ORACLE

CLINTRIAL™

Admin and Design

release 4.7.1

Part Number: E27579-01

Copyright © 2003-2012, Oracle and/or its affiliates. All rights reserved.

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

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

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software -- Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This documentation may include references to materials, offerings, or products that were previously offered by Phase Forward Inc. Certain materials, offerings, services, or products may no longer be offered or provided. Oracle and its affiliates cannot be held responsible for any such references should they appear in the text provided.

Contents

Preface

Overview

Clintrial™ 4 software (hereafter referred to as Clintrial software) is a comprehensive clinical research system for the collection, management, and review of clinical trials data. Clintrial software is designed for use by companies that must both:

- Collect clinical data to meet regulatory requirements for conducting clinical trials.
- Analyze data that is collected during those clinical trials.

Clintrial software enables companies to unify all of their clinical data collection and management, regardless of source or phase of development (pre- or post-market).

About this book

This book is written for all Clintrial software users. It explains Clintrial software concepts and describes the tasks you can perform with Clintrial software. Other chapters cover product installation, and setup of the Sample Studies.

About the Clintrial software documentation

The Clintrial software documentation includes books that contain conceptual information. The Clintrial software Help contains procedures for the tasks that you perform with the Clintrial software.

The Clintrial software documentation assumes that you know how to perform basic tasks on your computer.

What are the Clintrial software books?

The Clintrial 4.7 documentation includes the documents in the following table. All documentation is available from the Phase Forward Download Center.

Title:	Content:
Release Notes	The *Release Notes* document describes enhancements introduced and problems fixed in the current release, upgrade considerations, release history, and other late-breaking information.
Known Issues	The *Known Issues* document provides detailed information about the known issues in this release, along with workarounds, if available. **Note:** The most current list of known issues is available on the Phase Forward Extranet. To sign in to the Extranet, go to https://extranet.phaseforward.com and click **Customer Login**. Enter your email address and password, and navigate to the **Known Issues** section. Select a product, and then enter your search criteria.
Getting Started	The *Getting Started* guide: • Provides a summary of each Clintrial module, a description of the relationships between modules, and descriptions of key concepts. • Describes how to install, upgrade, and de-install the Clintrial software. • Describes how to configure the Clintrial application. • Provides information and procedures for customizing the Windows Registry. • Explains how to use the Medika Sample Studies.
Admin and Design	The *Admin and Design* document describes how to use: • The Admin module to work with user accounts, access rights, parameters, and system administration tools. • The Design module to set up and maintain Clintrial application objects, such as protocols, panels, and study books.
Secure Configuration Guide	The *Secure Configuration Guide* provides an overview of the security features provided with the Clintrial application including details about the general principles of application security, as well as how to install, configure, and use the Clintrial application securely.

Title:	Content:
Reference Guide	The *Reference Guide* provides: • Definitions of the Oracle database tables that store Clintrial metadata and clinical data. • Descriptions of the use of PL/SQL for Clintrial-specific procedures. • Explanations of data types and naming conventions. • Information on using SQL, setting up custom menus, and running batch jobs. • A glossary of terms.
Manage, Classify, and Lab Loader	The *Manage, Classify, and Lab Loader* document describes how to use: • The Manage module to perform data management tasks such as coding (including integration with Central Coding), global modification, validation, auditing, and batch loading of clinical data. • The Classify module to track, review and solve for values that fail automatic coding; to audit the contents of a coding thesaurus protocol; and to build and test effective thesaurus algorithms. • The Lab Loader module to batch load laboratory data and to set up Lab Loader objects.
Enter, Resolve, and Retrieve	The *Enter, Resolve, and Retrieve* document describes how to use: • The Enter module to enroll subjects, enter and edit data, verify data, and work with reports. • The Resolve module to identify, track, and report data discrepancies, as well as how to customize the Resolve module, including writing rules that reference data items. • The Retrieve module to extract clinical data from the database and work with query results.
Multisite	The *Multisite* document describes: • How to distribute codelists and protocols. • How to set up a replication environment. • How other Clintrial modules work differently in a Multisite environment.
Quick Reference Card for Enter	The *Quick Reference for Enter* lists Enter module menu commands and shortcut keys.

Conventions

The following conventions are used in the Clintrial software books:

Convention:	**Description:**
Italics	Italics are used to indicate the following: • New terms • Titles of books • Variable names in code examples or file names
Ctrl + c	Key combinations where you press the first key and hold it down while you press the second key. For example, to copy selected text to the clipboard, you press the **Ctrl** key and hold it down while pressing the **c** key.
bold	Menu names, command names, dialog box buttons, and key names appear in bold type. Additionally, the text you enter in fields during procedures appears in bold type.
COMMENT IS NULL	Examples of programming code (such as PL/SQL) or SQL commands are emphasized with a different font.
	This caution symbol advises users that failure to take or avoid a specified action could result in significant data problems.

Medika Sample Studies

The Clintrial software provides three sample studies that you can optionally install and use as a learning aid.

For information about installing and using the sample study, see the *Clintrial Getting Started* guide, Chapter 7.

Clintrial 4.7 compatibility with other Oracle Health Sciences products

The *Products Compatibility Matrix*, which identifies Clintrial compatibility with other Oracle Health Sciences products, can be downloaded from https://extranet.phaseforward.com.

To sign in, click **Customer Login**. Enter your email address and password, and navigate to the **Bulletins** section.

If you need assistance

If you are an Oracle customer with a maintenance agreement, you can contact the Global Support Center for assistance with product issues.

Your maintenance agreement indicates the type of support you are eligible to receive and describes how to contact Oracle. Additionally, the Oracle website lists the toll-free support number for your product, location, and support level:

http://www.oracle.com/support/

In the event that our toll-free telephone service is interrupted, please use either of the following methods to contact the Global Support Center:

- email

  saasclinicalsupport_ww@oracle.com
- telephone

In the US:	1-800-633-0925
Outside of the US:	+44 (0) 207 131 2801

Oracle also provides assistance with User Management, Site Assessment, and Provisioning. Please refer to your Master Services Agreement and individual Statement of Work to determine if you are eligible to use these services.

1 Introduction to the Clintrial Software

Overview

This chapter introduces the basic concepts you need to use the Clintrial software. For installation information, see *Getting Started*.

The information in this chapter is generally conceptual. Procedures for most of these topics appear in the Clintrial software Basics Help that you can access through the Help menu on any Clintrial software module. A brief overview on how to use Clintrial software Help appears in "Using Help" on page 9.

To access the Clintrial software and its Help topics, you must start a Clintrial software module. Instructions for starting a Clintrial software module appear in this chapter as well as in the Clintrial software Basics Help.

Clintrial software modules

The Clintrial software consists of a set of integrated modules that can be installed as needed. This modular approach enables you to describe, collect, and manage clinical data according to the needs of your company's studies.

The Clintrial software core modules

The Clintrial software includes the core modules Admin, Design, Enter, Manage, and Retrieve.

About the Admin module

Use the Admin module to perform the system administration tasks. You can:

- Create user accounts and usergroup accounts.
- Manage passwords.
- Set up and manage access rights for users, usergroups, and protocols for all Clintrial software modules.
- Set system parameters.
- Monitor database space.
- Produce auditing reports for users and security.
- Produce reports about system activities.

About the Design module

Use the Design module to design and create the Clintrial database and the study books that you need to enter clinical data. You can:

- Design the clinical database to model your clinical protocol and meet your needs for storing and retrieving data.
- Create online representations of your paper CRFs for data-entry, verification, and editing.
- Create and manage your metadata standards.
- Produce reports about metadata.
- Import/export protocols.

About the Enter module

Use the Enter module to enter clinical data in the database interactively. You can:

- Add subjects to a study.
- Enter clinical data interactively.
- Verify clinical data.
- Edit clinical data.
- Add flags and notes to clinical data.
- Produce reports about clinical data.
- View scanned pages of your paper CRFs.
- Manually create and edit discrepancies, if the Resolve extended module is installed.

About the Manage module

Use the Manage module to perform data management tasks. You can:

- Batch load and apply data-entry checks to clinical data.
- Code clinical data using a coding thesaurus.
- Validate and merge clinical data in the database.
- Make global changes to or delete clinical data.
- Edit records using an Error Log.
- Track the auditing of data.
- Produce reports about metadata objects.

About the Retrieve module

Use the Retrieve module to access and extract clinical data from the database. You can:

- Create queries using:
  - Query By Form.
  - Query By Panel.
  - Ad Hoc Query.
  - Query By SQL.
- Save query specifications in a query library.
- Save query results to a variety of formats, such as SAS or spreadsheet files.

The Clintrial software extended modules

In addition to the Clintrial software core modules, your company may have purchased one or more of the following Clintrial software extended modules: Classify, Lab Loader, Multisite, and Resolve.

About the Classify module

Use the Classify extended module to work with thesaurus protocols and automatic coding. You can:

- Build and test complex coding algorithms.
- Find, track, and review solutions for values that fail automatic coding.
- Examine the contents of a coding thesaurus protocol, and compare different coding thesaurus protocols.
- Audit the contents of a coding thesaurus protocol.

About the Lab Loader module

Use the Lab Loader extended module to load laboratory data into Clintrial protocols. You can:

- Extend the batch loading capabilities provided in the Manage module.
- Build and maintain a set of lab normal ranges.
- Process loaded lab data.
- Batch load lab data to a source protocol.
- Perform preparatory work on lab data prior to transfer to a clinical data protocol.
- Transfer lab data into a clinical data destination protocol.

About the Multisite module

Use the Multisite extended module to perform interdatabase instance operations. You can:

- Distribute protocols and codelists to multiple sites.
- Replicate clinical and account data between multiple sites.
- Copy functions and base tables.
- Use protocols and codelists in a global environment.
- Share tables and PL/SQL functions across multiple sites.

About the Resolve module

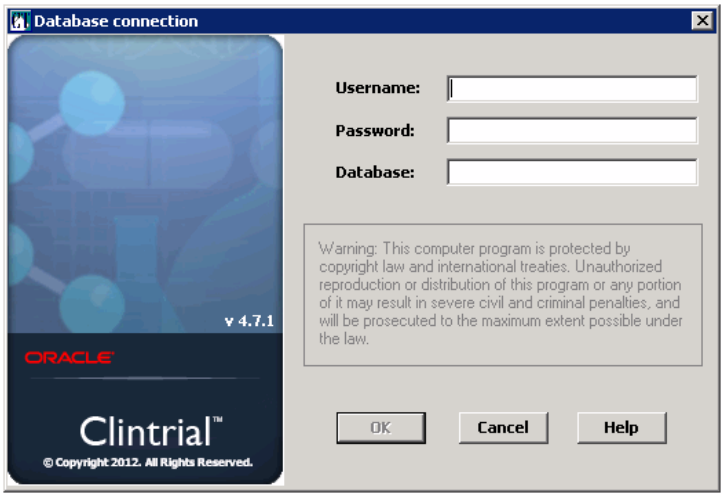
Use the Resolve extended module to manage discrepancy and resolution capabilities that support the work you perform in other Clintrial software modules. You can:

- Identify, track, and resolve potential or actual discrepancies in clinical data (inconsistent or missing data).
- Check for discrepancies automatically or by manual inspection.
- Record investigation and resolution information.

Starting a module

To start a module, from the Windows **Start** menu, select **Programs**. Select the Clintrial program group, then the module.

When you start a module, the Database Connection dialog box opens:



Starting a module for the first time

The first time you start a Clintrial software module, the fields in the Database Connection dialog box are empty. You must specify the following:

- Your user name
- Your password
- The Oracle Net Service Name for the database you will be using

If you do not know your user name or password, or the database service name that you should use, see your Clintrial software administrator.

Starting a module other than the first time

The next time you start a Clintrial software module, the user name and database service name that you last used to start a Clintrial software module on your computer are displayed as defaults in the Database Connection dialog box.

Using a Clintrial software module

The basic tasks necessary to use a Clintrial software module are:

- Using the Switchboard
- Setting the protocol
- Switching databases
- Re-ordering columns in list windows
- Changing your password
- Exiting the module

How to use the Switchboard

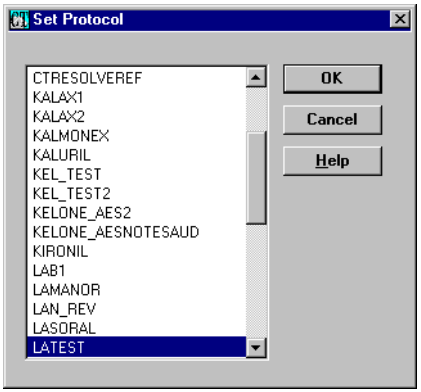
You can start any of the installed Clintrial software modules from the Switchboard **Run** menu. When you start the Switchboard, you provide a user name, password, and database service name. Each time that you start a different Clintrial software module from Switchboard, the Clintrial software uses the database connection information that you initially provided when you started Switchboard. More complete instructions appear in the Clintrial software Basics Help.

Note: Although you can always start the installed Clintrial software modules fromthe Switchboard, menu commands in the modules are available only if you have the appropriate access rights.

How to set a protocol

A *Clintrial software protocol* is a logical container that organizes the objects and clinical data for a clinical study.

When you first connect to the database, the Set Protocol dialog box opens:



Select the protocol in which you want to work. If you do not know the protocol in which you should work, see your Clintrial software administrator.

Note: The next time you start a Clintrial software module, the Clintrial software automatically selects the protocol that your user account most recently selected.

How to switch databases

To switch from the current database to another database without exiting the Clintrial software, close any open windows, and from the **File** menu, select **Connect**. More complete instructions appear in the Clintrial software Basics Help.

How to re-order columns in list windows

For windows that display list grid views, such as lists of logs in Manage or lists of Discrepancies in Resolve, you may use the mouse to drag columns to new positions to tailor the view to your needs. For example, you can drag and drop the most significant columns to the left portion of the open window, or place two related columns side-by-side in order to ease comparison of the data.

How to change your password

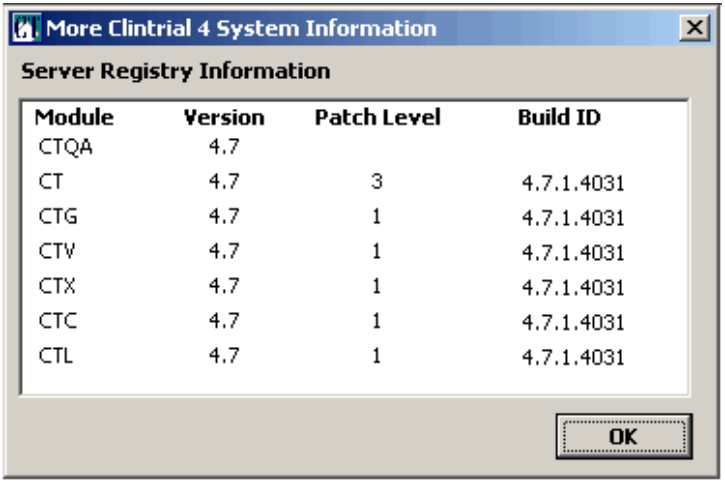
To modify the password for your user account, from the **File** menu, select **Password**. More complete instructions appear in the Clintrial software Basics Help.

How to access the Server Registry Information

You can access the Server Register Information report from any of the Clintrial software modules.

To open the report:

1. From the **Help** menu, select **About**.
2. Click **More**. The Server Registry Information opens, for example:



How to exit a module

To exit a module, from the **File** menu, select **Exit**. When you exit the application, the main window of the module closes and the Clintrial software disconnects you from the database. More complete instructions appear in the Clintrial software Basics Help.

Using Help

Each Clintrial software module is delivered with Help. Help includes:

- Context-sensitive Help for windows and dialog boxes
- Procedural instructions for tasks
- Brief overviews of concepts

- Descriptions of menus and options
- A glossary of Clintrial software terms

A Clintrial software Help file is installed automatically for each module that you install. In addition, the Help files that are not module-specific are always installed on your computer.

Below is an example of the screen that appears when you choose **Help: About** when you are in the Clintrial Admin module:

How to search for information

To search for information in the Help:

1. From the **Help** menu, select **Help Topics**. The Help Topics dialog box opens.
2. To find a topic in the Help:
   - Click the **Contents** tab to view topics by category.
   - Click the **Index** tab to view a list of index entries.
   - Click the **Find** tab to search for specific words in the Help.

How to get help on windows and dialog boxes

To get help on an open window or dialog box, do one of the following:

- Press **F1**.
- On the toolbar, click .
- Click **Help**.

