enrollment and the CRF data.

Detail 2:

Test: PAT_SYNCH_RETURN = 'FAILURE' AND D.BIRTH_DATE != ENR_BIRTH_DATE

Message: Patient's Birth Date differs between Patient Enrollment and the CRF data.

Type Coercion and Comparison of Derived Values

Be sure to convert data types before passing them to the PAT_SYNCH function.

If you have not selected **Continue If Discrepancy?** and an incorrect value is derived, whether a length or a data type error, the Procedure's execution stops. All derivations and validations performed by that Procedure are rolled back to reflect the last correct values. Errors are written to the log file (the batch validation log file or the single procedure execution log file). However, the next time the Procedure is run, it is run for all patients.

Calculation expressions specified by users can be one of three data types: VARCHAR2, NUMBER, and DATE. Derived Questions can be one of four types: CHAR, NUMBER, DATE, and TIME. TIME Questions have a length of 4 or 6, and DATE Questions have a length of 4, 6, or 8 and are stored in YYYYMMDD format. CHAR and NUMBER Questions also have predefined lengths that must be enforced.

The following table shows conversions of Oracle Clinical types to Oracle native types. These CHAR conversions are automatically performed by the Procedure in the Calculation phase. Note that decimal precision on numbers is not enforced nor coerced.

Oracle Clinical Derived Question Type	Oracle Native Type of Derived Expression		
	varchar2	number	date
char	Length is checked	Error 20104	Error 20104
number	Error 20104	Length is Error 20104 checked	Error 20104
date	Date is checked in YYYYMMDD format; partial dates allowed	Error 20104	Date is converted to YYYYMMDD format before being stored
time	Time is checked in hhmm or hhmmss format, depending on the length of the varchar2 passed in	Error 20104	Error 20104

Test Mode

This chapter describes Oracle Clinical's Test mode, which is a reduced set of features for testing a study's design and some data definitions. It is a scaled-down version of some Production mode functionality. Test mode is separate from production mode, so it is unsuitable for validating an installation or system functions. This chapter includes the following topics:

**Note:**

For a new database installation, implementing Test mode requires these preliminary tasks:

- The Global Librarian should create the domain `INSTALL_TEST` to hold objects for testing purposes. (See [Using Domains within the Global Library](#) for more information.)
- Manually enforce the constraint that two domains cannot reference each other, and that only `STANDARD` objects can be available for production work.

For more information, see:

- [Overview](#)
- [Launching Test Mode Sessions](#)
- [Setting Up Test Study Design](#)
- [Creating and Loading Test Data](#)
- [Testing Definitions](#)

Overview

Test mode is a tool for testing your procedures and your data entry layouts before you go into production.

You can test the usability of data entry screens, that the Validation Procedures identify possible discrepancies, and that the Derivation Procedures correctly calculate the responses you want. Here are some ways you can use Test mode in the Definition subsystem:

- You can test the appearance of data entry form layouts in Oracle Clinical and RDC Onsite.
- You can test the algorithms in validation and derivation procedures.
- You can use Test mode to train new data entry operators on a particular study's data entry screens.
- You can test data extract views before entering production data.
- You can test Mass Changes before applying them to production data.

See [Testing Definitions](#) for more information.

In this section:

- [A Typical Test Mode Workflow](#)
- [How Test Data and Production Data Differ](#)
- [Test Mode Limitations](#)

A Typical Test Mode Workflow

This is a typical testing sequence:

1. Test a live study's design. See [Setting Up Test Study Design](#).
2. Add some definitions to the study from Production mode. These are some production definitions you can test in Test mode:
 - DCMs
 - DCM Layouts
 - DCIs
 - Patient Positions assigned to sites and Investigators
 - Clinical Planned Events
 - Validation and derivation Procedures
 - Mass Changes
3. Start a Test mode session and refine patient, Investigator, and site definitions. Oracle Clinical generates some dummy definitions for you, but you can add to them.
4. Create test data. See [Creating Test Data](#).
5. Batch load your test data. If you have a lot of test data, you can batch load it. See [Creating and Loading Test Data](#).
6. Batch validate your data. See [Testing Validation Procedures](#).
7. Review the results of the batch validation. See [Checking Validation Results](#).
8. Test Mass Change functions.
9. Test your Data Extract Views; see [Testing Data Extract Views](#).
10. If you are using Remote Data Capture (RDC), test data entry layouts; see [Testing Layouts in Remote Data Capture \(RDC\) OnSite](#).

How Test Data and Production Data Differ

Oracle Clinical prevents using test data in Production mode and production data in Test mode. For example, when you navigate to **Design, Test a Study Design**, and **Investigators**, you can enter or select only dummy Investigators, sites, and patients; the system places any data you enter for the CRFs in tables designed to contain only test data. Conversely, if you navigate to **Design**, then **Investigators and Sites**, you can only enter or select non-test objects; the system places the data in Production mode tables.

The design objects you see from the Test a Study Design navigation path is not shared. However, DCMs, DCM layouts, DCIs, procedures, and Mass Change criteria are shared between Production and Test modes.

Test Mode Limitations

Oracle does not support using Test mode for the certain operations:

- Oracle does not support using Test mode to validate an entire Oracle Clinical installation, or individual system functions.
- Test mode cannot emulate the discrepancy management workflow process. In particular, you cannot generate Data Clarification Forms (DCFs), in Test mode.
- Test mode does not optimize indexes or queries.
- Test mode does not preserve or track the results of previous test executions.
- Test mode cannot handle the large amounts of patient data you would typically collect in Production mode.
- Test mode deletes all existing discrepancies and derived responses each time you run batch validation in Test mode.
- Batch Validation in Test mode runs derivation and validation procedures with Provisional status. (Production mode does not.) This difference can impact performance in Test mode if you have many provisional procedures.

Launching Test Mode Sessions

Open a session in Test mode by entering a particular URL in your browser as follows, where *computer_name.domain* is the connecting application server's network name, a separating period (.), and your organization's network domain name (for example, oracle.com). These are the Test mode URLs:

- RDC in Admin Test mode:

`http://computer_name.domain/opa50/rdcadmint.htm`

- RDC Onsite Test mode:

`http://computer_name.domain/olsa/oc/rdcLogin.do?event=doSetup&mode=T&d`

Oracle Clinical - There is no separate Test mode URL for Oracle Clinical. Log in as usual and choose menu items that include the word "Test."

Setting Up Test Study Design

When you create a live clinical study version, Oracle Clinical automatically assigns test sites, Investigators, and ten patient positions to it. This provides the framework you need to test your definitions. You can use Test mode layouts and forms as you would their Production mode counterparts, ensuring that they correctly represent the study's requirements.

When you enter data into a CRF, you specify or select Investigator, Site, Patient, Clinical Planned Events (CPEs), and Visits. Oracle Clinical uses Test mode data for Investigator, Site, and Patient information. However, Oracle Clinical uses the same CPE and Visit information for both Production and Test mode. (The data comes from the same tables.)

The test sites, Investigators, and patient position names that are included with the installation have a leading x to distinguish them from production data. If you need additional cases you can modify or add to this data. For example, you can modify the generated details in a patient position where you need procedures requiring different sexes or dates of birth.

If you modify or supplement these test objects, Oracle recommends adhering to a naming convention that avoids confusion with production data. Oracle Clinical prevents using test data in Production mode, but hard copy reports or SAS data sets with test data that have similar names to production data could cause confusion.

The Design Test mode systems are identical to their production counterparts. For instructions for using these utilities, refer to the references included in this list:

- Testing Investigator Design: See [Maintaining Investigator Records](#). From the **Design** menu, select **Test a Study Design**, then **Investigators**.
- Testing Site Design: See [Maintaining Site Records](#). From the **Design** menu, select **Test a Study Design**, then **Sites**.
- Testing Study Sites: See [Assigning Sites to a Study](#). From the **Design** menu, select **Test a Study Design**, then **Study Site**.
- Testing Site Investigator Assignments and Details: See [Assigning Investigators to Study Sites](#).
- Testing Patient Positions Maintenance: See [Maintaining Patient Positions](#). From the **Design** menu, select **Test a Study Design**, then **Creating/Deleting Patient**.
- Linking patients to a test study: See [Linking Patient Positions in Different Studies](#). From the **Design** menu, select **Test a Study Design**, then **Patients**, and click the **Patient Positions** button.
- Maintain test study Patient Positions: See [Maintaining Patient Positions](#) for more information. From the **Design** menu, select **Test a Study Design**, then **Patients**, and click the **Patient Positions** button.

For more information, see:

- [About Test Mode Synonyms](#)

About Test Mode Synonyms

Oracle Clinical creates the following private synonyms that point to the test tables or to the views in Test mode. When you access a Test mode option, Oracle Clinical creates synonyms for you that point to the test tables. Oracle Clinical drops the synonyms for Production mode.

Table 17-1 Synonyms

Test Synonym	Production Mode Table or View
ACTUAL_EVENTST	ACTUAL_EVENTS
ACTUAL_EVENTS1T	ACTUAL_EVENTS1
ACTUAL_EVENTSVT	ACTUAL_EVENTSV
DISCREPANCY_ENTRIEST	DISCREPANCY_ENTRIES
DISCREPANCY_ENTRY_REVIEW_HISTT	DISCREPANCY_ENTRY_REVIEW_HIST

Table 17-1 (Cont.) Synonyms

Test Synonym	Production Mode Table or View
DISCREPANCY_MANAGEMENTT	DISCREPANCY_MANAGEMENT
INVESTIGATORST	INVESTIGATORS
INVESTIGATORST	OCL_INVESTIGATORS
SITEST	OCL_SITES
RXC_STUDY_SITEST	OCL_STUDY_SITES
STUDY_SITE_ROLEST	OCL_STUDY_SITE_ROLES
OPA_LEVEL_PRIVST	OPA_LEVEL_PRIVS
PATIENT_POSITIONST	PATIENT_POSITIONS
PATIENT_POSITIONS_VT	PATIENT_POSITIONS_V
RDC_AUDIT_VIEWT	RDC_AUDIT_VIEW
RDC_DISCREPANCIEST	RDC_DISCREPANCIES
RDC_PDF_VRVST	RDC_PDF_VRVS
RDC_VRVST	RDC_VRVS
RDCI_HISTORYT	RDCI_HISTORY
RECEIVED_DCIST	RECEIVED_DCIS
RECEIVED_DCIS_VT	RECEIVED_DCIS_V
RECEIVED_DCMST	RECEIVED_DCMS
RECEIVED_DCMS_VT	RECEIVED_DCMS_V
RECEIVED_PAGE_HISTORYT	RECEIVED_PAGE_HISTORY
RECEIVED_PAGEST	RECEIVED_PAGES
RESPONSEST	RESPONSES
RXC_STUDY_SITEST	RXC_STUDY_SITES
SITEST	SITES
STUDY_SITE_PATIENT_POST	STUDY_SITE_PATIENT_POSITIONS
STUDY_SITE_ROLEST	STUDY_SITE_ROLES
VALIDATION_REPORTED_VALUEST	VALIDATION_REPORTED_VALUES
VALIDATION_REPORTED_VALUES_VT	VALIDATION_REPORTED_VALUES_V

Until you move a study to production, *all* of your Oracle Clinical sessions for that study are in Test mode. When you are in Test mode, your Oracle account points to test tables. This condition applies even after you log out and log back in. You can check whether you are pointing to test or production tables by looking in the User Synonyms table in a SQL session.

Creating and Loading Test Data

You can batch load test data. This can be helpful, for example, to test sample files from labs to confirm they are delivering the correct file formats. Oracle Clinical enables you to batch load data into test database tables, batch delete data from the test tables, and execute a batch validation session on test data. Test mode Batch Data Load is a way to get test data into the system, and to check that your files formats are correct. However, the Test mode

batch system is not optimized for large volumes of data, and it uses separate tables; so there's no correlation between test mode and production mode performance.

For more information, see:

- [Creating Test Data](#)
- [Batch Loading Test Data](#)
- [Using Test Mode Batch Data Delete](#)

Creating Test Data

These are some guidelines for creating test data:

- Use actual clinical study data from a previous study for the best test data. You can batch load data in Test mode.
- Test mode Batch Data Load is not tuned for performance. If you have a large BDL file, consider taking a subset of the file to reduce load time.
- Embed the test cases in error-free data.
- List all the ways each Question on the DCM for which you are building test data can be invalid. Consider these points of view:
 - Misleading responses from patients
 - Misunderstanding the patient response by the clinical staff (Omissions and commissions)
 - The data entry operators' mistakes reading the CRF, misreading the CRF, mistakes of entry, misunderstanding wording on the DCM layout).
- Include responses that are invalid from the point of view of the system, but not from the point of view of the person responding.
- Modify your test data to create responses that are out of range, not in the correct DVG, incorrect units, incompatible with other responses on the DCM, or from earlier planned events.

Create test data that is valid, slightly invalid (just over the range, for example), and extremely invalid; sometimes Validation Procedures catch errors near a set point, but fail to handle errors totally outside the expected range.

Batch Loading Test Data

Batch load test data to make the following tests:

- Internal study definition—of, for example, format masks, DCIs, DCMs, and DCI_Modules
- Validation and derivation Procedures

Test batch data load works like production batch data load, giving similar errors and warnings in similar circumstances; however, objects in a test load are test objects, and data is loaded into test database tables.

Test batch data load differs from production batch data load in two additional respects:

- You can apply both provisional and active format masks to test data files and test operating system files.

- You can use both provisional and active DCIs, DCMs, and DCIs in test batch load files.

Using Test Mode Batch Data Delete

Deleting data from test database tables allows reproducible testing of study conduct and data entry on provisional objects, which is particularly useful during the initial setup of a study. Deleting batch data also frees up database space.

When full study data delete is run in Test mode, all relevant test mass change information is deleted for the study.

The data you delete depends on the input parameters from the test database tables for a study. Parameters are DCI, Document Number, Data File, and Data File Group. DCIs can be Active or Provisional; but the three other parameters must be Test. When data is deleted, it can be deleted from actual tables as well as the staging, according to the parameters input.

Testing Definitions

You test data entry screens by using them to enter test data into a special test database in Oracle Clinical. Oracle Clinical holds this test data in separate tables from production data, which lets you change test data in ways you cannot change production data.

The test data is created following a naming convention so that you can easily recognize it as test data. In some cases, Oracle Clinical automatically applies a naming convention (for example, in Responses and Discrepancies). Testers should follow a convention for all others so that they don't confuse hard copies of reports and SAS data sets with test data.

Oracle Clinical prevents using test data in Production mode and production data in Test mode. For example, when you navigate to **Design, Test a Study Design**, then **Investigators**, you can enter or select only dummy Investigators, sites, and patients; the system places any data you enter for the CRFs in tables designed to contain only test data. Conversely, if you choose **Investigators and Sites** from the **Design** menu, you can only enter or select non-test objects; the system places the data in production tables.

You can test Validation Procedures by running them against the test data. The system writes any resulting discrepancies to a special test discrepancy table. Each time you initiate a new test, the system erases the previous test discrepancy data.

For more information, see:

- [Testing Validation Procedures](#)
- [Testing Derivation Procedures](#)
- [Batch Validating Test Data](#)
- [Testing Layouts in Oracle Clinical](#)
- [Testing Layouts in Remote Data Capture \(RDC\) OnSite](#)
- [Testing in Debug Mode](#)
- [Checking Validation Results](#)
- [Testing Mass Changes](#)
- [Testing Data Extract Views](#)

Testing Validation Procedures

Testing Validation procedures consists of running each Validation procedure against your test data before making the procedure active. The system deletes any existing discrepancies each time you run a procedure.

From the Validation or Derivation procedures forms (from the **Definition** menu, select **Validation Procs** or **Derivation Procs**), you can review discrepancies without leaving the form. The menu options accessible through the **Special** menu inherit the security of the external menu items, so your access to objects is the same. The menu options are:

- **Execute in Test**
- **Show Test Discrepancy**
- **Execute in Production**
- **Show Production Discrepancy**

Testing Derivation Procedures

You test a Derivation Procedure by executing it for one or more test responses, executing the Procedure in Test mode. Oracle Clinical writes validation test results to the Test Discrepancy Database. Oracle Clinical writes test results of calculations to the Test Responses database. The system stores the test data and discrepancies separately from production data and discrepancies. You view results from the reports and Test mode data entry screens.



Note:

In V3.1-style Procedures, you must explicitly allow for the possibility of pointing to either test or production database tables. See [V3.1-Style and Pre-V3.1-Style Procedures](#) for information.

To run a Procedure in Test mode, you must first have entered or loaded data in Test mode. Follow these steps to run the procedure in Test mode:

1. Launch a single Procedure execution; from the **Definition** menu, select **Test a Study**, then **Execute Single Procedure**. The PSUB submission window appears. Other paths:

From the **Definition** menu, select **Derivation Procs**, then **Procedures**, choose **Special**, and select **Execute in Test**.

From the **Definition** menu, select **Validation Procs**, then **Procedures**, choose **Special**, and select **Execute in Test**.

2. Enter values for the **Procedure Name** and **Version Number**; both fields are list-enabled.
3. A Debug mode is available for testing and lets you view the individual values as the system derives or validates them. If you want to execute the Procedure in Debug mode, enter Y for the **Execute Procedure in debug_ mode?** parameter.

Batch Validating Test Data

Executing batch validation on test database tables tests multiple Procedures at one time and confirms both univariate settings and indicator Question specifications. The goal is to review the current state of all validation specifications as executed against all data defined in the test database tables for a study. You can document the results through the Test Discrepancy Database reports.

In Test mode, batch validation validates all existing data entered for all existing test patients, regardless of previous validation status of the data. All discrepancies are deleted at the beginning of processing and re-created by the validation session, where applicable. Similarly, all existing derived test responses are deleted and the highest compiled version of all non-retired Derivation and Validation Procedures is run on the test data. These actions derive the appropriate responses and re-create the necessary discrepancies, which you can view through the Test Discrepancy Database (from the **Definition** menu, select **Test a Study**, then **Discrepancy Database**).

Testing Layouts in Oracle Clinical

Initial Log-In is the first step in testing a DCM layout. From the **Definition** menu, select **Test a Study**, then **Test Data Entry**, and choose one of its forms.