Admin and Design

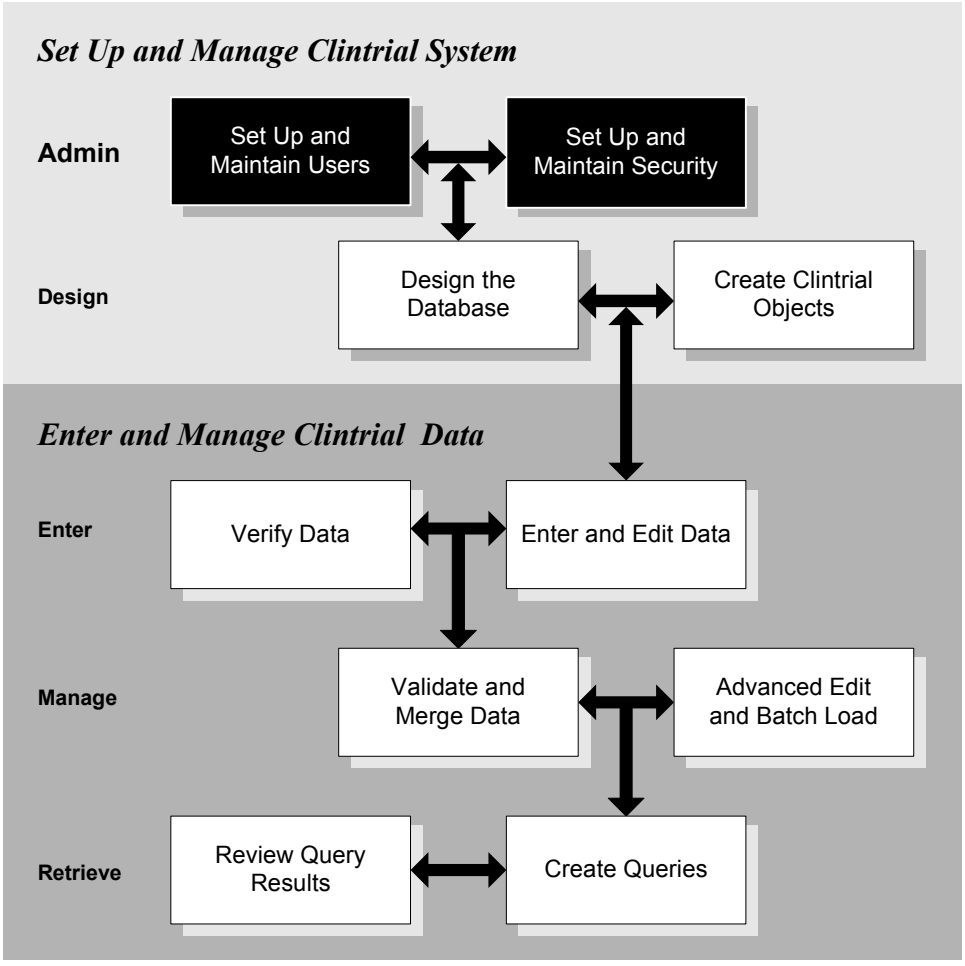
Section 1: Admin

2 Introduction to Admin

Overview

The Clintrial software Admin module is the core module used for administering the Clintrial software, a process that includes:

- Setting up and managing access to all Clintrial software modules by maintaining accounts, security, and parameters
- Creating reports that summarize tasks performed in Admin



What are the Admin tasks?

Using Admin, you can:

- Define who can access the Clintrial software by creating and managing accounts and usergroups (groups of user accounts).
- Define how users access the Clintrial software by assigning access rights.
- Define database-wide behavior of the Clintrial software by setting system parameters.
- Monitor database storage.
- View reports on the activities defined by Admin.

Who are Admin users?

The Clintrial software database administrator (DBA) or system administrator uses Admin.

Clintrial software and Oracle database tables

At installation, the Clintrial software creates tables to contain data that defines Clintrial software objects and other Clintrial software structures such as parameters and user accounts. For example, when you create a user account in Admin or create a protocol in Design, data that defines the user account or protocol is entered in tables created at installation.

When protocols are created in Design, the Clintrial software creates additional protocol-specific tables related to error handling, flags, and notes. When panels are installed in Design, you can optionally instruct the Clintrial software to create protocol-specific tables to store data that is entered in Enter.

What is metadata?

Metadata is data that defines:

- Clintrial software objects, such as protocols and codelists
- Clintrial software system information, such as system parameters and user accounts

The database tables that hold the metadata are called *data dictionary tables*.

What is clinical data?

Clinical data is data that is collected during a clinical trial; for example, data about a subject collected on a Case Report Form (CRF) page during a clinical trial and entered in Enter. The database tables that hold clinical data are called *clinical data tables*.

For more information

For more information on Oracle databases, see the Oracle documentation. For descriptions of the database tables, see the *Reference Guide*.

For more information on protocols, see Chapter 9.

For more information on panels, see Chapter 10.

Creating accounts

Oracle *accounts* provide access to the Clintrial software and the underlying database. The Oracle accounts are organized into three Clintrial software-specific types:

- User accounts
- System accounts
- Protocol accounts

For information on accounts, see Chapter 3.

Managing security

Clintrial software security is controlled by:

- Access rights
- Access levels

What are access rights and access levels?

An *access right* is a predefined set of Clintrial software activities that can be associated with a usergroup or a user. Some access rights relate to activities that require access to protocols and must be associated with a protocol as well as with a usergroup or user.

An *access level* determines what type of access the user has to the activities defined by an access right. The access levels are None, Full, Read, NoDelete, Publish, Write, and Basic.

For information on access rights and access levels, see Chapter 4.

Setting system parameters

There are four types of parameters that may be set in Admin:

- System parameters
- Protocol parameters
- User preferences
- Storage parameters

What are system parameters?

System parameters define the characteristics of the working environment for all users of an Oracle database instance to which the users connect through the Clintrial software. For example, the PASSWORD_MINIMUM system parameter sets the minimum password length for all Clintrial software users.

What are protocol parameters?

Protocol parameters are a subset of system parameters that can be set by the Design user for a specific protocol. Protocol parameters override the default system parameter settings.

For example, the CONNECT_REQD_DD system parameter can be set in Admin to Yes or No, to specify whether all objects must be connected when copied. CONNECT_REQD_DD is also a protocol parameter; therefore, in Design you can override the system parameter setting for each protocol.

What are user preferences?

User preferences are a subset of system parameters. Default values for user preferences come from corresponding system parameters. Each user can change the default values of the parameters for use in their account by changing the user preferences.

For example, the ENT_FORM_INSERT system parameter sets a default specifying when data entered by all data-entry operators is committed. Each data-entry operator can override the default in their user account by setting the corresponding ENT_FORM_INSERT user preference.

What are storage parameters?

Storage parameters determine the space allocation for database tables and indexes.

For information on storage parameters, see Chapter 5.

System administration tools

Admin allows you to:

- Monitor space utilization and space parameters through online reports.
- Use interactive SQL to query the database.
- Register the database.
- Set and monitor a System Activity log.

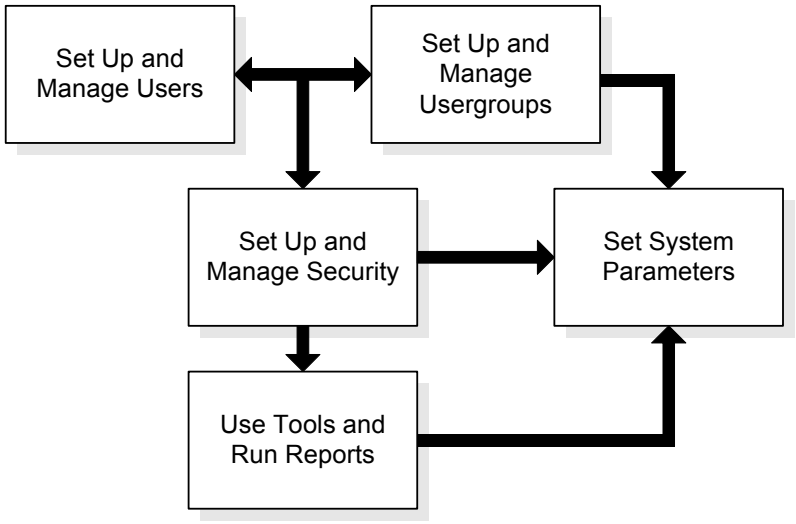
Admin also provides reports on:

- Users
- Usergroups
- Access rights
- System parameters
- Storage parameters
- User procedures
- Protocol lock history
- Security audits
- System parameter audits

For information on system administration tools, see Chapter 7.

Workflow within Admin

The following figure shows the typical workflow within Admin:



3 Accounts

Oracle accounts and Clintrial software accounts

Oracle accounts provide access to the Clintrial software and the underlying database.

Types of accounts

The Oracle accounts are organized into three Clintrial software -specific types:

- User accounts
- System accounts
- Protocol accounts

Who creates accounts?

The Clintrial software creates the CTSYS user account at installation. You can use the CTSYS account to create other user accounts. Any user account with the Full Admin System access right can create another user account.

The Clintrial software creates system accounts at installation.

The Clintrial software also creates a protocol account for each protocol that is created in Design.

For more information on access rights, see Chapter 4.

User accounts

A Clintrial software *user account* is an Oracle account that a Clintrial software user connects to in order to work. A user account must be granted access rights to a protocol account before the user can add, modify, display, or delete data associated with that protocol. Each user should be given an account. Multiple users may not share one account.

How to create or modify a user account

To create a user account, you name the account (up to 20 characters long) and specify other user account attributes. To modify a user account, you change any of the attributes other than the user account name. You can also delete a user account.

To change an existing Oracle account into a Clintrial software user account so that Admin access rights can be assigned, log in to SQL*Plus with a privileged Oracle account (for example, SYSTEM) and grant the CT_USER role to that user, as follows:

SQL> **GRANT CT_USER to *account-name*;**

This allows the Admin user to assign access rights for that account and modify or delete the account and password.

How to supply a user account password

When you create a user account, you supply a password. You can modify the password later. Any user logged in to a user account can also modify the password.

List of user account attributes

The following table lists the attributes of a user account:

Attribute:	**Description:**
User Name	Name of the user account.
Password	Password for the user account. The minimum length of the password is set by the PWD_MINIMUM system parameter.

Attribute:	Description:
Default Tablespace	Name of the default tablespace for user accounts. The tablespaces are usually created outside of the Clintrial software. You can name the default tablespace from among those created at installation or by the system administrator after installation. Defaults to the value of the USR_DFT_TSP system parameter setting.
Temporary Tablespace	Name of the default temporary tablespace for the tables owned by this user account. Oracle sometimes uses temporary tables when performing database activities. You can specify a temporary tablespace name. Defaults to the value of the USR_TMP_TSP system parameter setting.
Profile	The Oracle profile for this account. Clintrial software uses the default profile created by Oracle. Defaults to the value of the USR_DB_PROFILE system parameter.
Description	Description of the user account, for example, a job description such as investigator.
Full Name	Necessary for signing study pages, this is the first and last name for the user for whom the account is designated. It is used when a study page is signed.
Investigator Site	This field is obsolete.

Considerations for managing user accounts

As the system administrator, you must decide who the users are, whether they should be grouped into usergroups, and what combination of access rights, access levels, and protocols (for protocol-based access rights) is appropriate for each group or individual user.

Usergroups

A *usergroup* is a set of user accounts to which you grant and revoke access rights as a group.

How to plan a usergroup

If there are multiple users who require the same access rights, you can group their user accounts into one or more usergroups. You assign or revoke access rights and access levels to the usergroup, rather than to each user account individually. Usergroups make it easier for you to manage user access rights and to set up new users with the appropriate access rights.

A usergroup can contain any number of users.

To create a usergroup, you must determine the common characteristics of a set of users with which you intend to associate access rights and access levels. Therefore, before you create each usergroup, consider the following:

- Which Clintrial software activities do these users need to perform?
- What metadata or clinical data do these users need access to?
- For the specified data, what tasks do these users need to perform (that is, read, write, modify, or delete data)?

For descriptions of the database tables, see the *Reference Guide*.

List of Clintrial software-supplied usergroups

The Clintrial software supplies three predefined usergroups, with access rights and access levels for activities associated with the following modules:

- ADMIN — for Admin
- ANALYSIS — for Retrieve
- ENTRY — for Enter

You can add users to these usergroups as needed. You can also delete these usergroups, or modify the default access rights and access levels for them.

For information about the default access rights and access levels for these usergroups, see the Admin Help.

How to create or modify a usergroup

To create a usergroup, name it, then add one or more user accounts. To modify a usergroup, add or delete user accounts from the usergroup.

List of usergroup attributes

The following table lists the usergroup attributes:

Attribute:	Description:
Usergroups	List of existing usergroups
Members	User accounts that belong to the usergroup
Nonmembers	User accounts that do not belong to the usergroup

For more information

For more information on access rights and access levels, see "Access rights and access levels" on page 44.

For more information on the default access rights and access levels for Clintrial software-supplied usergroups, see "List of access rights for usergroups" on page 60.

System accounts

A Clintrial software *system account* is an Oracle account that contains database-wide information.

List of system accounts

The following table lists the system accounts:

System account:	Description:
CISUSER account	The CISUSER account is used by the CIS synchronization process to transfer information to and from InForm. ***Note:*** This account does not store information.
CTS account	The CTS system account contains database-wide information about protocol accounts, flags, notes, user accounts, and user privileges.
CTSDD account	The CTSDD system account contains database-wide information about protocol-specific metadata definitions; for example, item and panel attributes for all installed panels.
CTSCODES account	The CTSCODES system account contains database-wide codelist information. CTSCODES consists of five tables plus one table for each unaggregated codelist.
CTS$LOAD_*protocol-name* account	The CTS$LOAD _*protocol-name* account handles privileges for batch loading. There is one CTS$LOAD account for each protocol.
CTPROC account	The CTPROC system account stores internal Clintrial software procedures.
CXFR_SEND account	The CXFR_SEND system account contains codelists when they are exported.
CXFR_RECV account	The CXFR_RECV system account contains codelists when they are imported.
PXFR_SEND account	The PXFR_SEND system account contains protocol metadata when the protocol is exported.
PXFR_RECV account	The PXFR_RECV system account contains protocol metadata when the protocol is imported.

System account:	Description:
CTCLASSIFY account	The CTCLASSIFY account is created automatically when you install Classify. This account stores information about Classify-specific objects, such as omissions.
CTSRM account	The CTSRM account is created automatically when you install the Distribution (CTC) server component of Multisite. This account stores information that is used to manage distribution across multiple sites.
CTSRP account	The CTSRP account is created automatically when you install the Replication (CTX) server component of Multisite. This account stores information that is used to manage replication across multiple sites.
REVIEWADMIN account	The REVIEWADMIN account is created automatically when you install AdHoc Query. This account stores information that is used in the Retrieve module.

How to access system account tables

Access to the tables created by system accounts is granted automatically based on access rights assigned to a user account. For example, a user account with access rights to create a user account has access rights to the tables created by the CTS account, and a user account with access rights to create or modify a codelist has access rights to the tables created by the CTSCODES account.

How to set a system account password

The password for a Clintrial system account is created by the system administrator during the installation process. You can change the password in Admin.

For more information

For more information on exporting and importing protocols and codelists, see Chapter 20.

For more information on the database tables for system accounts, see the *Reference Guide*.

Protocol accounts

A *protocol account* is an Oracle account that stores information specific to a protocol. Each protocol account contains the following information:

- Clinical data for a particular clinical study or group of studies
- Error log records, flags, and notes associated with one or more clinical studies
- One or more views of the clinical data stored in the protocol

Flag and note data and error log records are stored in protocol-specific tables, and clinical data is stored in panel-specific update, data, and audit tables.

How to access protocol accounts

Access to the tables created by protocol accounts is granted automatically based on access rights assigned to a user account.

How to set a protocol account password

The Clintrial software-supplied password for a protocol account is generated internally and encrypted. You can change the password in Admin; however, it will no longer be encrypted unless the Admin module system parameter PASSWORD_ENCRYPTION is set to **Yes**.

For more information

For more information on flags and notes, see Chapter 13 of this book, and *Enter, Resolve, and Retrieve*.

For more information on clinical data tables, see Chapter 10 of this book, and *Enter, Resolve, and Retrieve*.

Account and audit reports

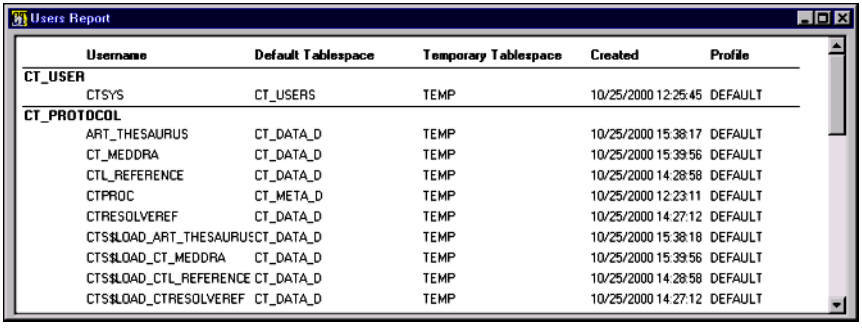
Admin provides the following account and audit reports:

- Users Report
- User Audit Report
- User Access Audit Report
- Usergroups Report — Sorted by:
  - User
  - Usergroup
- Usergroup Audit Report
- Usergroup Access Audit Report

The account reports can be run under the **Reports** menu, and the audit reports are run under the **Audit** menu in Admin.

What is the Users Report?