In Test mode Oracle Clinical redefines synonyms that point to the test data tables, so you maintain all system functionality. (Additionally, you can revise First-Pass data, which you cannot access in production mode.) This is why you can use existing patient positions and CPEs without affecting the actual study database.

From the **Special** menu in the Initial Log-In and Entry form, you can select another study, enroll another patient, or change the date format. Most importantly, you can go directly to data entry by choosing **Change Task** and then selecting **First-Pass Entry** from the pop-up window. You can also reach the next form you need from the **Test a Study** menu item. Completing this form lets you begin Test Data Entry.

Each option in the **Special** menu also maps to a function key. Use the List function to determine the mappings.

Testing Layouts in Remote Data Capture (RDC) OnSite

Log in to RDC Onsite using the test-mode URL; see [Launching Test Mode Sessions](#) and enter test data for a patient to test graphic DCM layouts. In addition, generate and print a Patient Data Report and check that each page is complete.

Testing in Debug Mode

You can execute both Validation and Derivation Procedures in Debug mode, against either test or production data. In Debug mode the small-scale operations of a Procedure and the data it sees after it has run are determined. Such detail-oriented modes are often called "verbose." The only time you cannot run a Procedure in Debug mode is when it is run automatically as a part of batch validation.

Each Procedure execution results in an output file on the server. When you launch the job, a dialog box lists the name of this file. The file name format is usually `rx_c_log:xxxxx.out`, where `xxxxx` is a unique job number. To view a log file for a Procedure execution, you must log on to the server and then view the file.

In the example of a verbose Procedure execution file that follows, Procedure ZAK processes three patients, with patient numbers 101, 102, and 107. For each patient, the system retrieves the responses to the ZAK_ENROLLMENT Question for processing. Date format in this file is always YYYYMMDD format.

```
Connecting... Connected.
Single procedure execution for study PARIS, Procedure ZAK, Version 0
Execution Start: 12-JUL-1994 18:25:25
Procedure ZAK: Debug Mode Enabled.
Processing Patient: 101

Z.VISIT_NUMBER = 1
Z.DCM_DATE = 19930202
Z.REPEAT_SN = 1
Z.ZAK_ENROLLMENT = 19930101

Z.VISIT_NUMBER = 2
Z.DCM_DATE = 19930202
Z.REPEAT_SN = 1
Z.ZAK_ENROLLMENT = 19930102

Z.VISIT_NUMBER = 3
Z.DCM_DATE = 19930202
Z.REPEAT_SN = 1
Z.ZAK_ENROLLMENT = 19930103
Processing Patient: 102
Z.VISIT_NUMBER = 1
Z.DCM_DATE = 19930202
Z.REPEAT_SN = 1
Z.ZAK_ENROLLMENT = 19940104
Processing Patient: 107

Z.VISIT_NUMBER = 1
Z.DCM_DATE = 19930202
Z.REPEAT_SN = 1
Z.ZAK_ENROLLMENT = 19930123
Discrepancies: 0 new, 0 remain current, 0 became obsolete
Procedure execution completed at 12-JUL-1997 18:25:35.
```

Checking Validation Results

The last step in testing Validation Procedures is to refer to the Test Discrepancy Database, which is accessed from the **Test a Study** menu. The Maintain Discrepancy Database form opens with a query about the filter you want. You can list the available values for **Status** and **Review Status**. Most of the entries on this form are clearly defined by context or by previous entries. For obscure fields, invoke the List function.

Testing Mass Changes

Mass Changes, except you execute it through the Test Mode menu options. Oracle Clinical uses the same tables for creating the Mass Changes Criteria for Production and Test modes, but when you use the criteria to create a Candidate Data Set or

operate on the Candidate Data Set in a Test mode session, Oracle Clinical operates against Test data.

Testing Data Extract Views

To create test data extract views in Test, follow these instructions:

1. Run PSUB to create the view: Navigate to **Conduct**, then **Data Extract**, and finally **Data Extract Views**.
2. Select TEST for account type and *study_name*\$TEST for the account.
3. Click **Submit**.

Integrating with a Clinical Trial Management System

Oracle Clinical can be integrated with a clinical trial management system (CTMS) to automatically pass certain data and metadata from one system to the other. There is an existing Oracle AIA process integration pack (PIP), that integrates Oracle Clinical with Siebel Clinical; see *Oracle Study, Subject, and Visit Synchronization Integration Pack for Siebel Clinical and Oracle Clinical 3.1 Implementation Guide* at: http://download.oracle.com/docs/cd/E21374_01/doc.31/e20497.pdf.

This integration:

- Creates Oracle Clinical sites, Investigators and study sites based on Protocol site Information sent from Siebel Clinical
- Passes information about patients and visits from Oracle Clinical to Siebel Clinical
- Allows you to use the Oracle Clinical user interface to define Oracle Clinical/Remote Data Capture (RDC) data collection events that should be treated as visits/activities in Siebel Clinical and passes information o Siebel Clinical to signal the completion of visits/activities for patients

If you are using the Siebel Clinical integration, you must follow the instructions in this chapter.

You can use the same tools used to develop the Siebel Clinical integration to develop your own integration with a different system. New web services are available in Oracle Clinical to create and update sites, Investigators and study sites; see the *Oracle Clinical Application Programming Interface Guide* for further information. In addition, you must follow the instructions in this chapter to enable patient activity integration and define activities. Oracle Clinical writes information about patient visit/activity completion to the CLINICAL_STUDY_AQ activities queue in the database where a web service can pick it up and send it to another application.

For more information, see:

- [Enabling Patient and Subject Activity Integration](#)
- [Setting Up and Using Activity Completion Integration](#)

Enabling Patient and Subject Activity Integration

In this section:

- [Enabling Integration at the Database Level](#)
- [Overriding Default Integration Values for Individual Studies](#)

Enabling Integration at the Database Level

You can make two types of Oracle Clinical information available to another application: Patient Positions and Subject Activities.

You set the default values for new studies at the database level and, if required, override the default values for individual studies.

**Note:**

Setting or changing these values does not affect existing studies.

To set the default values at the database level:

1. Go to the **Admin** menu and then click **Integration Database Settings**. The Integration Database Settings window opens.
2. Click the Integration Settings node (+).
3. Enter values:
 - **Enable Patient Integration?** If set to **Y**, Patient Integration is enabled by default for new studies. Oracle Clinical writes information about a patient position to a queue, where it is available to send to other applications. For each patient position, this occurs for the first time when a patient is assigned to a patient position: when the enrollment date, informed consent signature date, or first CRF is entered for the patient position. Oracle Clinical also writes any subsequent updates for a patient position in the `patient_positions` table or `patient_statuses` table, or a change in the patient position's study site assignment, to the queue.

If set to **N**, Patient Integration for studies is not enabled by default. This is the shipped value. If set to **Y**, Patient Integration is enabled by default.

- **Enable Subject Activity Integration?** If set to **Y**, Subject Activity integration is enabled by default for new studies. You can define completion criteria in Oracle Clinical so that patient data collected in Oracle Clinical or RDC can signal visit/activity completion in a clinical trial management system (CTMS); see [Setting Up and Using Activity Completion Integration](#).

If set to **N**, Subject Activity Integration is not enabled by default. This is the shipped value. If set to **Y**, Subject Activity Integration is enabled.

**Note:**

In order to use Subject Activity Integration in the Siebel Clinical integration, you must set both values to **Y**. It is possible to set **Enable Subject Activity Integration?** to **Y** and **Enable Patient Integration?** to **N**, but the integration will not work unless both values are set to **Y**.

4. For each setting you can select **Enforce Local DB Setting** or not:
 - If selected, all new Studies are created with the value for the setting that you specify here and the value cannot be changed at the study level.
 - If unselected, all new Studies are created with the value for the setting that you specify here and the value can be changed at the study level.
5. Save.

Overriding Default Integration Values for Individual Studies

To override the default values for a particular study:

1. Go to the **Design** menu and then click **Integration Study Settings**. The Integration Database Settings window opens.
2. Click the Integration Settings node (+).
3. Enter values:
 - **Enable Patient Integration?** If set to **Y** for this study, Oracle Clinical writes information about a patient position to a queue, where it is available to send to other applications. For each patient position, this occurs for the first time when a patient is assigned to a patient position: when the enrollment date, informed consent signature date, or first CRF is entered for the patient position. Oracle Clinical also writes any subsequent updates for a patient position in the `patient_positions` table or `patient_statuses` table, or a change in the patient position's study site assignment, to the queue.

If set to **N**, Patient Integration is not enabled for the study. If set to **Y**, Patient Integration is enabled for the study.
 - **Enable Subject Activity Integration?** If set to **Y**, you can define completion criteria in Oracle Clinical so that patient data collected in Oracle Clinical can signal visit/ activity completion to a clinical trial management system (CTMS); see [Setting Up and Using Activity Completion Integration](#).

If set to **N**, Subject Activity Integration is not enabled for the study.

 **Note:**

In order to use Subject Activity Integration, you must set both values to **Y**.

4. The **Inherit from Local DB** indicates whether or not the setting is inherited from the database level:
 - If selected, the setting is inherited. If you can deselect it, the setting is not enforced and you can change the value.
 - If unselected, the setting is not inherited. You can select it if you want to be sure you are using the default setting. You can change the setting as required for your study.
5. Save.

Setting Up and Using Activity Completion Integration

In this section:

- [Creating Activity Completion Criteria](#)
- [Adding Activity Completion Criteria Details](#)
- [Running the Track Activities Batch Job](#)
- [Completion Criteria Processing](#)

Creating Activity Completion Criteria

Clinical trial management systems and other applications may have a concept of an **activity** that, when completed, triggers action of some kind; for example, a change in status. Tracking activity completion may help to monitor the progress of a trial at each site.

For each study, you specify which Oracle Clinical events constitute an activity in the other application: visits or activities within a visit, and their *completion criteria*—data that, when entered for a patient, signals the completion of the activity to the other application. Completion criteria can include one or more CRFs (DCIs), one or more CRF sections (DCMs), or one or more Questions. You can even require a specific response to a Question; for example, a response of **Yes** to the Question "Was an MRI performed?" could trigger activity completion but a response of **No** would not.

The Track Activity batch job evaluates all active completion criteria and determines if patient data entered or updated since the last time the job ran causes the completion status or date of any activity to change; see [Running the Track Activities Batch Job](#)

To define an activity, navigate to **Definition**, then **Activity Completion Criteria**. The Activity Completion Criteria Summary window opens.

The first time you define an activity and its Completion Criteria you must select **Insert Record** from the **Data** menu in the Summary window, then navigate to the Details window to define the completion criteria details. After that, you can either create new activities the same way or entirely in the Details window, using the **Add New Activity** button.

Enter values as follows:

Activity - Enter a name for the activity. You may want the name to correspond to the name of the activity in the other application.

Type - Select a value:

- **Visit**: A particular visit (Clinical Planned Event, or CPE) as a whole constitutes the activity.
- **Activity during a visit**: One or more DCIs, DCMs, or Questions constitute the activity. You can either define the activity as occurring only during a particular visit, or each time the specified DCIs, DCMs, or Questions are collected for a patient at any visit.

Status - Select a value:

- **P** (Provisional) This is the default value. You can modify the Activity Completion Criteria only if its status is Provisional.
- **A** (Active) The Track Activities job processes only active Completion Criteria. You can set the status to Active if:
 - At least one DCI, DCM or DCM Question have been specified in the Details section and there is a value for both Completion Date Level and Completion Date Source.
 - If Criteria Level is set to **DCI**, all specified DCIs are Active.
 - If Criteria Level is set to **DCM**, all specified DCMs are Active.

- If Criteria Level is set to **DCM Question**, all specified DCMs that include the DCM Question are Active.

 **Note:**

If you set an Activity Completion Criteria definition to Active, use it, and then set its status back to Provisional to modify it:

- * Activities already completed are not affected.
- * Patient data entered while the status is Provisional does not trigger activity completion. However, when you set the status back to Active, the next time the Track Activity batch job runs it re-evaluates all patient data against this criteria and triggers activity completion for patients as appropriate.

- **R (Retired)** If you no longer want to use an Activity Completion Criteria, set its status to Retired and specify a Retirement Reason. The Track Activity batch job no longer evaluates patient data against it. Changing the status to Retired does not affect patients' previously completed activities.

Visit Specific Criteria - (Available only in the Details window.) Select this option if your activity is of type Visit, or if your activity is of type Activity Within a Visit and you want to specify a particular visit during which the completion criteria—DCI(s), DCM(s) or Questions—must be collected to trigger activity completion.

If the activity is of type Activity Within a Visit and you do not specify a particular visit, then each time the DCIs, DCMs, or Questions are collected for a patient in any visit, the activity is considered complete.

DCI Book - (Optional) If you are using more than one DCI Book, select the relevant DCI Book. The system then allows you to select only DCIs, DCMs, and DCM Questions that are associated with the visit you specify, in the DCI Book you specify.

The list of DCI Books includes all DCI Books defined as either Active or Provisional for the current study.

Clinical Planned Event - (CPE): If you want the activity to be considered complete if the specified DCI(s), DCM(s), or Question(s) are collected at a particular visit, select the name of the visit (CPE).

Completion Date Level - Oracle Clinical sends the activity completion date to the other application. You may need to collect the completion date for different activities in a different manner, according to the requirements of the study and your business practices.

Select one value:

- Select **DCI** to use a DCI header date as the completion date. Under Completion Date Source, you can then specify using the earliest or latest DCI (in case they are on different days) or a particular DCI.
- Select **DCM** to use a DCM header date as the completion date. Under Completion Date Source, you can then specify using the earliest or latest DCM (in case they are on different days) or a particular DCM.
- Select **DCM Question** if you have a DATE-type Question on the CRF for the purpose of collecting the date.

Completion Date Source - If the Completion Date Level is DCI or DCM, choose how to determine the completion date at the specified level (in case the specified DCIs or DCMs are collected over more than one day):

- **Visit Start Date:** The system uses the earliest visit date of all DCIs or DCMs listed in the Details section.
- **Visit End Date:** The system uses the most recent visit date of all DCIs or DCMs listed in the Details section.
- **Specific Date:** Rather than using the earliest or latest, you can specify a particular DCI or DCM whose value you want to use. If you choose Specific Date here, you must define the Criteria Level to be the same as the Completion Date Level: DCI, DCM, or DCM Question, and in the Criteria Details section you must specify the DCI, DCM, or Question you want to use.

If the Completion Date Level is DCM Question, Completion Date Source defaults to Specific Date and cannot be changed. In the Criteria Details section of the window you must check **Use as Specific Date** next to the DCM Question of data type DATE defined for the purpose of collecting the date.

Criteria Level - Select the level of data—DCI, DCM, or DCM Question—that should be collected for a patient to signal the completion of the activity. If you selected Specific Date for the Completion Date Source, you must set Criteria Level to the same value as the Completion Date Level.

- **DCI:** The system considers the activity complete when one or more DCIs are collected for the patient. You specify which DCIs under Criteria Details.
- **DCM:** The system considers the activity complete when one or more DCMs are collected for the patient. You specify which DCMs under Criteria Details.
- **DCM Question:** The system considers the activity complete when one or more DCM Questions in one or more DCMs are collected for the patient. You specify which DCM Questions under Criteria Details. The DCM Questions cannot be part of a DCM repeating Question Group.

You can require a particular response to one or more Questions as the completion criterion. For example, if you want the activity to be considered complete only if the Question "Was an MRI performed?" has a response value of Yes, select **DCM Question** here, and in the Details section select the name of the Question in the **DCM Question** field and enter **Yes** in the **Response Value** field in the Details section.

No. of Criteria - (Available only in the Summary window.) The system provides high level information on how many Criteria Details must be completed for the activity to be considered complete.

Retirement Reason - If you change the status to Retired, the system prompts you to enter a reason, which is then displayed in this field.

If you are working in the Summary window, you must save your work before clicking the **Details** button to complete the Activity Completion Criteria definition. See [Adding Activity Completion Criteria Details](#) .

Adding Activity Completion Criteria Details

Select the particular DCI(s), DCM(s), or DCM Question(s) (depending on the Criteria Level you selected) required as completion criteria. You can do this in two ways:

- Add criteria one at a time, using the list of values in the appropriate field.
- Add more than one criterion at a time using the Add Multiple option. This is especially helpful if your Criteria Level is DCM or DCM Question since it allows you to narrow your search by first selecting the DCI, then the DCM, and then, if the Criteria Level is DCM Question, the Question(s).

If you selected a DCI Book, the system limits your choices to DCIs, DCMs, or DCM Questions associated with the visit in that DCI Book. Otherwise, the system allows you to select any DCI, DCM, or DCM Question included in the study.

The Criteria Level you specified determines which of the following fields are available:

Use as Specific Date - If you defined the Completion Date Source as **Use Specific Date**, select this box for the DCI, DCM, or DCM Question whose date you want to use.

DCI Name - If the Criteria Level is DCI you must select one or more DCIs that must be collected for the activity to be considered complete. If you did not specify a visit (CPE), the system considers the activity complete each time the specified DCIs are collected for a patient. For each DCI, the system displays the DCI name, type, and status in the list of values or after you click **Add Multiple**.

DCM Name - If the Criteria Level is DCM you must specify one or more DCMs that must be collected for the activity to be considered complete. If you did not specify a visit, the system considers the activity complete each time the specified DCMs are collected for a patient. For each DCM, the system displays the DCM name, subset, qualifying value (if any), type, and description in the list of values or after you click **Add Multiple**. Using **Add Multiple**, you can drill down to the DCM from its DCI.

If the Criteria Level is DCM Question, select each DCM Question's DCM.

DCM Subset # - When you select a DCM, you simultaneously select one of its subsets. The system enters the subset number in the next field as well as the name. You can select a different valid subset of the same DCM if there is one.

Qualifying Value - If the same DCM exists in the visit more than once, you must select the qualifying value of the DCM(s) that must be collected for the activity to be considered complete. You must select a DCM name before you can enter a value in this field.