The Users Report provides information about user accounts. You choose the report from the **Reports** menu. The following example shows part of a Users Report:



Username	Default Tablespace	Temporary Tablespace	Created	Profile
CT_USER				
CTSYS	CT_USERS	TEMP	10/25/2000 12:25:45	DEFAULT
CT_PROTOCOL				
ART_THESAURUS	CT_DATA_D	TEMP	10/25/2000 15:38:17	DEFAULT
CT_MEDDRA	CT_DATA_D	TEMP	10/25/2000 15:39:56	DEFAULT
CTL_REFERENCE	CT_DATA_D	TEMP	10/25/2000 14:28:58	DEFAULT
CTPROC	CT_META_D	TEMP	10/25/2000 12:23:11	DEFAULT
CTRESOLVEREF	CT_DATA_D	TEMP	10/25/2000 14:27:12	DEFAULT
CTS$LOAD_ART_THESAURUS	CT_DATA_D	TEMP	10/25/2000 15:38:18	DEFAULT
CTS$LOAD_CT_MEDDRA	CT_DATA_D	TEMP	10/25/2000 15:39:56	DEFAULT
CTS$LOAD_CTL_REFERENCE	CT_DATA_D	TEMP	10/25/2000 14:28:58	DEFAULT
CTS$LOAD_CTRESOLVEREF	CT_DATA_D	TEMP	10/25/2000 14:27:12	DEFAULT

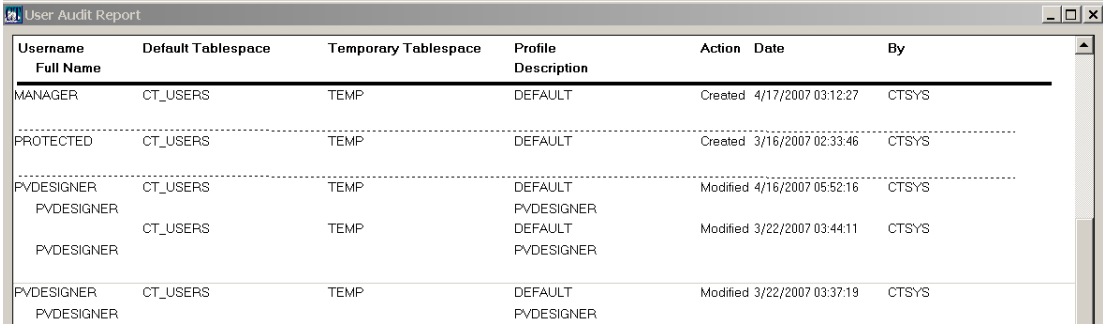
The Users Report contains the following information about accounts:

Report column:	**Description:**
Username	Name of each user account, grouped by the account categories User, Protocol, and Other.

Report column:	Description:
Default Tablespace	Name of the default tablespace for the tables created by this user account.
Temporary Tablespace	Name of the default temporary tablespace for the tables created by this user account.
Created	Date and time this user account was created. ***Note:*** If a Clintrial instance has been imported from another Clintrial instance using Oracle Data Pump Import/Export after the installation or upgrade to Clintrial 4.7, the date and time listed here reflects the original date and time the account was created. In previous releases, the original date was overridden and this entry reflected the date that the user was created in the imported instance, which was the date and time of the import.
Profile	The Oracle profile for this account.

What is the User Audit Report?

The User Audit Report shows the history of the creation, modification or deletion of user accounts since the installation or upgrade to Clintrial 4.7. You choose the report from the **Audit** menu. The following example shows part of a User Audit Report:

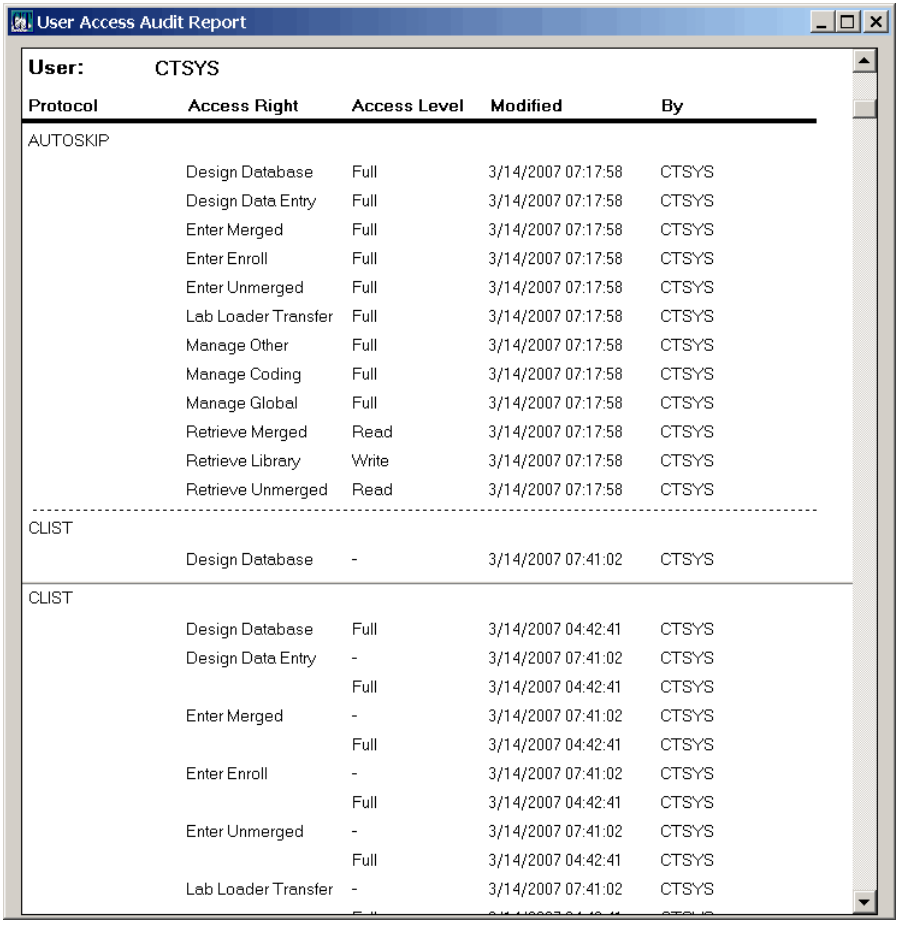


The User Audit Report contains the following information about accounts:

Report column:	Description:
Username	Name of each user account.
Default Tablespace	Name of the default tablespace for the tables created by this user account.
Temporary Tablespace	Name of the default temporary tablespace for the tables created by this user account.
Profile Description	The Oracle profile for this account.
Action	Creation, modification or deletion of the account.
Date	Date and time of action.
By	Name of user account that modified the account.

What is the User Access Audit Report?

The User Access Audit Report shows the history of the modification of user Protocol and Non-Protocol access rights since the installation or upgrade to Clintrial 4.7. Upon choosing the report under the **Audit** menu, dialog box appears allowing the selection of a single user. The following example shows part of a User Access Audit Report:



User Access Audit Report

User: CTSYS

Protocol	Access Right	Access Level	Modified	By
AUTOSKIP				
	Design Database	Full	3/14/2007 07:17:58	CTSYS
	Design Data Entry	Full	3/14/2007 07:17:58	CTSYS
	Enter Merged	Full	3/14/2007 07:17:58	CTSYS
	Enter Enroll	Full	3/14/2007 07:17:58	CTSYS
	Enter Unmerged	Full	3/14/2007 07:17:58	CTSYS
	Lab Loader Transfer	Full	3/14/2007 07:17:58	CTSYS
	Manage Other	Full	3/14/2007 07:17:58	CTSYS
	Manage Coding	Full	3/14/2007 07:17:58	CTSYS
	Manage Global	Full	3/14/2007 07:17:58	CTSYS
	Retrieve Merged	Read	3/14/2007 07:17:58	CTSYS
	Retrieve Library	Write	3/14/2007 07:17:58	CTSYS
	Retrieve Unmerged	Read	3/14/2007 07:17:58	CTSYS
CLIST				
	Design Database	-	3/14/2007 07:41:02	CTSYS
CLIST				
	Design Database	Full	3/14/2007 04:42:41	CTSYS
	Design Data Entry	-	3/14/2007 07:41:02	CTSYS
		Full	3/14/2007 04:42:41	CTSYS
	Enter Merged	-	3/14/2007 07:41:02	CTSYS
		Full	3/14/2007 04:42:41	CTSYS
	Enter Enroll	-	3/14/2007 07:41:02	CTSYS
		Full	3/14/2007 04:42:41	CTSYS
	Enter Unmerged	-	3/14/2007 07:41:02	CTSYS
		Full	3/14/2007 04:42:41	CTSYS
	Lab Loader Transfer	-	3/14/2007 07:41:02	CTSYS

Note: If a user is deleted and then recreated with the same name, the User Access Audit Report shows that all of its rights were revoked. There is no "Deleted" designation. Rights granted or revoked since the recreation will also appear on the report.

Oracle recommends that deleted users not be reused.

Additional audit information on deleted users may be obtained from the User Audit Report.

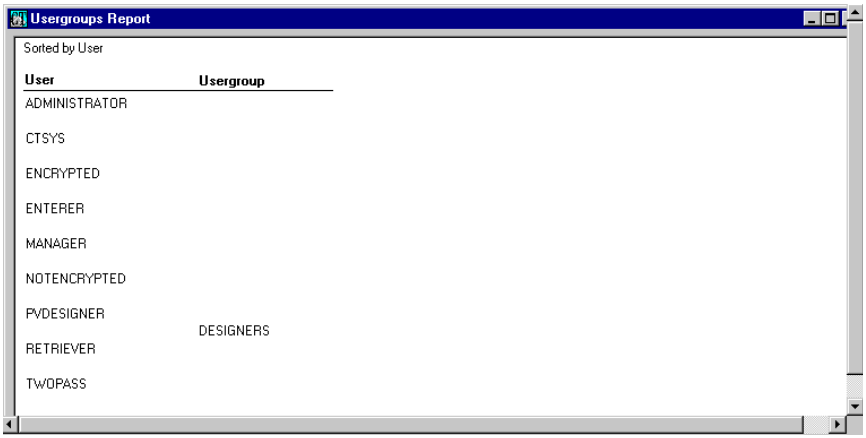
The Protocol column in the User Access Audit Report is blank for Non-Protocol Access rights, and these are sorted to the top.

The User Access Audit Report contains the following information about accounts:

Report column:	**Description:**
Protocol	Names of the protocols to which the user has access.
Access Right	Non-Protocol and Protocol Access Rights, with the Non-Protocol Access Rights sorted to the top of the list.
Access Level	The access level assigned to the user for the Access Right listed on the date specified.
Modified	Date and time of modification.
By	Name of user account that modified the Access Level.

What is the Usergroups Report?

The Usergroups Report provides a list of users and which users are members of the usergroups, if any. When sorted by user, the Usergroups Report orders the users alphabetically and lists usergroups associated with each user. You choose the report from the **Reports** menu. The following example shows part of a Usergroups Report, sorted by user:

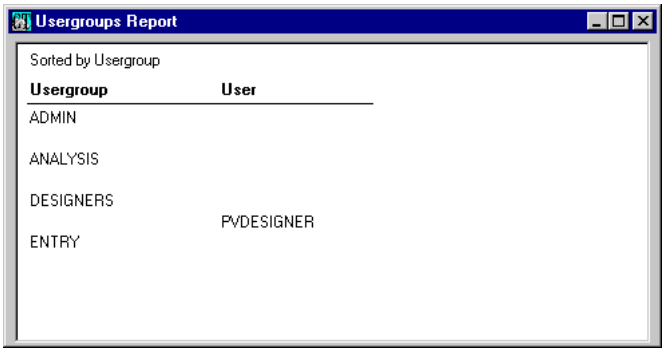


If a user belongs to one or more usergroups, the usergroups are listed below and to the right of the user.

The report contains the following information:

Report column:	**Description:**
User	The name of each user account.
Usergroup	The usergroups to which the user belongs, if any.

When sorted by usergroup, the Usergroups Report orders the usergroups alphabetically and lists users associated with each usergroup. The following example shows the Usergroups Report, sorted by usergroup:



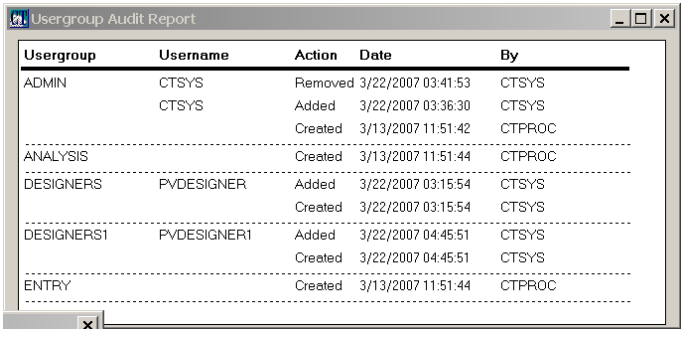
If a usergroup contains one or more users, the users are listed below and to the right of the usergroup.

The report contains the following information:

Report column:	Description:
Usergroup	The name of each usergroup.
User	The users, if any, in the usergroup.

What is the Usergroup Audit Report?

The Usergroup Audit Report shows the creation date of the usergroups and the history of the addition and removal of user accounts since the installation or upgrade to Clintrial 4.7. You choose the report from the **Audit** menu. The following example shows part of a Usergroup Audit Report:

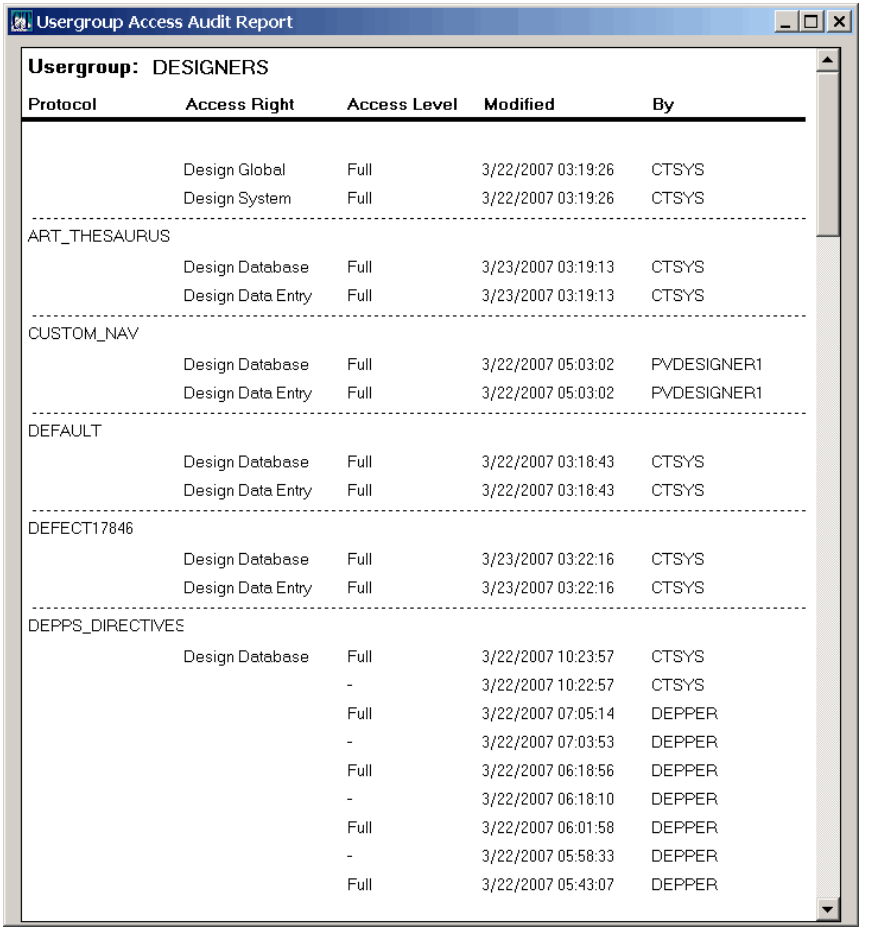


The Usergroup Audit Report contains the following information about accounts:

Report column:	Description:
Usergroup	The name of each usergroup.
Username	Name of each user account.
Action	Creation of a usergroup, addition or removal of a user account to a usergroup.
Date	Date and time of action.
By	Name of user account that modified the usergroup.

What is the Usergroup Access Audit Report?

The Usergroup Access Audit Report shows the history of the modification of usergroup Protocol and Non-Protocol access rights since the installation or upgrade to Clintrial 4.7. Upon choosing the report under the **Audit** menu, dialog box allowing the selection of a single usergroup. The following example shows part of a Usergroup Access Audit Report:



Usergroup Access Audit Report

Usergroup: DESIGNERS

Protocol	Access Right	Access Level	Modified	By
	Design Global	Full	3/22/2007 03:19:26	CTSYS
	Design System	Full	3/22/2007 03:19:26	CTSYS
ART_THESAURUS				
	Design Database	Full	3/23/2007 03:19:13	CTSYS
	Design Data Entry	Full	3/23/2007 03:19:13	CTSYS
CUSTOM_NAV				
	Design Database	Full	3/22/2007 05:03:02	PVDESIGNER1
	Design Data Entry	Full	3/22/2007 05:03:02	PVDESIGNER1
DEFAULT				
	Design Database	Full	3/22/2007 03:18:43	CTSYS
	Design Data Entry	Full	3/22/2007 03:18:43	CTSYS
DEFECT17846				
	Design Database	Full	3/23/2007 03:22:16	CTSYS
	Design Data Entry	Full	3/23/2007 03:22:16	CTSYS
DEPPS_DIRECTIVES				
	Design Database	Full	3/22/2007 10:23:57	CTSYS
		-	3/22/2007 10:22:57	CTSYS
		Full	3/22/2007 07:05:14	DEPPER
		-	3/22/2007 07:03:53	DEPPER
		Full	3/22/2007 06:18:56	DEPPER
		-	3/22/2007 06:18:10	DEPPER
		Full	3/22/2007 06:01:58	DEPPER
		-	3/22/2007 05:58:33	DEPPER
		Full	3/22/2007 05:43:07	DEPPER

Note: If a usergroup is deleted and then recreated with the same name, the Usergroup Access Audit Report shows that all of its rights were revoked. There is no "Deleted" designation. Rights granted or revoked since the recreation will also appear on the report.