DCM Question - If the Criteria Level is DCM Question, you must specify one or more DCM Questions. After you have selected the DCM (including the subset and qualifying value, if any) that contains the Question, select the Question Group and then the Question itself. Question Groups are displayed in the order in which they are defined in the DCM.

If you did not specify a visit, the system considers the activity complete each time the specified DCM Question(s) are collected for a patient in the context of the same DCM subset and qualifying value.

Occ # - Select the Occurrence number of the DCM Question in the Question Group. If the Question is not included in the Question Group more than once, the value defaults to zero (0).

Response Value - If a particular Question response is required to trigger the completion of the activity, enter it here. If the Question has a DVG, there is a list of values.

Running the Track Activities Batch Job

You must schedule the Track Activity batch job to run periodically to evaluate all active completion criteria and determine if patient data entered or updated since the last time the job ran causes the completion status or date of any Activity to change.

If new or modified Activity Completion Criteria have been set to Active since the last time the job was run, the job examines all existing patient data for the study. See [Completion Criteria Processing](#) for information on how the job works.

To set up the Track Activity process:

1. Navigate to **Conduct**, then **Data Validation**, then **Track Activities**. The system displays the Track Activity Completion window.
2. If your study is not already in context, select it using either the Change Study button or the Special menu.
3. The Track Activity process has no parameters. Submit or schedule the job.



Note:

The study must be enabled for Activity integration; see [Enabling Patient and Subject Activity Integration](#).

Completion Criteria Processing

In this section:

- [Completeness Logic](#)
- [Completion Date Updates](#)

Completeness Logic

In addition to the collection of the DCI(s), DCM(s), or DCM Question(s) defined as Activity Completion Criteria, activity completion logic requires that the Received DCI, Received DCM, or the Received DCM that contains the required DCM Question(s) meet the following criteria:

- A received DCI or DCM that is marked as blank is not considered complete.
- If the study requires two passes, only an RDCI or RDCM with the status PASS 2 COMPLETE or BATCH LOADED is considered complete.
- If the study does not require two passes, only a Received DCI or DCM with the status PASS 1 COMPLETE, PASS 2 STARTED, PASS 2 PENDING, PASS 2 COMPLETE or BATCH LOADED is considered complete.

Completion Date Updates

If the completion date for an activity changes, Oracle Clinical writes a message to the queue updating the completion date for the activity for the patient. The completion date can change if:

- The source date value changes, in which case Oracle Clinical passes the new date to the queue.
- The activity is no longer considered complete, in which case Oracle Clinical passes a blank date to the queue.

Activities previously considered complete may become incomplete due to the following kinds of changes:

- If an RDCI or RDCM that was mapped to an activity is deleted, the activity is no longer considered complete.
- If the blank flag is set to Y for an RDCI or RDCM that was mapped to a completed Activity, the Activity is no longer considered complete.
- If a response value for a DCM Question that was mapped to an Activity is modified to a value other than the one defined in the mapping, the Activity is no longer considered complete.
- If a response value for a DCM Question that was mapped to an Activity is removed leaving the response empty, the Activity is no longer considered complete.
- Key changes: Changes to the following header (key) information for a received DCI or DCM make an Activity previously considered complete become incomplete.
 - Patient
 - DCI Book
 - Visit Number

A

Flexible Study Design Examples

This section contains examples of defining rules in an enhanced DCI Book to support a flexible trial protocol.

For information about flexible studies, see [Designing a Flexible Study](#) and [Defining an Enhanced DCI Book](#).

For more information, see:

- [Basic Examples](#)
- [Complex Examples](#)

Basic Examples

Following are examples of how to set up a flexible study in Oracle Clinical by defining DCI Book rules. In the diagrams:

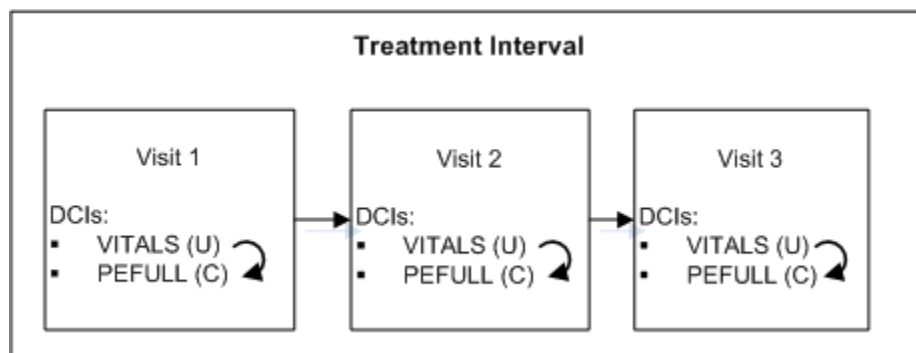
- Only the setup directly relevant to the example is described. In a real trial, all Intervals would have at least one CPE and all CPEs would have at least one DCI, for example. In addition, each DCI must contain at least one DCM, which must contain at least one Question Group, which in turn contains at least one Question.
- (U) means unconditional and (C) means conditional. By definition, a conditional DCI or Interval is the target of a rule, and an unconditional DCI or Interval is not.

For more information, see:

- [Enabling a DCI within a Clinical Planned Event](#)
- [Enabling DCIs Across Clinical Planned Events](#)
- [Enabling Intervals Based on the Collection of Any Data](#)
- [Enabling Intervals Based on a Data Value](#)
- [Skipping to an Interval](#)
- [Enabling the Next Interval](#)

Enabling a DCI within a Clinical Planned Event

During the treatment phase of a trial, a physical exam is required at each Clinical Planned Event if and only if the patient's vitals have changed since the last Clinical Planned Event.

Figure A-1 Conditional DCI within a Clinical Planned Event

To implement this design in Oracle Clinical:

- Define two DCIs: VITALS and PEFULL and assign them to every CPE in the Treatment Interval.
- At the end of the VITALS DCI, define Question PE_CHANGE asking "Have there been any changes to vitals since the last visit?" with a Yes/No Discrete Value Group.
- Define a DCI rule of type "Enable DCI Within CPE" with PE_CHANGE as the trigger Question, Yes as the trigger value, and PEFULL as the target DCI.

At each Clinical Planned Event, the response to the PE_CHANGE Question will determine whether the PEFULL DCI is required for the patient; if Yes, the exam is expected and RDC Onsite displays the corresponding CRF for the patient. If No is entered, it is not expected.



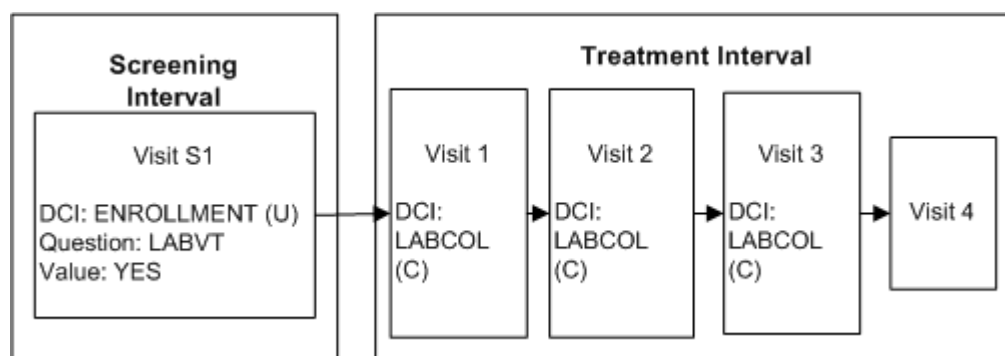
Note:

One rule covers every case where the VITALS and PEFULL DCIs occur in the same CPE.

Enabling DCIs Across Clinical Planned Events

A trial has a substudy that requires an extra blood draw at Visits 1, 2, and 3 of the Treatment phase. Not all patients should participate in the substudy.

Figure A-2 Conditional DCI Across Clinical Planned Events



To implement this design in Oracle Clinical:

- Define DCI ENROLLMENT and assign it to Visit S1 in the Screening Interval.
- Define Question LABVT in DCI ENROLLMENT with a Yes/No Discrete Value Group.
- Define DCI LABCOL to request the blood draw and record the lab results, and assign it to Visit T1 in the Treatment Interval.
- Define a DCI rule of type "Enable DCI Across CPEs" with LABVT as the trigger Question, Yes as the trigger value, and LABCOLL as the target DCI.

If a value of Yes is entered for Question LABVT, DCI LABCOLL becomes expected for the patient at every CPE to which it is assigned. RDC Onsite displays the corresponding CRF in visits that are currently expected. If a value of No is entered, DCI LABCOLL is never expected for the patient.