Oracle recommends that deleted usergroups not be reused.

Additional audit information on deleted users/usergroups may be obtained from the Usergroup Audit Report.

The Protocol column in the Usergroup Access Audit Report is blank for Non-Protocol Access rights, and these are sorted to the top.

The Usergroup Access Audit Report contains the following information about accounts:

Report column:	**Description:**
Protocol	Names of the protocols to which the usergroup has access.
Access Right	Non-Protocol and Protocol Access Rights, with the Non-Protocol Access Rights sorted to the top of the list.
Access Level	The access level assigned to the usergroup for the Access Right listed on the date specified.
Modified	Date and time of modification.
By	Name of user account that modified the Access Level.

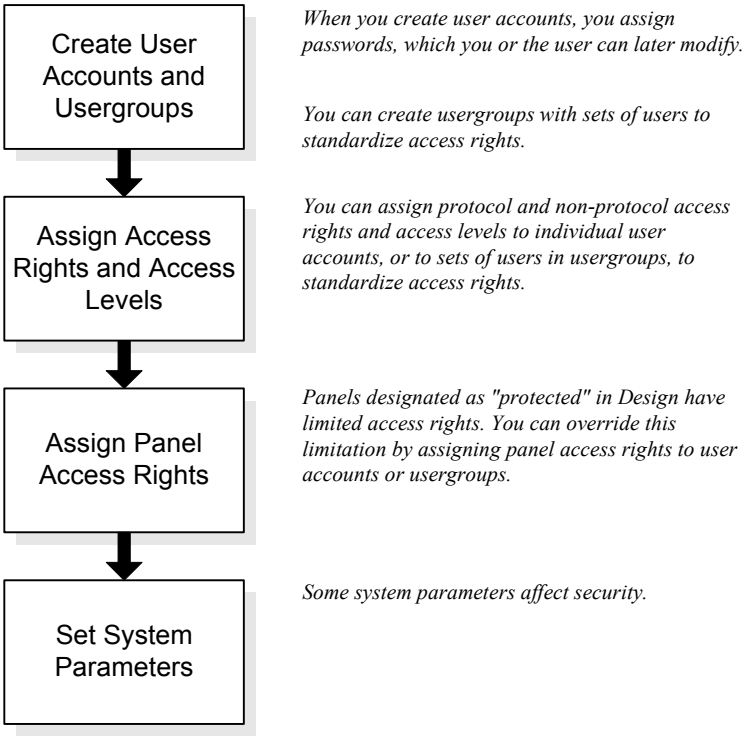
4 Security

Overview

As Clintrial software administrator, you have access to several functions that affect security:

- Access rights and levels — Described in this chapter
- Users and usergroups — Described in Chapter 3
- System parameters — Described in Chapter 5

The following figure shows a typical workflow in Admin for setting security functions:



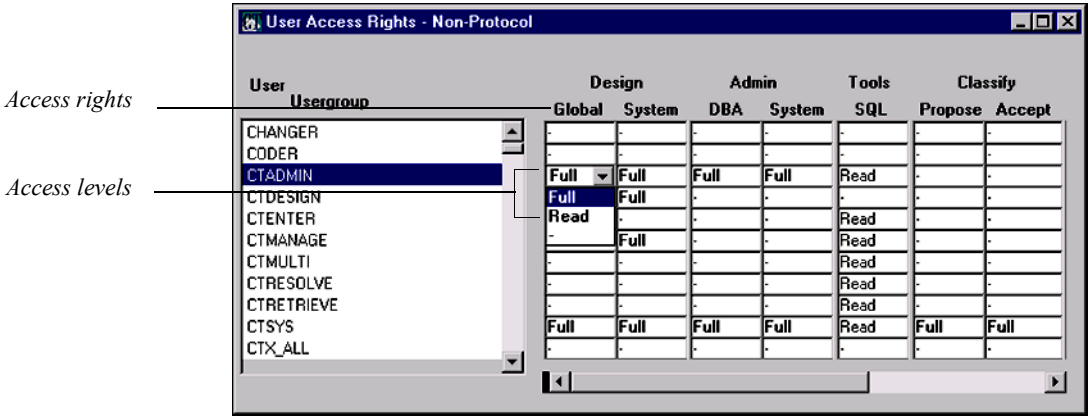
Access rights and access levels

What is an access right?

An *access right* is a predefined set of Clintrial software activities that can be associated with a usergroup or a user. Some access rights relate to activities that require access to protocols and must be associated with a protocol as well as with a usergroup or user.

What is an access level?

An *access level* determines what type of access the user has to the activities defined by an access right. The access levels are None, Full, Read, NoDelete, Publish, Write, and Basic. The following figure shows a display of access rights and access levels:



Note: The scroll bar at the bottom indicates that access rights for other modules are off the screen to the right.

How to set access rights and levels

You set access rights from a user account with Admin System access rights. The Clintrial software provides the Admin System access rights to the CTSYS account at installation.

You set protocol access rights by one of the following methods:

- Select a user or usergroup and specify the protocols to which the user or usergroup has access.
- Select a protocol and specify the users or usergroups that have access to the protocol.

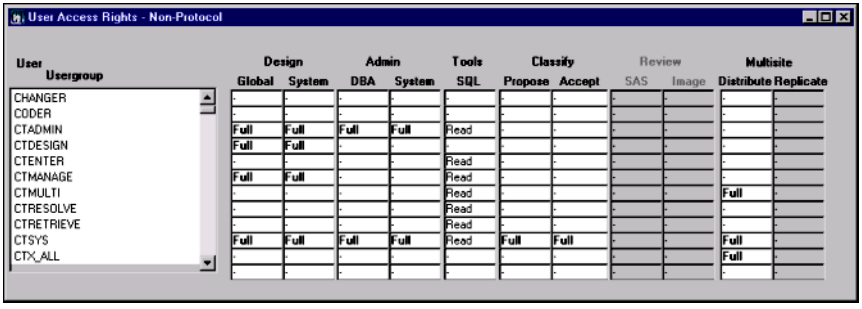
A user who is also part of a usergroup has the highest access rights assigned to that user account or to the usergroups to which the user belongs.

Each combination of a Clintrial software access right and access level corresponds to an underlying Oracle role.

Non-protocol access rights

A *non-protocol access right*, which is associated with a user or usergroup, pertains to Clintrial software activities that do not require access to a particular protocol. Examples of non-protocol access rights are codelist creation and modification in Design or setting parameters in Admin.

The following example shows the display of non-protocol access rights and access levels, with the CTSYS account selected:



Non-protocol access rights are categorized by the Clintrial software module within which they operate: Design, Admin, Classify, Review, and Multisite. There are also non-protocol access rights for SQL access in Admin, Manage, Retrieve, and Multisite.

Note: Access rights for extended modules are available only if the extended module is installed on the server. For non-protocol access rights, this applies to Classify, Review, and Multisite.

The CTSYS account that is created at installation has all non-protocol access rights.

List of Design non-protocol access rights

The following table lists the Design non-protocol access rights and access levels:

Access right:	User can:	Access levels:
Global	Work on codelists, flags, and notes.	Full — Can perform all valid actions on these objects. Read — Can view these objects, but not edit them. None — Cannot access these objects.
System	Work on protocols and thesaurus objects.	Full — Can perform all valid actions on the protocol objects. Read — Can view these objects, but not edit them. None — Cannot access these objects.

List of Admin non-protocol access rights

The following table lists the Admin non-protocol access rights and access levels:

Access right:	User can:	Access levels:
DBA	• Edit storage parameters. • Register sites. • View related reports and the Activity Log.	Full — Can perform all valid actions on these objects. Read — Can view, but not edit data. None — Cannot access these options.

Access right:	User can:	Access levels:
System	• Create user accounts. • Assign access rights. • Edit system parameters. • View related reports and Protocol Lock History Report.	Full — Can perform all valid actions on parameter settings and security functions. Read — Can view but not work on parameter settings and security functions. None — Cannot view parameter or security settings.

List of SQL non-protocol access rights in the Clintrial software

The following table lists the SQL non-protocol access rights and access levels that affect the use of SQL in Admin, Manage, and Retrieve:

Access right:	User can:	Access levels:
SQL	Use SQL.	Read — Can perform SQL read-only statements (SELECT and DESCRIBE) from within the Clintrial software. None — Cannot use SQL from within the Clintrial software.

List of Classify non-protocol access rights

The following table lists the Classify non-protocol access rights and access levels:

Access right:	User can:	Access levels:
Propose	• Review omission record. • Propose solutions. • Edit proposed solutions. • Clear proposed solutions. • Reuse solutions.	Full — Can perform all the tasks listed in the column to the left. Read — Can review omission records, but not perform any of the other tasks. None — Cannot perform any of the tasks listed in the column to the left.
Accept	• Review omission records. • Accept proposed solutions. • Purge solved and obsolete omissions.	Full — Can perform all of the tasks listed. NoDelete — Can review omission records and accept proposed solutions, but cannot purge solved and obsolete omissions. Read — Can review omission records, but not perform other tasks listed. None — Cannot perform any of the tasks listed.

List of Review non-protocol access rights

The following table lists the Review non-protocol access rights:

Access right:	User can:	Access levels:
SAS	Run SAS procedures.	Full — Can view and run SAS procedures that are designated “Basic” or “Full.” Basic — Can view and run SAS procedures that are designated “Basic.” None — Cannot view or run SAS procedures.
Image	View CRF images.	Read — Can view CRF images. None — Cannot view CRF images.

List of Multisite non-protocol access rights

The following table lists the Multisite non-protocol access rights and access levels:

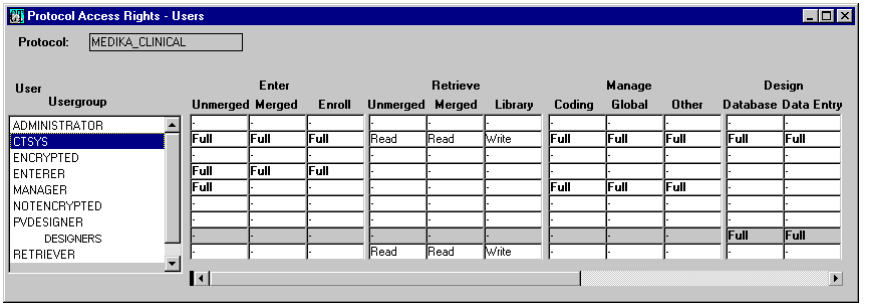
Access right:	User can:	Access levels:
Distribute	• Distribute codelists and protocols. • Copy base tables and functions. • Register sites for distribution.	Full — Can distribute codelists and protocols. Read — Can access browsers and read-only windows related to distribution. None — Cannot access any distribution-related windows.

Access right:	User can:	Access levels:
Replicate	• Replicate protocols and accounts. • Register sites for replication.	Full — Can replicate protocols and accounts. Read — Can access browsers and read-only windows related to replication. None — Cannot access any replication-related windows.

Protocol access rights

A *protocol access right* pertains to Clintrial software activities that require access to specific protocols. A protocol access right must be associated with a protocol as well as with a user or usergroup. Examples of protocol access rights are panel creation or modification in Design or data entry in Enter.

The following figure shows the display of protocol access rights for MEDIKA_CLINICAL for all user accounts in the database instance:



Note: The scroll bar at the bottom indicates that access rights for other modules are off the screen to the right.

Protocol access rights are categorized by the Clintrial software module within which they operate: Enter, Retrieve, Manage, Design, Resolve, Lab Loader, and Classify.

Note: Access rights for extended modules are available only if the extended module is installed on the server. For protocol access rights, this applies to Resolve, Lab Loader, and Classify.

Note: Only Design access rights can be applied to Data Dictionary protocols.

What are default protocol access rights?

The DEFAULT protocol access rights are assigned to a protocol when it is created. DEFAULT is not an actual protocol, but rather a group of access rights you select as the default for all other protocols. In this manner, standard access right settings can be defined for protocols without having to assign them to each protocol that is created.

For example, if data-entry operators are allowed full access to unmerged data in Enter, regardless of the protocol, you could set the access level for the Enter-Unmerged access right to Full. The Enter-Unmerged access right will be set to Full for all protocols.

You can change these access rights by the same method in which you change access rights for other protocols.

List of Enter protocol access rights

The following table lists the Enter protocol access rights and access levels:

Access right:	**User can:**	**Access levels:**
Unmerged	Work on unmerged data.	Full — Can view and update unmerged data. NoDelete — Can view and update, but not delete unmerged data. Read — Can view but not update or delete unmerged data. None — Cannot view the data. Sign — This field is obsolete. ***Note:*** Unmerged access cannot be applied to View protocols.

Access right:	User can:	Access levels:
Merged	Work on merged data.	Full — Can view and update merged data. NoDelete — Can view and update, but not delete merged data. Read — Can view but not update or delete merged data. None — Cannot view the data. ***Note:*** Merged access cannot be applied to Lab Loader protocols. ***Note:*** Only Read level can be applied for View protocols.
Enroll	Enroll subjects.	Full — Can enroll subjects but not delete them. Read — Can view enrollment study book, but not enroll or delete enrollment data. None — Cannot enroll subjects. ***Note:*** Read access must be granted for either the Merged or Unmerged access right in order for Enroll access rights to be used. ***Note:*** Only Read level can be applied to View protocols.

List of Retrieve protocol access rights

The following table lists the Retrieve protocol access rights and access levels:

Access right:	User can:	Access levels:
Unmerged	Query the update and audit tables.	Read — Can query unmerged data. None — Cannot query unmerged data. ***Note:*** Unmerged access cannot be applied to View protocols.
Merged	Query the data and audit tables.	Read — Can query merged data. None — Cannot query merged data. ***Note:*** Merged access cannot be applied to Lab Loader protocols.
Library	Save queries.	Publish — Can make, save, use and delete private and public queries. Write — Can make, save, use, and delete private queries. Read — Can use public queries, but not make, save or delete either private or public queries. None — Cannot make, save, use or delete queries.

List of Manage protocol access rights

The following table lists the Manage protocol access rights and access levels:

Access right:	User can:	Access levels:
Coding	Code data using values from a coding thesaurus.	Full — Can code. Read — Can view coding, but not make changes None — Cannot code. ***Note:*** Coding access cannot be applied to View protocols.
Global	Make global changes and deletions.	Full — Can make global changes and deletions. Read — Can open global change dialog boxes, but not make changes or deletions. None — Cannot make global changes and deletions. ***Note:*** Global access cannot be applied to View protocols.
Other	Perform Manage activities other than code or make global changes and deletions.	Full — Can perform all these Manage tasks; Can view reports except Coded Item Report; Can purge reports except Global Delete Report, Global Change Report, and Coded Item Report. Read — Can view, but not perform other Manage tasks. None — Cannot perform these Manage tasks. ***Note:*** Other access cannot be applied to View protocols.

Note: If you plan to edit records directly from the Error Log, you must also have the Unmerged protocol access right for Enter, with the Full access level. If you plan to edit records in the data table from the Error Log, you must have the Merged protocol access right for Enter, with the Full access level.

List of Design protocol access rights

The following table lists the Design protocol access rights and access levels:

Access right:	User can:	Access levels:
Database	Work on panels, items, rules, derivations, and coding targets.	Full — Can perform all valid actions on these objects. Read — Can view objects' attributes, but not work on these objects. None — Cannot view these objects.
Data Entry	Work on page sections, page templates, study pages, and study books.	Full — Can perform all valid actions on these objects. Read — Can view objects' attributes, but not work on these objects. None — Cannot view these objects.

List of Resolve protocol access rights

Resolve protocol access rights can only be applied to protocols set up to use Resolve through the **Setup for Resolve** activity in Resolve. The following table lists the Resolve protocol access rights and access levels:

Access right:	User can:	Access levels:
Create	Create new discrepancy records manually. Add associated items to or delete them from manually created records. Modify the Additional Message, Replacement Message, and Proposed Resolution values in any records that the user created manually, provided their discrepancy status is New.	Full — Can create and view discrepancy records. NoDelete — Can create discrepancy records but cannot delete them. Read — Can view discrepancy records but cannot edit them. None — Cannot create discrepancy records.
Propose	• Modify values for a discrepancy record in Resolve, except Comments 1 and Query for Confirmation, which can be modified only by users with Manage access. • Change the discrepancy status of a discrepancy record. ***Note*:** In some instances you cannot change the discrepancy status. For more details, see the *Resolve* section of *Enter, Resolve, and Retrieve.*	Full — Can modify or delete values for discrepancy records or change the discrepancy status of discrepancy records. NoDelete — Can modify, but cannot delete, the values for discrepancy records and can change the discrepancy status for discrepancy records. Read — Can view but not edit discrepancy records. None — Cannot modify or delete values for discrepancy records or change the discrepancy status.

Access right:	User can:	Access levels:
Manage	• Modify values for a discrepancy record in Resolve, except Comments 2, which can be modified only by users with Propose access. • Change the discrepancy status of a discrepancy record. • Run predefined reports. • Apply new data values to data records. • Delete discrepancy records. ***Note*:** In some instances you cannot change the discrepancy status. For more details, see the *Resolve* section of *Enter, Resolve, and Retrieve.*	Full — Can perform all of the listed tasks. NoDelete — Can view and edit data but cannot delete discrepancy records. Read — Can view but not edit discrepancy records. None — Cannot perform any of the listed tasks.
Produce	Print data discrepancy forms.	Full — Can print data discrepancy forms. Read — Can view, but not print, discrepancy records. None — Cannot print data discrepancy forms.