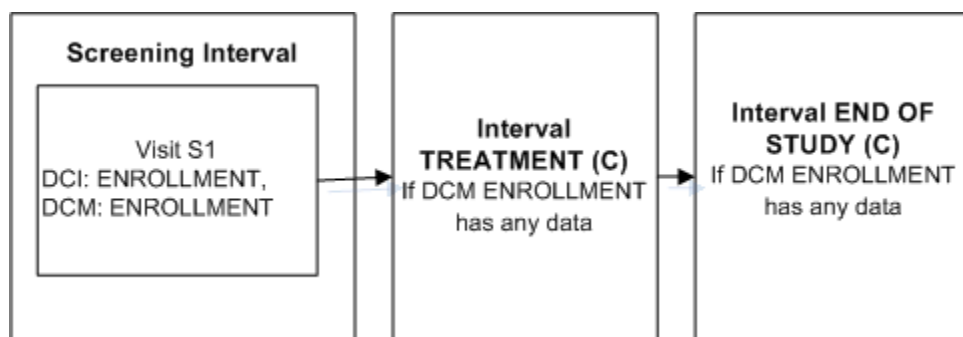
Note:

One rule enables DCI LABCOLL in every CPE to which it is assigned. If you want the same blood test to be performed but without being conditional on the response to Question LABVT, copy the DCI, rename it, and assign the renamed DCI to the appropriate CPEs. If you want it to be conditional upon a different Question response, define another rule. If you want it to be performed on all patients unconditionally, do not define any rule with it as target.

Enabling Intervals Based on the Collection of Any Data

All patients in a trial have the same visit schedule expected, going from the screening Interval to the treatment Interval, to the end-of-study Interval.

Figure A-3 Enable Intervals Based on Any Data



To implement this design in Oracle Clinical:

- Define DCI ENROLLMENT that includes DCM ENROLLMENT and assign it to Visit S1 in the Screening Interval.
- Define Intervals TREATMENT and an END OF STUDY.
- Define an Interval rule with action "Enable" with DCM ENROLLMENT as the trigger and [Any Data] as the trigger value, and Intervals TREATMENT and END OF STUDY as the target.

As soon as an operator enters any data for a Question in DCM ENROLLMENT and saves, Intervals TREATMENT and END OF STUDY and all their CPEs and unconditional DCIs become expected for the patient. RDC Onsite displays all corresponding expected CRFs and all visits that contain expected CRFs.



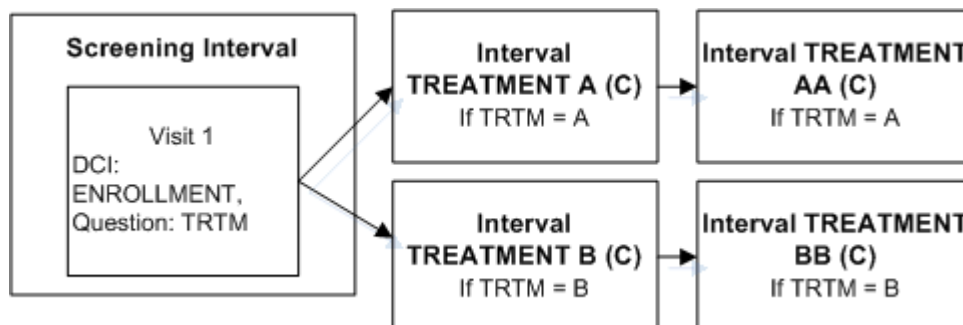
Note:

An Interval rule with action "Enable" can make one or more Intervals expected for a patient.

Enabling Intervals Based on a Data Value

Patients in a trial must be assigned to one of two treatment arms depending on their response to a Question collected in the Enrollment DCI at the first visit.


Figure A-4 Enable Intervals Based on a Data Value



To implement this design in Oracle Clinical:

- Define DCI ENROLLMENT and assign it to Visit S1 in the Screening Interval.
- Define Question TRTM in DCI ENROLLMENT with a Discrete Value Group that includes values A and B.
- Define two Intervals, TREATMENT A and TREATMENT B.
- Define an Interval rule with action "Enable" with TRTM as the trigger Question, A as the trigger value, and Intervals TREATMENT A and TREATMENT AA as the target.
- Define an Interval rule with action "Enable" with TRTM as the trigger Question, B as the trigger value, and Intervals TREATMENT B TREATMENT BB as the target.

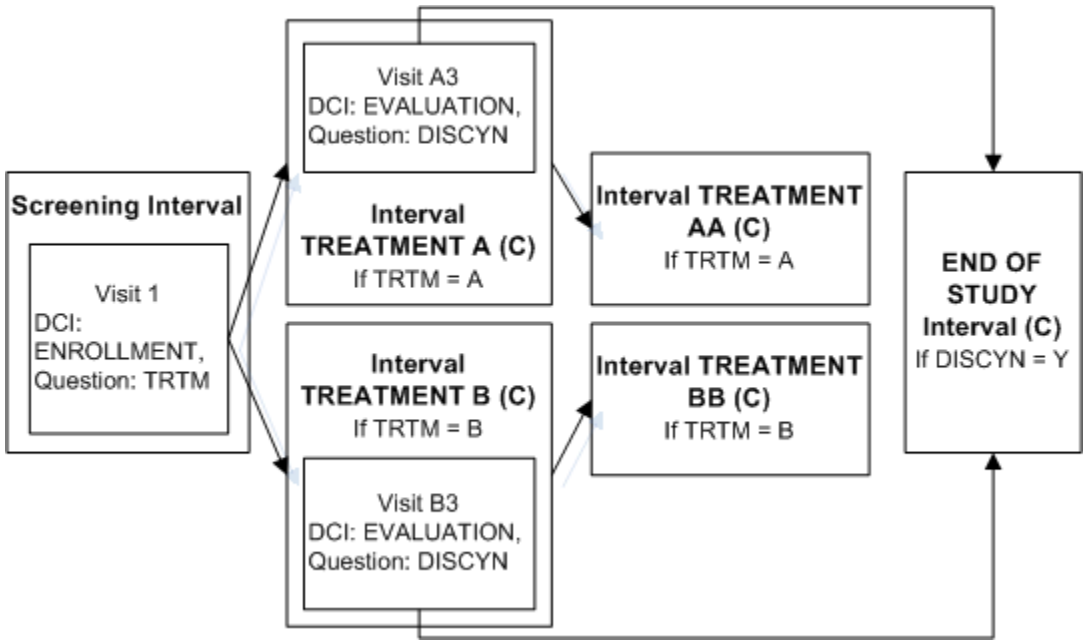
If a value of A is entered for Question TRTM, Interval TREATMENT A and all its CPEs and unconditional DCIs become expected for the patient. If a value of B is entered for Question TRTM, Interval TREATMENT B and all its CPEs and unconditional DCIs become expected for the patient. RDC Onsite displays all corresponding expected CRFs and all visits that contain expected CRFs.

 **Note:**
An Interval rule with action "Enable" can make one or more Intervals expected for a patient.

Skipping to an Interval

In the same trial, patients in both treatment arms may need to exit the study after the first treatment Interval. An evaluation DCI must be collected ending with a summary Question to collect the decision to discontinue the patient from the study or not.

Figure A-5 Bypass To Interval



To implement this design in Oracle Clinical:

- Define the Screening Interval and Intervals TREATMENT A, TREATMENT AA, TREATMENT B, TREATMENT BB as in the example [Enabling Intervals Based on a Data Value](#).
- Define Interval END OF STUDY.
- Define DCI EVALUATION and assign to Visit A3 in Interval TREATMENT A and Visit B3 in TREATMENT B.
- Define Question DISCYN in DCI Evaluation with a Yes/No Discrete Value Group assigned.
- Define an Interval rule with action "Bypass To" with DISCYN as the trigger Question, Yes as the trigger value, and Interval END OF STUDY as the target.

If a value of Yes is entered for Question DISCNY, Interval TREATMENT AA is no longer expected for a patient in TREATMENT A; or TREATMENT BB is no longer expected a patient in Interval TREATMENT B. In both cases Interval END OF STUDY becomes the next expected Interval. RDC Onsite no longer displays either TREATMENT AA or TREATMENT BB but does display all CRFs corresponding to the unconditional DCIs in Interval END OF STUDY and the Visit(s) that contain them.

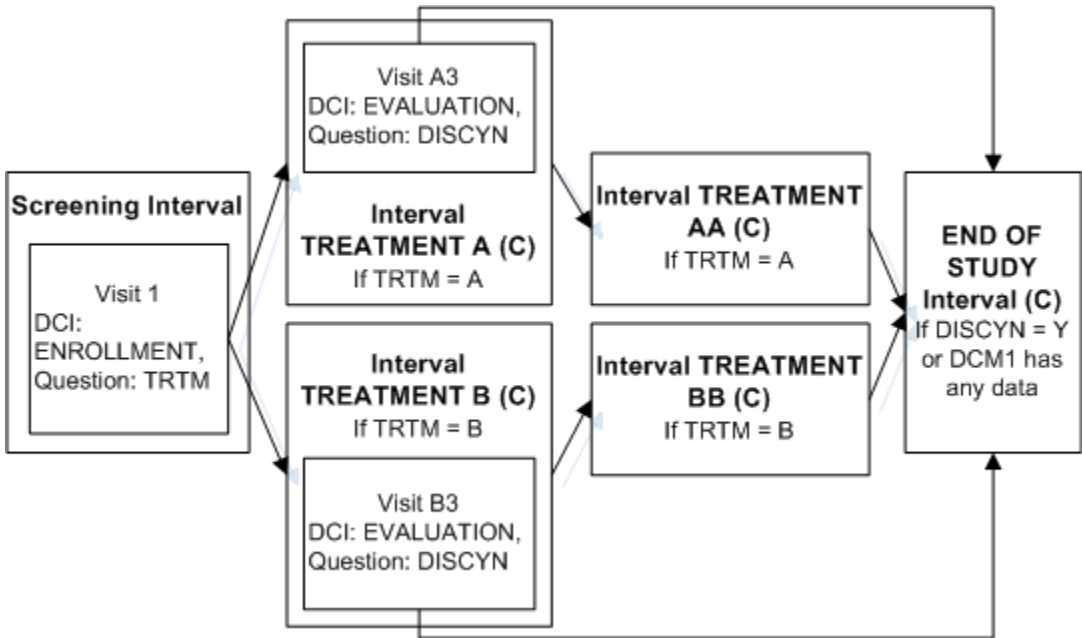
If a value of No is entered for Question DISCNY, there is no change. The patient completing TREATMENT A proceeds to TREATMENT AA and the patient completing TREATMENT B proceeds to TREATMENT BB. RDC Onsite continues to display all CRFs corresponding to the unconditional DCIs in those Intervals and the Visit(s) that contain them.