List of Lab Loader protocol access rights

The following table lists the Lab Loader protocol access rights and access levels:

Access right:	User can:	Access levels:
Design	Work on control files and transfer maps.	Full — Can perform all actions on control files and transfer maps. Read — Can view attributes of control files and transfer maps. None — Cannot access control files or transfer maps. ***Note:*** The Design access right can be applied only to a Lab Loader protocol.
Transfer	Transfer records from a Lab Loader protocol into the clinical data protocol on which this right will be set, and manage duplicates. Granting this option allows the user to transfer data to any panel.	Full — Can transfer records, view and purge the Transfer Log, and view maps and reports. Read — Can view transfer maps, the Transfer Log, and Duplicates reports. None — Cannot transfer records. ***Note:*** The Transfer access right cannot be applied to a view protocol or Lab Loader protocol.

Note: Manage access rights control access to Lab Loader activities that are also accessible from Manage.

Classify protocol access right

The following table lists the Classify protocol access right:

Access right:	User can:	Access levels:
Thesaurus	Search for terms or codes in a coding thesaurus. Run thesaurus reports. Test thesaurus algorithms. Edit support elements for transformation.	Full — Can perform all the listed tasks. Read — Can search for terms or codes, run reports, and test thesaurus algorithms, but cannot edit support elements. None — Cannot perform any of the listed tasks. ***Note:*** The Thesaurus access right applies to coding thesaurus protocols only.

Note: For access to the **Thesaurus** and **Algorithm** menu commands in Classify, the user must also have the Design System access right.

Access rights for Clintrial software-supplied usergroups

The Clintrial software supplies three predefined usergroups:

- ADMIN
- ANALYSIS
- ENTRY

List of access rights for usergroups

The following table lists the Clintrial software-supplied usergroups and their default access rights and access levels:

Clintrial software-supplied usergroup:	Default access rights:	Default access level:
ADMIN	Non-protocol access rights:	
	Design — Global, System	Full
	Admin — System, DBA	Full
	Tools — SQL	Read
ANALYSIS	Protocol access rights:	
	Retrieve — Merged	Read
	Library	Write
ENTRY	Protocol access rights:	
	Enter — Unmerged	Full

You can add users to or delete users from these usergroups, and you can change the access rights and access levels for the usergroups. You can also delete these usergroups.

For information on Clintrial software usergroups, see "Usergroups" on page 25.

Access to protected panels

Protocol access rights are for all panels in a protocol. In some cases the designer may want to limit access rights to certain panels. Access to a panel that has the Protected attribute set is limited. A user or usergroup with the following protocol access rights cannot exercise these rights in a protected panel, unless specifically authorized in Admin:

- Enter — Merged or Unmerged
- Retrieve — Merged or Unmerged
- Manage — Coding, Global and Other
- Tools — SQL
- Resolve — Create, Propose, Manage and Produce.

- Lab Loader — Design and Transfer

By default, each of these access rights has the access level None if the panel is protected. You can override the default and allow these access rights in the same way that you allow access rights for users and usergroups for the protocol.

Other protocol access rights, Enter - Enroll, and Retrieve - Library, work normally on both protected and non-protected panels.

Only panels to which a user has the proper rights will be displayed in dialogs.

For information on panel attributes, see Chapter 10.

Lab Loader Protected Panel access

In order to **create**, **edit**, **copy** and **delete maps**, the user must have **Full** protocol rights to the **Source protocol** and **Full Lab Loader Design** rights to protected panels in the Source protocol, as well as at least **Read** protocol rights to the **Destination protocol** and at least **Read Lab Loader Transfer** rights to the protected panels in the Destination protocol.

A minimum of **Read** rights to the protocols are required to **show transfer maps**, if the map refers to protected panels, you need at least **Read Lab Loader Design** and **Transfer** rights to protected panels in the protocols.

If the panel in the Source protocol is protected, in order to actually **transfer data** the user must have at least **Read** rights to the **Source protocol** and at least **Read Lab Loader Design** rights to protected panels in the Source protocol. **Full** rights to the **Destination protocol** and **Full Lab Loader Transfer** rights to protected panels in the Destination protocol are also required.

This may be summarized in the following table:

Tasks	Protocol Rights	Panel Lab Loader Rights Required if Protected
Create, Edit, Copy or Delete Maps	Design: Full on Source Transfer: Read on Destination	Design: Full on Source Transfer: Read on Destination
Show Maps	Design: Read on Source Transfer: Read on Destination	Design: Read on Source Transfer: Read on Destination

Tasks	Protocol Rights	Panel Lab Loader Rights Required if Protected
Transfer Data	Design: Read on Source	Design: Read on Source
	Transfer: Full on Destination	Transfer: Full on Destination

Manage Protected Panel access

Note: There is no restriction for Flags and Notes, since there are no access rights for Flags and Notes

Note: All logs and control files are not included. Some log files and control files can be saved as external files. You should protect them from the Windows level.

Resolve Protected Panel access

Discrepancy count reports will not exclude data on protected panels since only counts, not actual data, can be seen. To help avoid confusion, the message "May include data for protected panels" will be put on any report run in a protocol that contains any protected panels.

Access rights and access audit reports

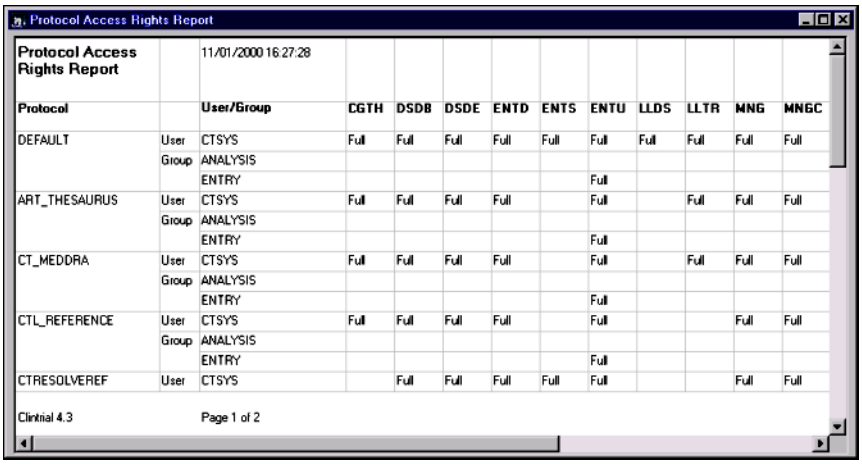
Clintrial software provides six access rights and access audit reports:

- Protocol Access Rights Report
- Protocol Access Audit Report
- Non-Protocol Access Rights Report
- Non-Protocol Access Audit Report
- Panel Access Rights Report
- Panel Access Audit Report

The access rights reports are created in Admin under the **Security** menu. The audit reports are created in Admin under the **Audit** menu.

Protocol Access Rights Report

The Protocol Access Rights Report provides a listing of user and usergroup protocol access rights. It is created in Admin under the **Security** menu. The following figure shows part of a Protocol Access Rights Report:



Protocol Access Rights Report

11/01/2000 16:27:28

Protocol		User/Group	CGTH	DSDB	DSDE	ENTD	ENTS	ENTU	LLDS	LLTR	MNG	MNGC
DEFAULT	User	CTSYS	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
	Group	ANALYSIS										
		ENTRY						Full				
ART_THESAURUS	User	CTSYS	Full	Full	Full	Full		Full		Full	Full	Full
	Group	ANALYSIS										
		ENTRY						Full				
CT_MEDDRA	User	CTSYS	Full	Full	Full	Full		Full		Full	Full	Full
	Group	ANALYSIS										
		ENTRY						Full				
CTL_REFERENCE	User	CTSYS	Full	Full	Full	Full		Full			Full	Full
	Group	ANALYSIS										
		ENTRY						Full				
CTRESOLVEREF	User	CTSYS		Full	Full	Full	Full	Full			Full	Full

Clintrial 4.3 Page 1 of 2

Note: The scroll bar at the bottom indicates that access rights for other modules are off the screen to the right.

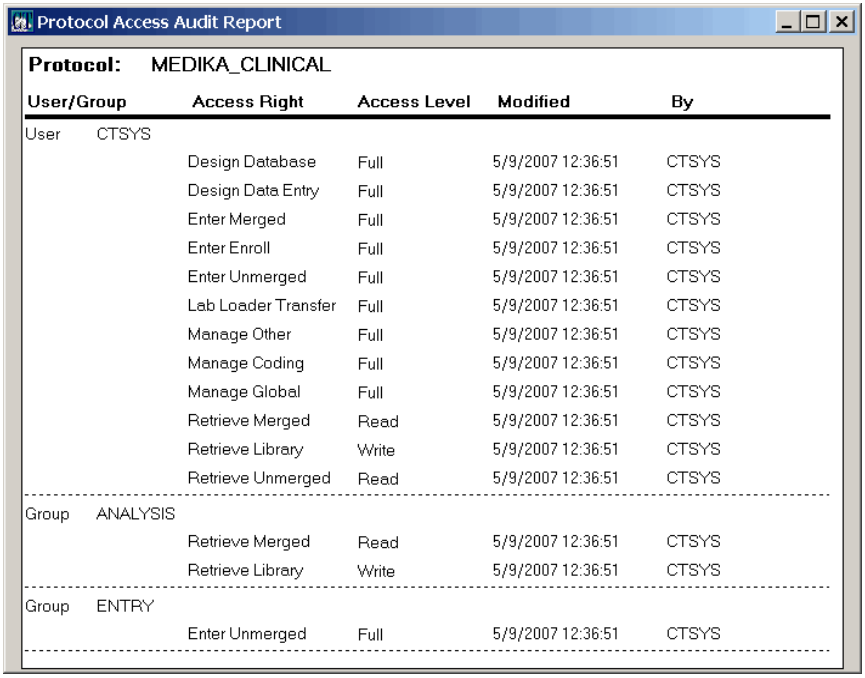
The following table lists the information in the Protocol Access Rights Report:

Report column:	**Description:**
Protocol	Name of each protocol in the database.
User/Group	Names of each user and each usergroup.
Access rights and levels	Access rights and access levels for each user or usergroup.

Note: If an access right has not been granted to any user, no column for that access right appears in the report.

Protocol Access Audit Report

The Protocol Access Audit Report shows the history of the modification of user or usergroup protocol access rights since the installation or upgrade to Clintrial 4.7. Upon choosing the report under the **Audit** menu, a dialog box allowing the selection of a single protocol appears. The following example shows part of a Protocol Access Audit Report:



Protocol Access Audit Report

Protocol: MEDIKA_CLINICAL

User/Group		Access Right	Access Level	Modified	By
User	CTSYS				
		Design Database	Full	5/9/2007 12:36:51	CTSYS
		Design Data Entry	Full	5/9/2007 12:36:51	CTSYS
		Enter Merged	Full	5/9/2007 12:36:51	CTSYS
		Enter Enroll	Full	5/9/2007 12:36:51	CTSYS
		Enter Unmerged	Full	5/9/2007 12:36:51	CTSYS
		Lab Loader Transfer	Full	5/9/2007 12:36:51	CTSYS
		Manage Other	Full	5/9/2007 12:36:51	CTSYS
		Manage Coding	Full	5/9/2007 12:36:51	CTSYS
		Manage Global	Full	5/9/2007 12:36:51	CTSYS
		Retrieve Merged	Read	5/9/2007 12:36:51	CTSYS
		Retrieve Library	Write	5/9/2007 12:36:51	CTSYS
		Retrieve Unmerged	Read	5/9/2007 12:36:51	CTSYS
Group	ANALYSIS				
		Retrieve Merged	Read	5/9/2007 12:36:51	CTSYS
		Retrieve Library	Write	5/9/2007 12:36:51	CTSYS
Group	ENTRY				
		Enter Unmerged	Full	5/9/2007 12:36:51	CTSYS

Note: Auditing information is not always available.

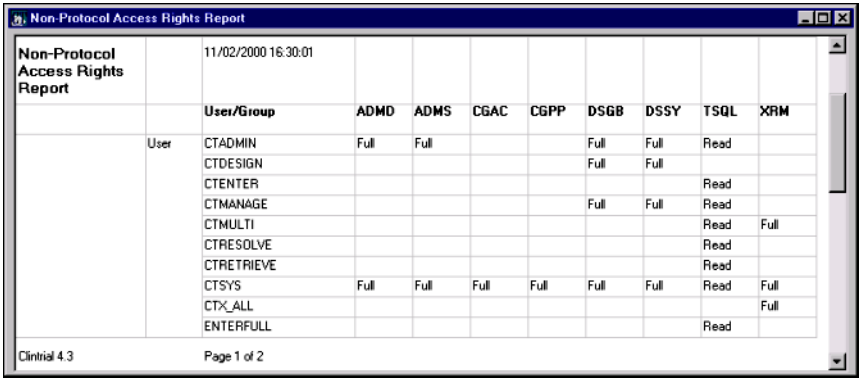
- When a protocol is deleted, all access auditing information is deleted.
- If a protocol with the same name is deleted and then recreated, its audit history starts from the recreation.
- When a protocol is imported, its access audit history starts from the time of import.

The Protocol Access Audit Report contains the following information about accounts:

Report column:	Description:
User/Group	Names of each user and each usergroup.
Access Right	Protocol Access Rights.
Access Level	The access level assigned to the user or usergroup for the Access Right listed on the date specified.
Modified	Date and time of modification.
By	Name of user account that modified the Access Level.

Non-Protocol Access Rights Report

The Non-Protocol Access Rights Report provides a listing of user or usergroup non-protocol access rights. It is created in Admin under the **Security** menu. The following figure shows a Non-Protocol Access Rights Report:


Non-Protocol Access Rights Report

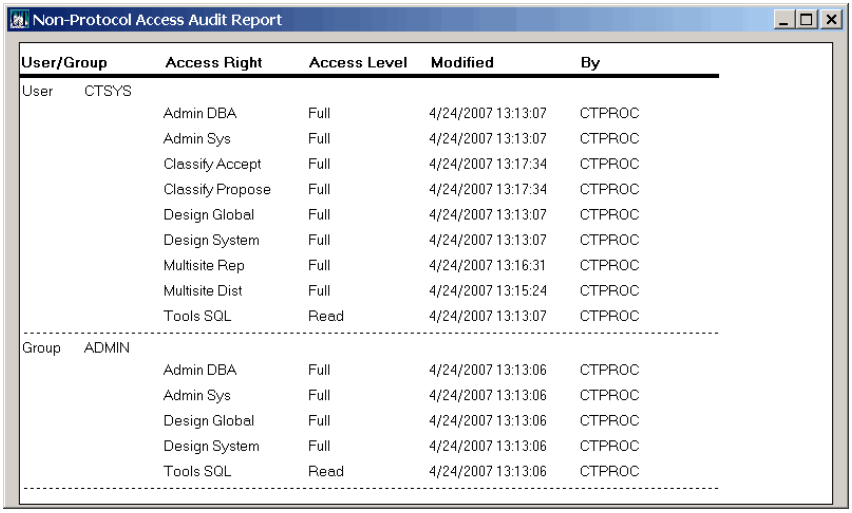
Non-Protocol Access Rights Report		11/02/2000 16:30:01								
		User/Group	**ADMD**	**ADMS**	**CGAC**	**CGPP**	**DSGB**	**DSSY**	**TSQL**	**XRM**
	User	CTADMIN	Full	Full			Full	Full	Read	
		CTDESIGN					Full	Full		
		CTENTER							Read	
		CTMANAGE					Full	Full	Read	
		CTMULTI							Read	Full
		CTRESOLVE							Read	
		CTRETRIEVE							Read	
		CTSYS	Full	Full	Full	Full	Full	Full	Read	Full
		CTX_ALL								Full
		ENTERFULL							Read	
Clintrial 4.3		Page 1 of 2								

The following table lists the information in the Non-Protocol Access Rights Report:

Report column:	Description:
User/Group	Name of each user and each usergroup.
Access rights and levels	Access rights and access levels for the user or usergroup.

Non-Protocol Access Audit Report

The Non-Protocol Access Audit Report shows the history of the modification of user or usergroup non-protocol access rights since the installation or upgrade to Clintrial 4.7 It is created in Admin under the **Audits** menu. The following example shows part of a Non-Protocol Access Audit Report:



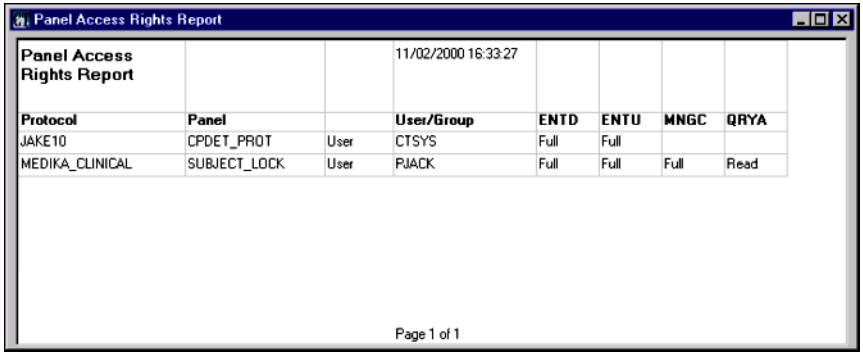
User/Group		Access Right	Access Level	Modified	By
User	CTSYS				
		Admin DBA	Full	4/24/2007 13:13:07	CTPROC
		Admin Sys	Full	4/24/2007 13:13:07	CTPROC
		Classify Accept	Full	4/24/2007 13:17:34	CTPROC
		Classify Propose	Full	4/24/2007 13:17:34	CTPROC
		Design Global	Full	4/24/2007 13:13:07	CTPROC
		Design System	Full	4/24/2007 13:13:07	CTPROC
		Multisite Rep	Full	4/24/2007 13:16:31	CTPROC
		Multisite Dist	Full	4/24/2007 13:15:24	CTPROC
		Tools SQL	Read	4/24/2007 13:13:07	CTPROC
Group	ADMIN				
		Admin DBA	Full	4/24/2007 13:13:06	CTPROC
		Admin Sys	Full	4/24/2007 13:13:06	CTPROC
		Design Global	Full	4/24/2007 13:13:06	CTPROC
		Design System	Full	4/24/2007 13:13:06	CTPROC
		Tools SQL	Read	4/24/2007 13:13:06	CTPROC

The Non-Protocol Access Audit Report contains the following information about accounts:

Report column:	**Description:**
User/Group	Names of each user and each usergroup.
Access Right	Protocol Access Rights.
Access Level	The access level assigned to the user or usergroup for the Access Right listed on the date specified.
Modified	Date and time of modification.
By	Name of user account that modified the Access Level.

Panel Access Rights Report

The Panel Access Rights Report provides a listing of user or usergroup panel access rights. It is created in Admin under the **Security** menu. The following figure shows a Panel Access Rights Report for the protected SUBJECT_LOCK panel in MEDIKA_CLINICAL:

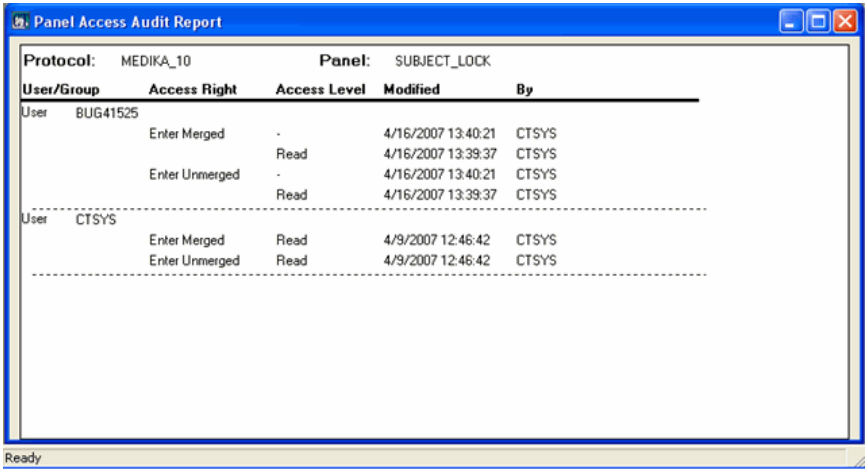


The following table lists the information in the Panel Access Rights Report:

Report column:	**Description:**
Protocol	Name of the protocol containing the protected panel.
Panel	Name of the protected panel.
User/Group	Name of each user and each usergroup.
Access rights and levels	Access rights and access levels for each user or usergroup. By default, each of these access rights has the access level None if the panel is protected. You can override the default and allow these access rights in the same way that you allow access rights for users and usergroups for the protocol.

Panel Access Audit Report

The Panel Access Audit Report shows the history of the modification of user or usergroup protected access rights since the installation or upgrade to Clintrial 4.7. Upon choosing the report under the **Audit** menu, a dialog box allowing the selection of a single protected panel appears. The following example shows part of a Panel Access Audit Report:



Note: If a panel is de-installed or deleted, then re-installed or recreated, the Panel Access Audit Report shows that all rights were revoked. Any rights granted again also appear on the report. There is no "Deleted" designation.

The Panel Access Audit Report contains the following information about the selected panel:

Report column:	**Description:**
User/Group	Names of user and each usergroup.
Access Right	Panel Access Rights.
Access Level	The access level assigned to the user or usergroup for the Access Right listed on the date specified.
Modified	Date and time of modification.
By	Name of user account that modified the Access Level.

5 Parameters

System parameters

System parameters define characteristics of the work environment for all users of an Oracle database instance to which the users connect through the Clintrial software. One example of a system parameter is PASSWORD_MINIMUM, which determines the minimum length for passwords used in the Clintrial software.

How to set system parameters

You set system parameters from a user account that has the Admin System access right.

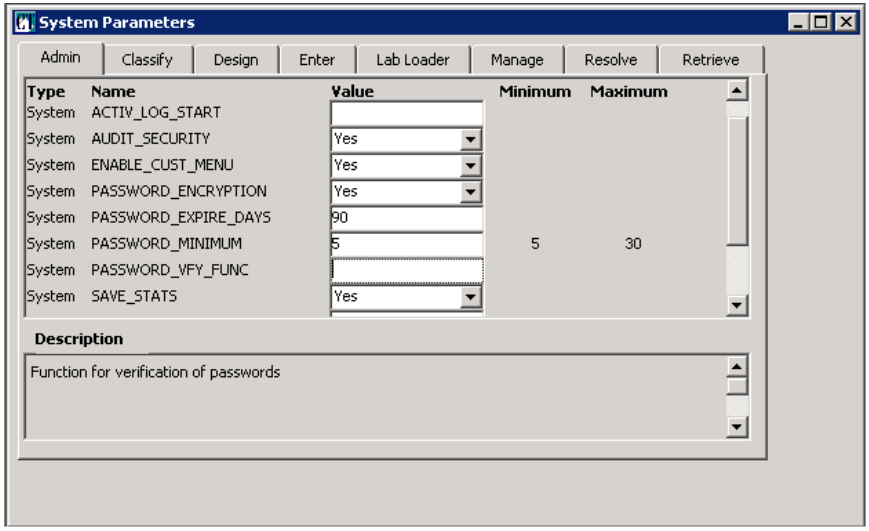
In Admin, system parameters are grouped by the module to which they pertain:

- Admin (see page 72)
- Classify (see page 74)
- Design (see page 75)
- Enter (see page 77)
- Lab Loader (see page 77)
- Manage (see page 78)
- Resolve (see page 79)
- Retrieve (see page 80)

Note: The Classify, Lab Loader, and Resolve parameters appear only if those modules are installed on the server.

The System Parameters window lists the system parameters, protocol parameters, and user preferences for each module that is installed, and provides the type and a description for each parameter. To access the System Parameters window, from the **Parameters** menu, select **System**.

The following figure shows the System Parameters window, with the Admin system parameters selected:



Some system parameters set defaults for protocol parameters and user preferences. A Design user can override the defaults for a protocol by changing the protocol parameters. Individual users of all modules can override the defaults for some of their system parameters by changing their user preferences.

For more information on system parameters, see "Protocol parameters" on page 82.

For more information on system parameters, see "User preferences" on page 90.

Admin system parameters

The following table lists the system parameters that affect Admin:

System parameter:	Description:
ACTIV_LOG_END Default: None	Date on which to stop running the System Activity log, which tracks menu commands that are used in all Clintrial software modules.
ACTIV_LOG_START Default: None	Date on which to start running the System Activity log, which tracks menu commands that are used in all Clintrial software modules.

System parameter:	Description:
AUDIT_SECURITY Default: Yes	If Yes, user and usergroup auditing and user and usergroup access auditing are turned on. If No, user and usergroup auditing and user and usergroup access auditing are turned off. These reports are controlled by this parameter: • User Audit Report • User Access Audit Report • Usergroup Audit Report • Usergroup Access Audit Report • Protocol Access Audit Report • Non-Protocol Access Audit Report • Panel Access Audit Report ***Note:*** If you decide to turn audit security off, then you should find an alternative method of auditing users and usergroups. ***Note:*** Toggling this parameter on and off will cause gaps in the audit trails for these reports. This is not recommended.
ENABLE_CUST_MENU Default: No	If Yes, the Custom Menu is enabled and visible. If No, the Custom Menu is disabled and not visible.
PASSWORD_ENCRYPTION Default: No	If Yes, passwords for user accounts are encrypted. If No, passwords for user accounts are not encrypted.
PASSWORD_VFY_FUNC Default: Blank	Name of the password verification function called by Clintrial. If a function name is listed, the function is applied to all users by Clintrial during the creation and modification of passwords. If blank, no password verification function is called by Clintrial.
PASSWORD_EXPIRE_DAYS Default: None	Number of days until expiration of the password after creation or modification. If blank, password does not expire.

System parameter:	Description:
PASSWORD_MINIMUM Default: 6	The minimum length of passwords for user accounts. Allowable range is 5-30.
SAVE_STATS Default: Yes	If Yes, statistics can be saved while the Database Space Utilization report is running. If No, statistics cannot be saved while the Database Space Utilization report is running.
USR_DB_PROFILE Default: DEFAULT	Name of the default database profile for user accounts.
USR_DFT_TSP Default: CT_USERS	Name of the default tablespace for user accounts.
USR_TMP_TSP Default: TEMP	Name of the default temporary tablespace for user accounts.

Classify system parameters

The following table lists the system parameters that affect Classify:

System parameter:	Description:
CTG_ALLOW_ CHANGEVERB Default: Yes	If Yes, proposals to change verbatim text are allowed. The **Propose >> Change Verbatim** menu item is enabled. If No, proposals to change verbatim text are not allowed. The **Propose >> Change Verbatim** menu item is suppressed.

System parameter:	Description:
CTG_AUTOACCEPT Default: No	If Yes, proposals are accepted automatically if defined by any user whose access level is Full for the Propose access right, and either Full or No Delete for the Accept access right. If No, proposals are not accepted automatically. **Note:** If you set this parameter to Yes, you will speed the omission resolution process, but forfeit the chance to review proposals before you finalize them.
CTG_EDIT_SYNTEXT Default: No	If Yes, normalized test strings can be edited before saving them as synonyms. If No, synonym text cannot be edited. **Note:** If you set this parameter to Yes, some Synonym solution omission records may fail to code successfully.

Design system parameters

The following table lists the system parameters that affect Design:

System parameter:	Description:
AUDIT_INVNOTES Default: Yes	If Yes, investigator notes are audited. If No, investigator notes are not audited. Investigator notes are notes with the category INVESTIGATOR.
AUDIT_PROTOCOL_MOD Default: Yes	If Yes, a protocol-specific audit start point can be set when a protocol is created or modified. If No, a protocol-specific audit start point cannot be set.
AUDIT_SPONSORNOTES Default: Yes	If Yes, sponsor notes are audited. If No, sponsor notes are not audited. Sponsor notes are notes with the category SPONSOR.

System parameter:	Description:
AUDIT_START_DEFAULT Default: MERGE	Default audit start point for all protocols in the database instance: ENTRY, VERIFICATION, VALIDATION, VALIDITY, or MERGE.
CC_HOST Default: None	The HTTP address of the Central Coding interface, including the **Interface Name** used when installing Central Coding. This is the format: NetworkMachineName/InterfaceName For example: rdcntlcdq027/codingserviceinterfacedev1q027
CODE_LABELS_MOD Default: Yes	If Yes, codelist labels and long labels can be changed when the codelist is being used. If No, codelist labels and long labels cannot be changed when the codelist is being used.
PROC_SITE_ACCOUNT Default: CTSITEPROC	Name of the Oracle account that owns site-specific PL/SQL procedures.
PROT_TMP_TSP Default: TEMP	Name of the temporary tablespace for protocol accounts.
USE_CENTRAL_CODING Default: Yes	If Yes, the protocol will be coded using enhanced Central Coding. If No, classic Clintrial coding will be used. If this parameter is changed from No to Yes in an existing protocol: • Targets configured using classic Clintrial coding will have to be reconfigured. • If any panels with Targets are already installed, a “Data Loss” warning will be issued before proceeding. Another warning message will request further confirmation of the parameter change.
XFER_TABLESPACE Default: None	Name of the default tablespace to be used during protocol export.

Enter system parameter

The following system parameter affects Enter:

System parameter:	Description:
ENT_DIR_IN_DATA Default: Yes	If Yes, then data can be entered directly into DATA table using Edit Merged Data mode. If No, then data cannot be entered directly into DATA table using Edit Merged Data mode.
VERIFY_SHOW_TAGS Default: Yes	If Yes, all flags and notes are available during verification. If No, only VERIFICATION flags, and no notes, are available during verification.

Lab Loader system parameter

The following system parameter affects Lab Loader:

System parameter:	Description:
CTL_DUPFLAG_LOC Default: DATA	Indicates whether a duplicate flag found during transfer is attached to the record in the Data table or the Update table.

The following table lists the system parameters that affect Manage:

System parameter:	Description:
ALLOW_MERGE_TO_VALID Default: Yes	If Yes, the Validate First check box in the Merge window is available. If No, the Validate First check box is not available. The Validate First check box allows users to specify that data in panels will be validated before it is merged.
AUDIT_REASON_REQD Default: Yes	If Yes, a reason for change must be specified when changes or deletions are made and auditing is in effect. If No, a reason for change does not need to be specified.
AUTOCODE_FULL_CNT Default: No	If Yes, NUM_MATCHES is set to the actual number of matches found during automatic coding. If No, NUM_MATCHES is set to >1 if more than one match is found during automatic coding.
AUTOCODE_SET_FAIL Default: Yes	If Yes, the workflow item is set to FAIL if no match is found during automatic coding. If No, the workflow item is set to AUTO whether a match is found.

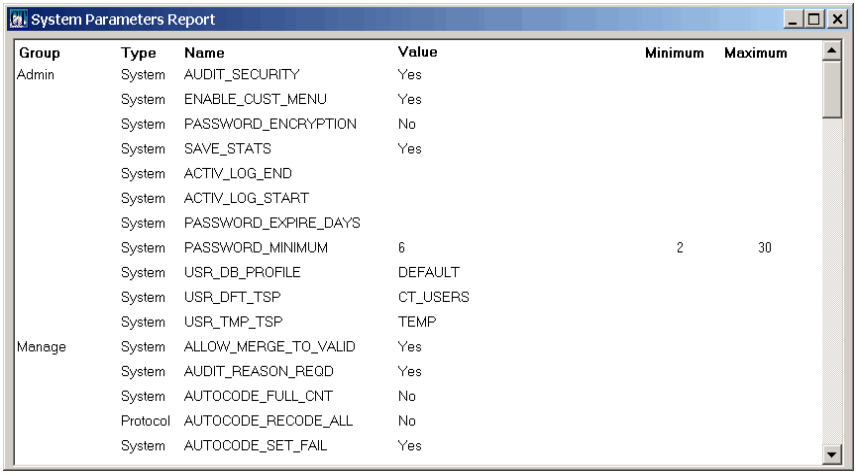
Resolve system parameters

The following table lists the system parameters that affect Resolve:

System parameter:	Description:
CTV_APPL_RES_CODE Default: 1	Determines the default Reason for Change Code Value in the discrepancy record when the **Apply Proposed Value** command is used. To define a value for this parameter, you select from the CTS_REASON_CODES codelist supplied with the core modules. The CTS_REASON_CODES codelist is initially empty, and can be modified in Design.
CTV_AUTOCLOSED_RES Default: None	Determines the default Reason for Change Code Value in the discrepancy record when the discrepancy is automatically closed because of data change. To define a value for this parameter, you select from the CTS_REASON_CODES codelist supplied with the core modules. The CTS_REASON_CODES codelist is initially empty, and can be modified in Design.
CTV_OBSOLETE_RES Default: None	Determines the default Reason for Change Code Value in the discrepancy record when the discrepancy is automatically closed because of data deletion or data model changes. To define a value for this parameter, you select from the CTS_REASON_CODES codelist supplied with the core modules. The CTS_REASON_CODES codelist is initially empty, and can be modified in Design.
CTV_USE_ERRORLOG Default: Yes	Determines whether both an Error Log entry and a Resolve discrepancy record are created when validation runs. If Yes, validation creates both; if No, validation creates only a Resolve discrepancy record.

System Parameters Report

Admin provides a System Parameters Report in the **Reports** menu. It shows the current values of the system parameters. The report is sorted by Group and then Name.



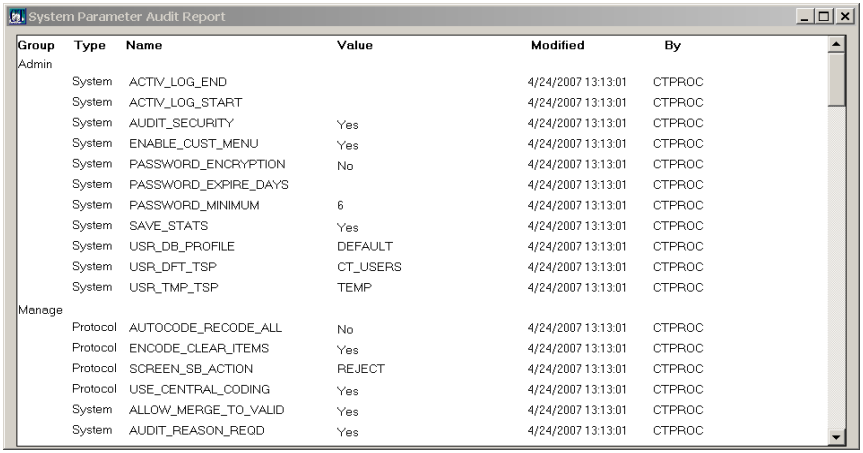
Group	Type	Name	Value	Minimum	Maximum
Admin	System	AUDIT_SECURITY	Yes		
	System	ENABLE_CUST_MENU	Yes		
	System	PASSWORD_ENCRYPTION	No		
	System	SAVE_STATS	Yes		
	System	ACTIV_LOG_END			
	System	ACTIV_LOG_START			
	System	PASSWORD_EXPIRE_DAYS			
	System	PASSWORD_MINIMUM	6	2	30
	System	USR_DB_PROFILE	DEFAULT		
	System	USR_DFT_TSP	CT_USERS		
	System	USR_TMP_TSP	TEMP		
Manage	System	ALLOW_MERGE_TO_VALID	Yes		
	System	AUDIT_REASON_REQD	Yes		
	System	AUTOCODE_FULL_CNT	No		
	Protocol	AUTOCODE_RECODE_ALL	No		
	System	AUTOCODE_SET_FAIL	Yes		

The System Parameters Report contains the following system parameter information:

Report column:	**Description:**
Group	Name of the Clintrial software module in which the parameter operates: Admin, Manage, Design, Enter, Retrieve, Resolve, Classify, or Lab Loader.
Type	Type of parameter: Protocol, System or User.
Name	Name of the system parameter.
Value	A value, if any, for the system parameter.
Minimum	A minimum value, if any, for the system parameter.
Maximum	A maximum value, if any, for the system parameter.