 **Note:**

In this trial, you can use any of several methods to ensure that patients who continue to TREATMENT AA or BB also complete the END OF STUDY Interval. For example, define an "Enable" Interval rule with [Any Data] in the first DCM scheduled for the first visit of TREATMENT AA as the trigger and Interval END OF STUDY as the target. Define a similar Interval rule for TREATMENT BB. The diagram then looks like [Figure A-6](#).

An Interval can be the target of two rules if one rule has the action "Enable" and the other has the action "Bypass To."

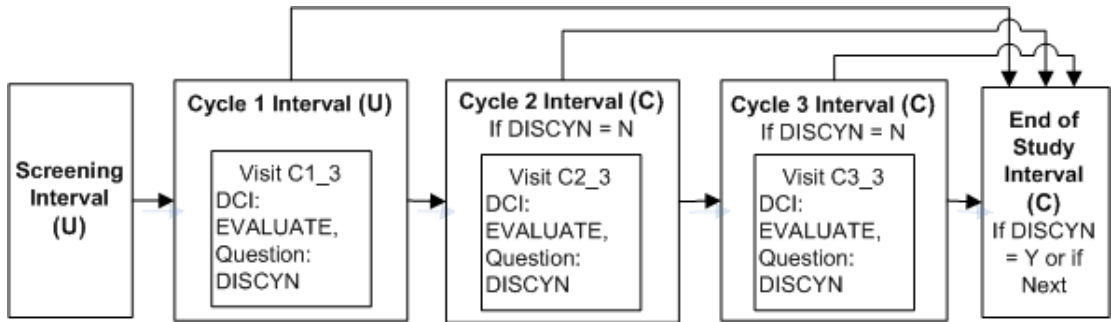
Figure A-6 Enabling and Bypassing To the Same Interval



Enabling the Next Interval

A trial requires a screening Interval followed by three cycle Intervals. The cycle Intervals are exactly the same, but each patient will move to the next cycle only if he or she passes an evaluation in the current cycle and the Discontinue? Question has No entered as a response. When the patient completes all three cycle Intervals or fails the evaluation he or she proceeds to the End of Study Interval.

Figure A-7 Enable Next Interval



To implement this design in Oracle Clinical:

- Define the Intervals SCREENING, END OF STUDY, and Interval CYCLE 1.
- Define DCI EVALUATE and assign to Visit C3 in Interval CYCLE 1.
- Define Question DISCYN in DCI EVALUATE with a Yes/No Discrete Value Group assigned.

- Define Intervals CYCLE 1, CYCLE 2, and CYCLE 3. Add their CPEs to the DCI Book, add DCIs to CYCLE 1, then copy those pages to the CPEs in the other cycle Intervals; see [Copying Pages for a CPE](#).
- Define an Interval rule with action "Enable" with DISCYN as the trigger Question, No as the trigger value, and [Next Interval] as the target.
- Define an Interval rule with action "Bypass To:" with DISCYN as the trigger Question, Yes as the trigger value, and Interval END OF STUDY as the target.

If a value of No is entered for Question DISCYN, the next CYCLEx Interval becomes expected for the patient. If a value of Yes is entered for DISCYN, the END OF STUDY Interval becomes expected for the patient. Until a value is entered for DISCYN, neither the next CYCLE Interval nor the END OF STUDY Interval is displayed in RDC Onsite.

 **Note:**

You must be careful to define more CYCLE Intervals than you will need. If not, every patient in the last defined CYCLE Interval proceeds to END OF STUDY because it is the Next Interval and therefore the target of both the Enable and the Bypass To rules.

If you need to add cycles after entering data, you must define additional Intervals in the study schedule, then set the Book to Provisional and add DCI Book Pages to the Intervals' CPEs, and set the Book's status back to Active.

Complex Examples

The following examples show how multiple rules interact:

- [Defining Multiple Paths Through a Trial](#)
- [Enabling DCIs in a Conditional Interval](#)
- [Defining Multiple Conditional Pretreatment Intervals and DCIs](#)

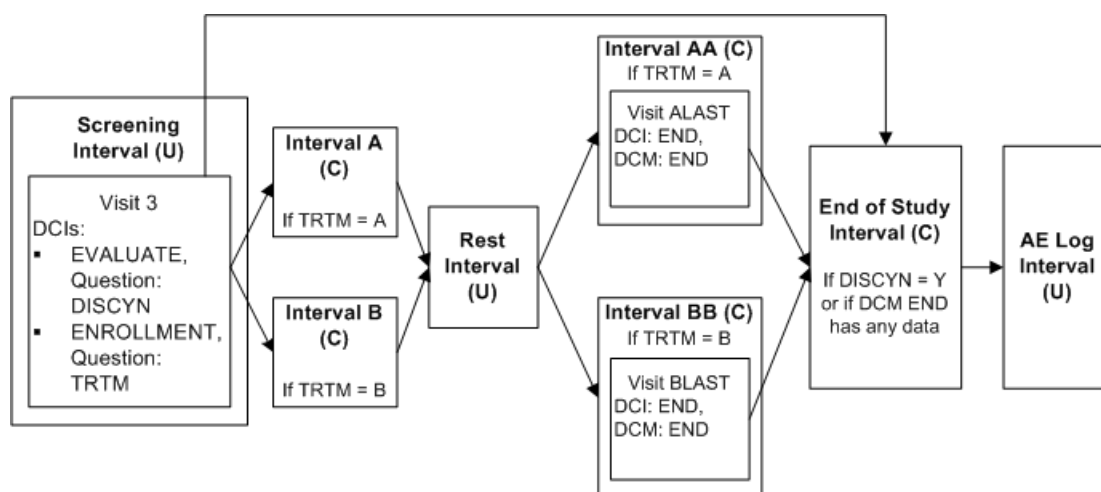
Defining Multiple Paths Through a Trial

This trial includes two treatment paths:

- Screening Interval, Interval A, Rest Interval, Interval AA, End of Study Interval, AE Log Interval
- Screening Interval, Interval B, Rest Interval, Interval BB, End of Study Interval, AE Log Interval

Both paths share the Screening, Rest, End of Study Intervals, and AE Log. In addition, if certain criteria are not met the path goes directly from the Screening Interval to the End of Study Interval.

Figure A-8 Multiple Paths



To implement this design in Oracle Clinical:

- Define Intervals Screening, A, B, Rest, AA, BB, End of Study and AE Log in this order in the study schedule.
- Define DCI EVALUATE with Question DISCYN with a DVG with values Yes and No, and assign it to Visit 3 in Screening Interval.
- Define DCI ENROLLMENT with Question TRTM with a DVG with values A and B and assign it also to Visit 3 in Screening Interval.
- Define DCI END with DCM END and assign it to Visit ALAST in Interval AA and Visit BLAST in Interval BB.
- Define Interval rule 1 with action "Enable" as follows:
 - Trigger: DCI ENROLLMENT, Question TRTM, value A, CPE Visit 3
 - Target: Interval A, Interval AA
- Define Interval rule 2 with action "Enable" as follows:
 - Trigger: DCI ENROLLMENT, Question TRTM, value B, CPE Visit 3
 - Target: Interval B, Interval BB
- Define Interval rule 3 with action "Enable" as follows:
 - Trigger: DCI END, DCM END [Any Data], (Question null, value null), CPE ALAST, BLAST
 - Target: Interval End of Study

RDC Onsite displays the Screening, Rest, and AE Log Intervals and all their unconditional DCIs and the CPEs that contain them, for each patient assigned to the DCI Book, before any data is entered. If you prefer not to display the Rest and AE Log Intervals until they are the next expected Interval, you can define a rule with a trigger in the preceding Interval and a target of [Next Interval].

If the response to DISCYN is N and the response to TRTM is A, all Intervals in path A become expected for the patient and RDC Onsite displays all the unconditional DCIs in Intervals A and AA and the Visit(s) that contain them. If the response DISCYN is N and the

response to TRTM is B, the system does the same for path B. In both cases, the End of Study Interval becomes expected for the patient when DCM END has any saved data.

However, if the response to DISCYN is Y, the patient proceeds immediately to the End of Study Interval even a value of A or B is entered for TRTM. None of the other Intervals is either expected or displayed in RDC Onsite.