System Parameter Audit Report

Admin provides a System Parameter Audit Report in the **Audits** menu. It shows the history of current and previous values of the system parameters since the installation or upgrade to Clintrial 4.7. The report is sorted by Group, Name and most recent modification date.



Group	Type	Name	Value	Modified	By
Admin					
	System	ACTIV_LOG_END		4/24/2007 13:13:01	CTPROC
	System	ACTIV_LOG_START		4/24/2007 13:13:01	CTPROC
	System	AUDIT_SECURITY	Yes	4/24/2007 13:13:01	CTPROC
	System	ENABLE_CUST_MENU	Yes	4/24/2007 13:13:01	CTPROC
	System	PASSWORD_ENCRYPTION	No	4/24/2007 13:13:01	CTPROC
	System	PASSWORD_EXPIRE_DAYS		4/24/2007 13:13:01	CTPROC
	System	PASSWORD_MINIMUM	6	4/24/2007 13:13:01	CTPROC
	System	SAVE_STATS	Yes	4/24/2007 13:13:01	CTPROC
	System	USR_DB_PROFILE	DEFAULT	4/24/2007 13:13:01	CTPROC
	System	USR_DFT_TSP	CT_USERS	4/24/2007 13:13:01	CTPROC
	System	USR_TMP_TSP	TEMP	4/24/2007 13:13:01	CTPROC
Manage					
	Protocol	AUTOCODE_RECODE_ALL	No	4/24/2007 13:13:01	CTPROC
	Protocol	ENCODE_CLEAR_ITEMS	Yes	4/24/2007 13:13:01	CTPROC
	Protocol	SCREEN_SB_ACTION	REJECT	4/24/2007 13:13:01	CTPROC
	Protocol	USE_CENTRAL_CODING	Yes	4/24/2007 13:13:01	CTPROC
	System	ALLOW_MERGE_TO_VALID	Yes	4/24/2007 13:13:01	CTPROC
	System	AUDIT_REASON_REQD	Yes	4/24/2007 13:13:01	CTPROC

The System Parameter Audit Report contains the following system parameter information:

Report column:	**Description:**
Group	Name of the Clintrial software module in which the parameter operates: Admin, Manage, Design, Enter, Retrieve, Resolve, Classify, or Lab Loader.
Type	Type of parameter: Protocol, System or User.
Name	Name of the system parameter.
Value	The value, if any, for the system parameter.
Modified	Date and time of modification.
By	Name of user account that modified the parameter.

Note: The *Protocol Parameter Audit Report* for auditing individual protocols is available in the Design module. It is described in "Protocol Parameter Audit" on page 500 in Chapter 23.

Protocol parameters

You can set protocol parameters for the following modules:

- Design
- Enter
- Manage
- Resolve

What are protocol parameters?

Protocol parameters are a subset of system parameters that can be defined by the Design user (with the System access right) for a specific protocol, to override the default established by the system parameter.

For example, the CONNECT_REQD_DD system parameter can be set in Admin to Yes or No, to specify whether all objects must be connected when copied. CONNECT_REQD_DD is also a protocol parameter; therefore, in Design you can override the system parameter setting for each protocol.

Note: The *Protocol Parameters Audit Report* for auditing individual protocols is available in the Design module. It is described in "Protocol Parameter Audit" on page 500 in Chapter 23.

How to set protocol parameters

You set default protocol parameters by setting system parameters in Admin from a user account that has the Admin System access rights. Design users with System access rights can change the default protocol parameter settings. With the Design Protocol Browser displayed, from the **Protocol** menu, select **Parameters**, and set any protocol parameters that you want to change from the default.

Design protocol parameters

The following table lists the protocol parameters that affect Design:

Protocol parameter:	Description:
AUTO_RECODE_ALL Default: No	If Yes, during automatic coding, items that were previously coded are recoded (that is, items for which the value of the workflow item is AUTO or FAIL). Also, if the protocol is set up for overriding of interactively assigned codes, then items for which the workflow is INT are also recoded. If No, items that were previously coded are not recoded.
AUDIT_METADATA Default: No	If Yes, metadata changes to connected objects are audited. If No, metadata changes to connected objects are not audited.
CC_HOST Default: None	The HTTP address of the Central Coding interface, including the **Interface Name** used when installing Central Coding. This is the format: NetworkMachineName/InterfaceName For example: rdcntlcdq027/codingserviceinterfacedev1q027
CONNECT_REQD_DD Default: No	If Yes, source and destination schema objects must be connected during copy operations. If No, schema object connections are optional.
CONNECT_REQD_FRM Default: No	If Yes, source and destination display objects must be connected during copy operations. If No, display object connections are optional.
CONNECT_REQD_VLD Default: No	If Yes, source and destination validation objects must be connected during copy operations. If No, validation object connections are optional.

Protocol parameter:	Description:
CREATE_OBJ_DD Default: Yes	If Yes, schema objects can be created. If No, schema objects can only be copied.
CREATE_OBJ_FRM Default: Yes	If Yes, display objects can be created. If No, display objects can only be copied.
CREATE_OBJ_VLD Default: Yes	If Yes, validation objects can be created. If No, validation objects can only be copied.
CTG_MEDDRA_TERM_COL Default: None	Name of column in LOW_LEVEL_TERM table containing LLT text to use for extended MEDDRA thesaurus (instead of LLT_NAME).
CTG_THES_TYPE Default: None	Tells Classify the type of extended thesaurus protocol to use. Choices are None, MEDDRA, WHODD.
DD_AUTOVALIDATE Default: No	If Yes, automatic revalidation of previously validated records in the UPDATE table occurs after panel revision. If No, revalidation must be done manually.
DD_STRICT_CODE Default: Yes	If Yes, an installed item's codelist attribute can only be changed to be a less restrictive subset. If No, an installed item's codelist attribute can be modified or removed without restriction. ***Note*:** For items created using connected copy, this parameter setting has no effect.
DD_STRICT_SEARCH Default: Yes	If Yes, only direct ancestors (through the parent hierarchy), or siblings (protocols with the same parent), can be added to the protocol's searchlist. If No, any protocol in the database can be added to the protocol's searchlist.
PROC_ACCOUNT Default: None	Name of the Oracle account that owns protocol-specific PL/SQL procedures.

Protocol parameter:	Description:
USE_CENTRAL_CODING Default: Yes	If Yes, the protocol will be coded using enhanced Central Coding. If No, classic Clintrial coding will be used. If this parameter is changed from No to Yes in an existing protocol: • Targets configured using classic Clintrial coding will have to be reconfigured. • If any panels with Targets are already installed, a "Data Loss" warning will be issued before proceeding. Another warning message will request further confirmation of the parameter change.

Enter protocol parameters

The following table lists the protocol parameters that affect Enter:

Protocol parameter:	Description:
DD_BLANKS_IN_PATIENT Default: No	If Yes, blank spaces can be included in the value for the subject item. If No, blank spaces cannot be included in the value for the subject item.
VERIFY_BLIND Default: No	If Yes, verification (double-entry) is blind. If No, verification is interactive; that is, differences between original data and re-entered data are displayed.

The following table lists the protocol parameters that affect Manage:

Protocol parameter:	Description:
AUTOCODE_RECODE_ALL Default: No	If Yes, during automatic coding, items that were previously coded are recoded (that is, items for which the value of the workflow item is AUTO or FAIL). Also, if the protocol is set up for overriding of interactively assigned codes, then items for which the workflow is INT are also recoded. If No, items that were previously coded are not recoded.
ENCODE_CLEAR_ITEMS Default: Yes	If Yes, when verbatim text for a coded item is changed, then the coding-related items are cleared. If No, the coding-related items are not cleared.
SCREEN_SB_ACTION Default: REJECT	Determines the error that occurs when you screen data and the study book is not defined. Possible actions: REJECT (Default) and REPORT. If the action is REJECT, then the following events occur if the study book is not defined: • The record status changes to indicate failure. • The entry in the Error Log shows REJECT in the Action column. If the action is REPORT, then the following events occur if the study book is not defined: • The record status changes to indicate that the record passed screening. • The entry in the Error Log shows REPORT in the Action column.

Protocol parameter:	Description:
USE_CENTRAL_CODING Default: Yes	If Yes, the protocol will be coded using enhanced Central Coding. Additionally: If No, classic Clintrial coding will be used. If this parameter is changed from No to Yes in an existing protocol: • Targets configured using classic Clintrial coding will have to be reconfigured. • If any panels with Targets are already installed, a "Data Loss" warning will be issued before proceeding. Another warning message will request further confirmation of the parameter change.

Resolve protocol parameters

The following table lists the protocol parameters that affect Resolve:

Protocol parameter:	Description:
CTV_FIRST_HEADER_PR Default: None	First line to print on data discrepancy forms. If this protocol parameter is set, it overrides the CTV_FIRST_HEADER user preference.

Protocol parameter:	Description:
CTV_ONE_INVESTIGATOR	SQL fragment to use within a selection statement to get the investigator information from the a field in the discrepancies table (VCT_ERRORSTATUS_UPDATE). Default: CT_RESOLVE_USER.INV_ID(VCT_ERRORSTATUS_UPDATE.CTV_CONTEXT1, '<PROTOCOL>') If there is a simple algorithm for computing the investigator information during retrieval, use that as the value. For example: • If the investigator is the first four characters of the subject item, set this parameter to SUBSTR(VCT_ERRORSTATUS_UPDATE.CTV_CONTEXT1,1,4). • If the protocol contains a context item that holds the investigator identifier, set the parameter to the name of that item preceded by VCT_ERRORSTATUS_UPDATE, for example, VCT_ERRORSTATUS_UPDATE.INVNO. ***Note:*** By editing this parameter to avoid the call to CT_RESOLVE_USER.INV_ID, you can speed processing time. Or, you can edit CT_RESOLVE_USER.INV_ID for more complicated customization of the retrieval of investigator information. For more information, see *Enter, Resolve and Retrieve*.
CTV_REQUESTER_PR Default: None	Name of the user who requests proposed resolutions from investigational sites. This name is included on all data discrepancy forms printed for the protocol. If set, it overrides the CTV_REQUESTER user preference.
CTV_SECOND_HEADER_PR Default: None	Second line to print on data discrepancy forms. If this protocol parameter is set, it overrides the CTV_SECOND_HEADER user preference.

Protocol parameter:	Description:
CTV_SELECT_INVESTIGA	SQL statement for retrieving a list of investigators as text. Required for the **View** menu's **Select By** command and for the **Discrepancy** menu's **New** command. Default: SELECT UNIQUE CT_RESOLVE_USER.INV_ID(<SUBJECT_ITEM>, '<PROTOCOL>')FROM <PROTOCOL>.SUBSTITUTE*PANEL_ SUBSTITUTE*DBTABLE. ***Note:*** If investigators are stored in a separate context item, then replace the call to the function CT_RESOLVE_USER.INV_ID with the name of the item to speed up processing time. ***For example:*** If the item name is **INVESTIGATOR**, the statement may be SELECT UNIQUE **INVESTIGATOR** FROM (PROTOCOL>.SUBSTITUTE*PANEL_SUBSTITUTE*DBTABLE ***Note:*** If you edit the CT_RESOLVE_USER.INV_ID function, this parameter retrieves investigator information based on that customization.
CTV_SELECT_SUBJECTS	SQL statement for retrieving a list of subjects as text. Required for the **View** menu's **Select By** command and the **Discrepancy** menu's **New** command. Default: SELECT UNIQUE **<SUBJECT_ITEM>** FROM<PROTOCOL>.SUBSTITUTE*PANEL_ SUBSTITUTE*DBTABLE WHERE CT_RESOLVE_ USER.INV_ID(<SUBJECT_ITEM>, '<PROTOCOL>') =SUBSTITUTE*INVESTIGATOR' ***Note:*** If investigators are stored in a separate context item, then replace the call to the function CT_RESOLVE_USER.INV_ID with the name of the item to speed up processing time. ***For example:*** If the item name is **CENTER**, the statement may be SELECT UNIQUE (SUBJECT_ITEM> FROM(PROTOCOL>.SUBSTITUTE*PANEL_SUBSTITUTE*DBTABLE WHERE **CENTER** = 'SUBSTITUTE*INVESTIGATOR' ***Note:*** If modified, this statement must contain the 'SUBSTITUTE*INVESTIGATOR' clause.

Protocol parameter:	Description:
CTV_SELECT_VISITS	SQL statement for retrieving a list of block items as text. Required for the **View** menu's **Select By** command. Default: SELECT UNIQUE *block_item* FROM *protocol*.SUBSTITUTE*PANEL_SUBSTITUTE* DBTABLE.
CTV_SELECT_VISITS_1S	SQL statement for retrieving a list of block items for a selected subject. Required for the **Discrepancy** menu's **New** command. Default: SELECT UNIQUE *block_item* FROM *protocol*.SUBSTITUTE*PANEL_SUBSTITUTE* DBTABLE WHERE *subject_item* = 'SUBSTITUTE*SUBJECT'.
CTV_STARTUP_VIEW_PR Default: None	Each time Resolve is started, this parameter identifies the view that displays automatically in a Summary window. If this protocol parameter is set, it supersedes the user parameter CTV_STARTUP_VIEW.

User preferences

User preferences are a subset of system parameters. Default values for user preferences come from corresponding system parameters. Each user can change the default values of the parameters for use in their account by changing the user preferences.

For example, the ENT_FORM_INSERT system parameter sets a default specifying when data entered by all data-entry operators is committed. Each data-entry operator can override the default in their user account by setting the corresponding ENT_FORM_INSERT user preference.

How to set user preferences

Users can change the default user preferences from within their user accounts in any Clintrial module to which they have access. From the **File** menu, select **Preferences**, and set any user preferences that you want to change from the default.

Admin user preferences

There are no user preferences for Admin.

Classify user preferences

The following table lists the user preferences that affect Classify:

User preference:	Description:
CTG_GENERAL_RESTRICT Default: None	Restricts the omissions displayed in the Omission Browser to those that are returned by a SQL restriction clause. The SQL restriction clause is entered into the Parameter field and it appears in the Server Restriction field in the Omission Browser. Use of this parameter requires knowledge of table and column names and SQL syntax, and should only be used when the CTG_PANEL_RESTRICT, CTG_PROT_RESTRICT, or CTG_THES_RESTRICT parameters are insufficient.
CTG_PANEL_RESTRICT Default: None	Restricts the omissions displayed in the Omission Browser to those from a specified list of panels. The list of panels is entered into the parameter field and panel names must be separated by commas. The resulting SQL clause restricting the panels appears in the Server Restriction field in the Omission Browser.
CTG_PROT_RESTRICT Default: None	Restricts the omissions displayed in the Omission Browser to those from a specified list of protocols. The list of protocols is entered into the parameter field and protocol names must be separated by commas. The resulting SQL clause restricting the protocols appears in the Server Restriction field in the Omission Browser.

User preference:	Description:
CTG_QUICK_PROPOSE Default: Yes	If Yes, Propose Solution dialog box will automatically close after you select the **File** menu's **Save** command. If No, Propose Solution dialog box must be closed manually.
CTG_REPORT_LIMIT Default: 100	The limit on the number of rows to return from thesaurus reports. If the default is exceeded, the system displays a message that some results are not shown.
CTG_SEARCH_LIMIT Default: 100	The limit on the number of rows to return from thesaurus searches. If the default is exceeded, the system displays a message that some results are not shown.
CTG_SEARCH_WARNING Default: Yes	If Yes, a warning will display when search text begins with a wildcard as the leading character. If No, a warning will not display.
CTG_THES_RESTRICT Default: None	Restricts the omissions displayed in the Omission Browser to those from a specified list of thesauruses. The list of thesauruses is entered into the Parameter field and protocol names must be separated by commas. The resulting SQL clause restricting the thesauruses appears in the Server Restriction field in the Omission Browser.

Design user preferences

The following table lists the Design user preferences:

User preference:	Description:
DD_DBFMT_FIXED Default: NUMBER(5)	Default database format for items of data type FIXED.