 **Note:**

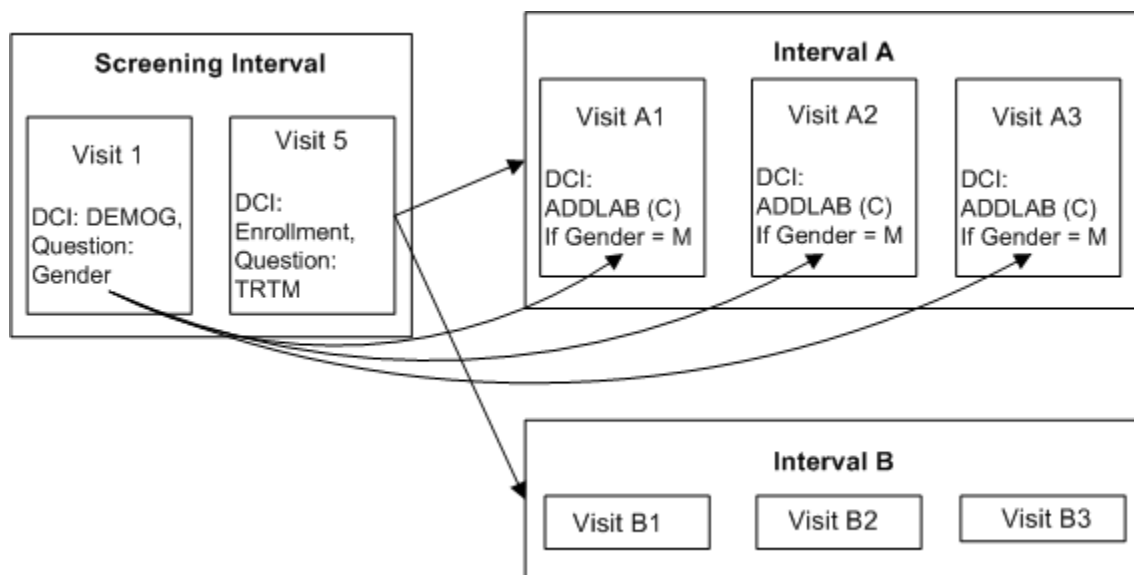
You can define an Enable Interval rule triggered in Interval AA and BB even though a Bypass rule exists that bypasses Intervals AA and BB when triggered.

The Rest and AE Log Intervals are automatically expected for patients on both paths because they are defined in that order in the study schedule and are not the target of any rule.

Enabling DCIs in a Conditional Interval

This trial has two treatment paths. In path A only, male patients are required to have an extra blood draw. Both male and female patients are assigned to both treatment paths.

Figure A-9 Conditional DCIs in a Conditional Interval



To implement this design in Oracle Clinical:

- Define Intervals Screening, A and B.
- Define DCI DEMOG with Question GENDER with a DVG with values M and F, and assign it to Visit 1 in the Screening Interval.

- Define DCI ENROLLMENT with Question TRTM with a DVG with values A and B, and assign it to Visit 5 of the Screening Interval.
- Define DCI ADDLAB and assign it to Visits A1, A2, and A3 in Interval A.
- Define Interval Rule 1 with action "Enable" as follows:
 - Trigger: DCI ENROLLMENT, Question TRTM, value A, CPE Visit 5
 - Target: Interval A
- Define Interval Rule 2 with action "Enable" as follows:
 - Trigger: DCI ENROLLMENT, Question TRTM, value B, CPE Visit 5
 - Target: Interval B
- Define DCI Rule 3 of type "Enable Across CPEs" as follows:
 - Trigger: DCI DEMOG, Question Gender, value M, CPE Visit 1
 - Target: DCI ADDLAB

At Visit 1, the response to the trigger Question GENDER is entered. However, no DCIs become expected for the patient even if the trigger response (M) is entered because the patient is not yet assigned to either Interval A or B. After the response to Question TRTM is entered for Visit 5 either Interval A or B becomes expected for the patient. If TRTM's value is A and GENDER's value is M, then DCI ADDLAB becomes expected in Visits A1, A2, and A3 in Interval A. Only then does RDC Onsite display both Interval A with all its unconditional DCIs and the CPEs that contain them, and DCI ADDLAB for male patients.

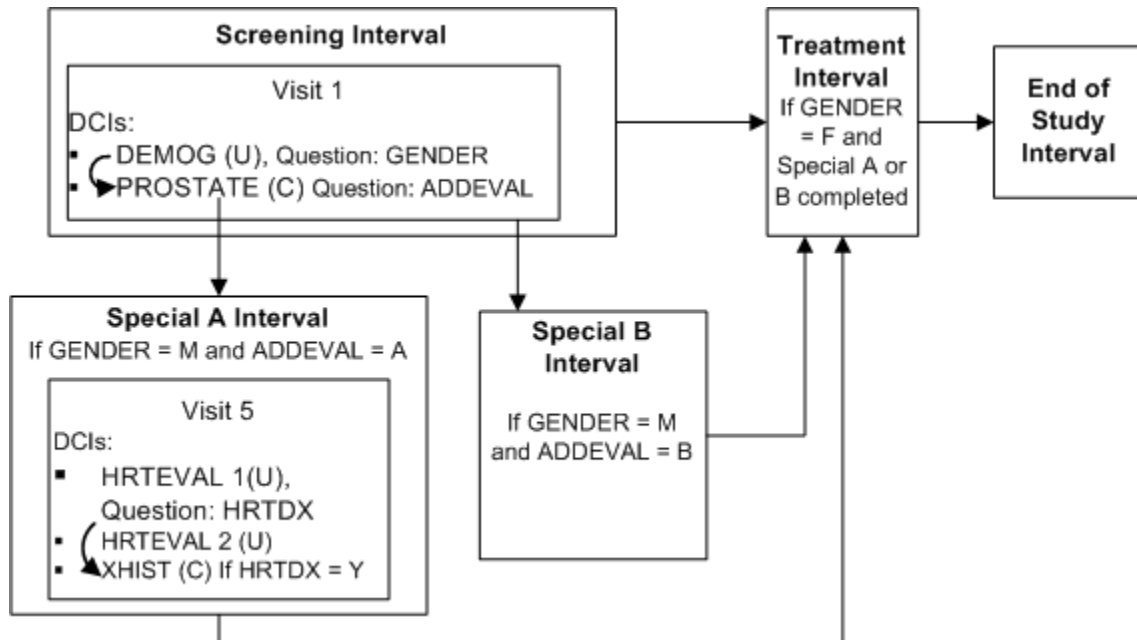
 **Note:**

Target DCIs can be contained in multiple CPEs in an Interval that is made expected by an Interval rule triggered by a DCI in a CPE later than the CPE that contains the DCI rule trigger.

Defining Multiple Conditional Pretreatment Intervals and DCIs

This trial has a single treatment path that includes a Screening, Treatment, and End of Study Interval. In addition, all male patients must complete a PROSTATE DCI during the Screening Interval and one of two additional Intervals (SPECIAL A or SPECIAL B) based on a response in the PROSTATE DCI. Patients in SPECIAL A must also complete a XHIST DCI if they have a history of heart disease.

Figure A-10 Multiple Conditional Pretreatment Intervals and DCIs



To implement this design in Oracle Clinical:

- Define Intervals Screening, Special A, Special B, Treatment, and End of Study in that order in the study schedule
- Define DCI DEMOG DCI with Question GENDER with a DVG with values M and F, and assign it to Visit 1 in the Screening Interval
- Define DCI PROSTATE with Question ADDEVAL with a DVG with values A and B, and assign it to Visit 1 in the Screening Interval
- Define DCI HRTEVAL1 with Question HRTDX with a DVG with values Y and N, and assign it to Visit 5 in Interval Special A
- Define DCI XHIST and assign it to Visit 5 in Interval Special A
- Define Interval Rule 1 with action "Enable" as follows:
 - Trigger: DCI DEMOG, Question GENDER, value F, CPE Visit 1
 - Target: Treatment Interval
- Define DCI Rule 2 of type "Enable Within CPE" as follows:
 - Trigger: DCI DEMOG, Question Gender, value M, CPE Visit 1
 - Target: DCI: PROSTATE
- Define DCI Rule 3 of type "Enable Within CPE" as follows:
 - Trigger: DCI HRTEVAL 1, Question HRTDX, value Y, CPE Visit 5
 - Target: DCI: XHIST

When a value of M is entered for Question Gender at Visit 1, the PROSTATE DCI becomes expected for the patient and must be collected at the same visit. If the value of Question ADDEVAL in the PROSTATE DCI is A, Interval Special A becomes expected for the patient. If the value of ADDEVAL is B, Interval Special B becomes

expected. Patients in Interval Special A must complete the HRTEVAL DCI, and if their response to Question HRTDX is Y, they must also complete DCI XHIST.

Male patients proceed to the Treatment Interval after completing either Interval Special A or Interval Special B because it is defined next in the study schedule. Female patients proceed to the Treatment Interval immediately after completing the Screening Interval. The Screening and End of Study Intervals are expected and displayed in RDC Onsite for all patients from the beginning of the trial.

RDC Onsite displays Intervals Special A and Special B and the conditional DCIs within them only when they become expected for each patient.

Index