User preference:	Description:
DD_DBFMT_FLOAT Default: NUMBER(6,2)	Default database format for items of data type FLOAT.
DD_DBFMT_TEXT Default: VARCHAR2(20)	Default database format for items of data type TEXT.
DD_SASNAME_UPDATE Default: No	If Yes, the default SAS name associated with an item is updated automatically when the item name is modified. If No, the SAS name is not updated automatically.
FRM_CODEENTRY Default: No	If Yes, Enter as Code is checked by default in the Define Item Style window for a field to which a codelist is attached in a page section layout. If No, Enter as Code is not checked by default.
FRM_OVERRIDE Default: No	If Yes, Override is checked by default in the Modify Object Attributes window for a field to which a checklist is attached in a page section layout. If No, Override is not checked by default.
FRM_USE_DESCRIP Default: No	If Yes, the item description displays as the default text label for an item in the page section layout. If No, the item name displays as the default text label.

Enter user preferences

The following table lists the user preferences that affect Enter:

User preference:	**Description:**
ENT_AUTOSKIP_DFLT Default: No	If Yes, the cursor advances automatically to the next enterable field in a study book after the maximum number of characters are entered in the current field. This applies only to items set in Design to perform automatic skipping. If No, the cursor does not go automatically to the next enterable field.
ENT_ENABLE_BEEP Default: Yes	If Yes, an audible sound will occur and a dialog box will open when there is a data entry or verification error. If No, only a dialog box will open.
ENT_ENR_STUDYBOOK Default: None	Name of the enrollment study book for enrolling new subjects when the **File** menu's **New Subject** command is chosen.
ENT_ENTER_AS_TAB Default: No	If Yes, an **Enter** key acts as a **Tab** key. If No, an **Enter** key can only be used in the last field of the page when it moves the cursor forward to the next page.
ENT_FORM_INSERT Default: No	If Yes, data is saved when the user closes a study page. If No, data is saved only when the user explicitly requests that it be saved.
ENT_OPEN_NAVIGATOR Default: Yes	If Yes, the Navigator will be opened automatically after study book selection. If No, the Navigator must be opened manually.

User preference:	Description:
ENT_OPEN_PAGES Default: 4	Number of study pages that can be open at the same time in Enter or Retrieve. Minimum — 1 Maximum — 10
ENT_TAB_PAGE_FORWARD Default: No	If Yes, pressing **Tab** in the last field of the page moves the cursor forward to the next page. If No, pressing **Tab** in the last field of the page moves the cursor to the first field in the same page.

Lab Loader user preference

The following user preference affects Lab Loader:

User preference:	Description:
CTL_TEST_MODE Default: No	If Yes, the Delete Records After Transfer check box in the Start Transfer Window is available. If No, the Delete Records After Transfer check box is not available. Successfully transferred records will be deleted from the source protocol.

Manage user preferences

The following table lists the user preferences that affect Manage:

User preference:	Description:
AUDIT_ITEM_LIST Default: None	Name of an additional context item to be included in the Audit report. (The Subject Item and Block Key Item are included by default.)
ERRLOG_DISPLAY_ITEM Default: None	Name of an additional item to be displayed in the Error Log for Type 0 panels. If empty, no additional items are displayed.

User preference:	Description:
SELECT_BY_PAGE Default: No	If Yes, records are selected by study page for validation and merge. If No, records are selected by panel.

Resolve user preferences

The following table lists the user preferences that affect Resolve:

User preference:	Description:
CTV_ENTER_CODES Default: No	If Yes, enter storage codes in items section of the Detailed window. If No, enter display values in items section of the Detailed window.
CTV_FIRST_HEADER Default: Company Name	First line to print on data discrepancy forms.
CTV_INITIAL_SORT Default: Yes	If Yes, the Summary window will be sorted whenever it is opened. If No, the Summary window will display in whatever order it is retrieved. ***Note***: For large numbers of discrepancies, it may be faster to set this parameter to No.
CTV_REQUESTER Default: None	Name of the user (such as a data manager) who requests discrepancy resolutions from investigational sites. This name is included on all data discrepancy forms.
CTV_SECOND_HEADER Default: Discrepancy Form	Second line to print on data discrepancy forms.

User preference:	Description:
CTV_STARTUP_VIEW Default: ALL	Each time Resolve is started, this parameter identifies the view that displays automatically in a Summary window. You can enter: A for All B for by Batch N for New Discrepancies R for Recently Changed S for Select by 0 (zero) for no startup view

Retrieve user preferences

The following table lists the user preferences that affect Retrieve:

User preference:	Options and meaning:
RTV_DEF_DECODE Default: 1	Codelist column used for decoding of coded items for a Query By Form (QBF) or Ad Hoc Query, or by the CT_DECODE procedure in a Query By SQL: 0 — Code 1 — Value 2 — Short label 3 — Long label
RTV_DEF_SOURCE Default: DATA	Type of clinical data table that QBF and Query By Panel (QBP) queries by default: UPDATE — Update table DATA — Data table ALL — Both the update table and the data table
RTV_INCL_SYS Default: No	If Yes, system items (MERGE_DATETIME, STATUS, ENTRY_ID, ENTRY_DATETIME, CT_RECID, SUBJECT_ID, and DB_ID) are included by default in a QBF and QBP. If No, system items are not included by default.

User preference:	Options and meaning:
RTV_SAS_RECLEN Default: 80	Record length of records generated when query results are saved to a SAS data file. Minimum — 80 Maximum — 1024
SAS_LIBNAME Default: None	Name of the default directory and file in which to store a newly created SAS Format Library.

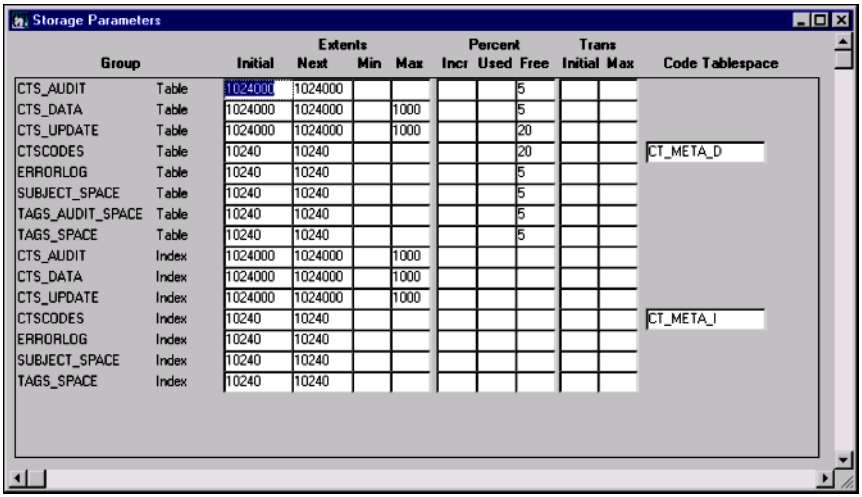
Oracle storage parameters

To analyze how the Oracle database tables and indexes used by the Clintrial software are stored and how they grow, you need two types of information:

- Oracle storage parameters that define the growth of the tables and indexes
- Database space that the tables and indexes occupy

How to set storage parameters

The Oracle database administrator can set default storage parameters for each tablespace of a database. You can use the Storage Parameters window to override the default settings for tables and indexes used by the Clintrial software. The following figure shows the Storage Parameters window in Admin:



The following table lists the storage parameter information:

Report column:	Description:
Group	Name of the group of tables or indexes for which this tablespace is defined.

Report column:	Description:
Extents	Initial — Size, in data blocks, of the first extent allocated when a segment is created. Next — Size, in bytes, of the next incremental extent to be allocated for a segment. Min — Total number of extents to be allocated when the segment is created. Max — Total number of extents, including the first, that can be allocated for the segment.
Percent	Incr — Percent by which each incremental extent grows over the last incremental extent allocated for a segment. Used — Minimum percent of data block containing data. Free — Minimum percent of data block maintained as unused space for data updates.
Trans	Initial — Preallocated amount of space for an initial number of transaction entries to access rows in the data block concurrently. Max — Number of transaction entries that can concurrently use data in a data block.
Code Tablespace	Name of the tablespace used when creating unaggregated codelist tables.

The following table lists the tablespace group descriptions:

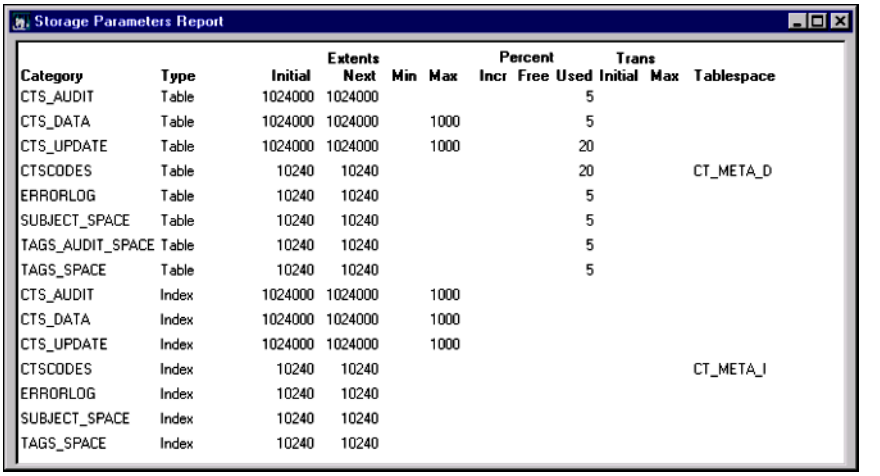
Tablespace Group:	Type:	Description:
CTS_AUDIT	Table	*panel-name*_AUDIT tables
	Index	Indexes on *panel-name*_AUDIT tables
CTS_DATA	Table	*panel-name*_DATA tables
	Index	Indexes on *panel-name*_DATA tables, including additional indexes on panels used by Thesaurus Views

Tablespace Group:	Type:	Description:
CTS_UPDATE	Table	*panel-name*_UPDATE tables
	Index	Indexes on *panel-name*_UPDATE tables, including additional indexes on Resolve Panels "VCT_ERRORSTATUS", "VCT_ERRORITEM"
CTSCODES	Table	Codelist table where aggregated = “Yes”
	Index	Indexes on codelist table where Aggregated="Yes"
ERRORLOG	Table	ERRORLOG table, CTL_DUPLICATE table (if protocol is a Lab Loader Transfer destination)
	Index	Indexes on ERRORLOG table, CTL_DUPLICATE table (if protocol is a Lab Loader Transfer destination)
SUBJECT_SPACE	Table	SUBJECT_BLOCK, SUBJECT_PAGE tables
	Index	Indexes on SUBJECT_BLOCK, SUBJECT_PAGE tables
TAGS_AUDIT_SPACE	Table	TAGS_AUDIT table
	Index	None
TAGS_SPACE	Table	TAGS table
	Index	Indexes on TAGS table

What is the Storage Parameters Report?

Admin provides a Storage Parameters Report containing default storage parameter information. The report shows the values you set in the Storage Parameters window.

The following figure shows the Storage Parameters Report in Admin:



Category	Type	Initial	Extents Next	Min	Max	Percent Incr	Free	Used	Trans Initial	Max	Tablespace
CTS_AUDIT	Table	1024000	1024000					5			
CTS_DATA	Table	1024000	1024000		1000			5			
CTS_UPDATE	Table	1024000	1024000		1000			20			
CTSCODES	Table	10240	10240					20			CT_META_D
ERRORLOG	Table	10240	10240					5			
SUBJECT_SPACE	Table	10240	10240					5			
TAGS_AUDIT_SPACE	Table	10240	10240					5			
TAGS_SPACE	Table	10240	10240					5			
CTS_AUDIT	Index	1024000	1024000		1000						
CTS_DATA	Index	1024000	1024000		1000						
CTS_UPDATE	Index	1024000	1024000		1000						
CTSCODES	Index	10240	10240								CT_META_I
ERRORLOG	Index	10240	10240								
SUBJECT_SPACE	Index	10240	10240								
TAGS_SPACE	Index	10240	10240								

For more information

For more information about space utilization and parameters, see Chapter 7.

6 Admin and Multisite

Overview

This chapter describes how Admin works differently in a Multisite environment, and contains the following sections:

- Users and usergroups
- Access rights and access levels

Users and usergroups

Users and usergroups are not replicated in a Multisite environment. You must define users and usergroups at each site, just as you would in a non-Multisite environment.

Access rights and access levels

This section describes the:

- Access rights and access levels for Multisite
- Access levels for read-only protocols in replication

Access rights and access levels for Multisite

The following table lists and describes the non-protocol access rights and access levels for Multisite:

Access Right:	Access Levels:
Distribute	None — The user or usergroup cannot access any distribution-related windows in Multisite. Read — The user or usergroup can access browsers and read-only windows related to codelist or protocol distribution, or the copying of base tables and functions. The user or usergroup cannot take any distribution-related actions on codelists or protocols, nor can the user or usergroup copy base panels or functions. Full — The user or usergroup can complete all distribution-related tasks for codelists and protocols, and copy base tables and functions.
Replicate	None — The user or usergroup cannot access any replication-related windows in Multisite. Read — The user or usergroup can access browsers and read-only windows related to protocol or account replication. The user or usergroup cannot take any replication-related actions on protocols or accounts. Full — The user or usergroup can complete all replication-related tasks for protocol replication and account replication.

Read-only protocols in replication

If a site in a replication environment for a protocol has been designated as read-only, then you can only assign the access level Read for certain protocol-specific access rights, as follows:

- Enter: Merged and Unmerged access rights
- Manage: Global, Other, Coding

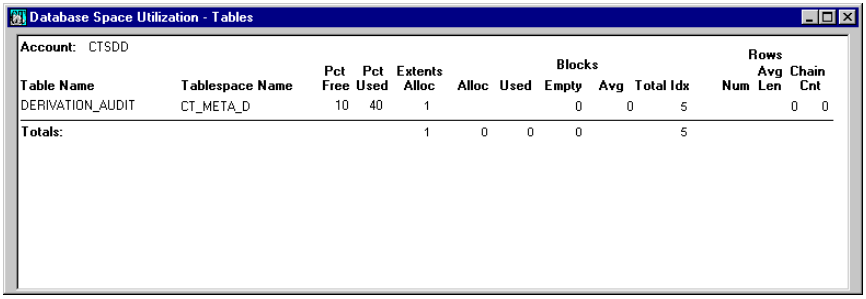
7 System Administration Tools

Monitoring space utilization and parameters

Admin provides two Database Space Utilization reports, one for tables and one for indexes.

Database Space Utilization — Tables

The following figure shows the Database Space Utilization — Tables report for the DERIVATION_AUDIT table in the CTSDD account:



You can select one or more tables for a single account for inclusion in the report.

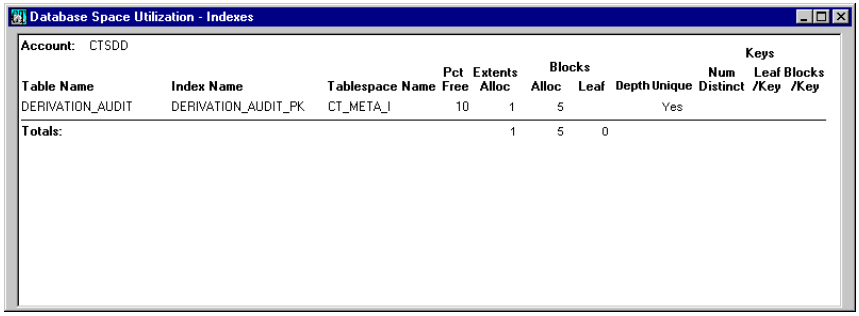
The Database Space Utilization — Tables report provides the following information:

Report column:	**Description:**
Table Name	Name of the table.
Tablespace Name	Name of the Oracle tablespace for the table.
Pct Free	Minimum percent of Oracle data blocks maintained as unused space for data updates.
Pct Used	Minimum percent of Oracle data blocks containing data.
Extents Alloc	Number of extents used by Oracle to store the table.

Report column:	Description:
Blocks	Alloc —Number of Oracle data blocks currently allocated. Used —Number of Oracle data blocks containing data. Empty — Number of Oracle data blocks empty. Avg — Average number of rows stored in each Oracle block. Total Idx — Number of Oracle data blocks allocated for the indexes associated with the table.
Rows	Num — Number of rows in the table. Avg Len — Average size (in bytes) of a row. Chain Cnt — Number of rows that are connected from one data block to another.

Database Space Utilization — Indexes

The following figure shows the Database Space Utilization — Indexes report for the DERIVATION_AUDIT index in the CTSDD account:



You can select one or more tables for a single account for inclusion in the report. All indexes for the selected tables are included.

The Database Space Utilization — Indexes report provides the following information:

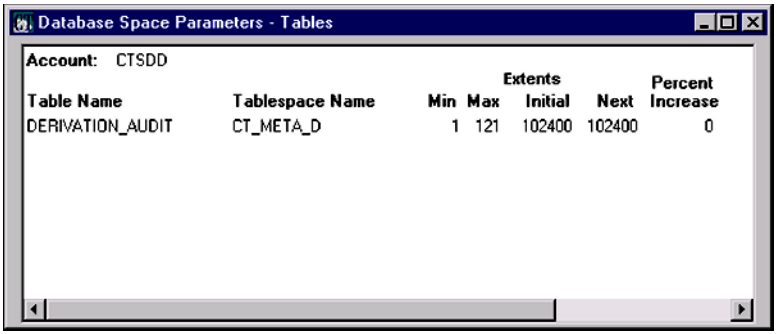
Report column:	Description:
Table Name	Name of the table selected.

Report column:	Description:
Index Name	Name of the index for the table.
Tablespace Name	Name of the Oracle tablespace for the table.
Pct Free	Minimum percent of data block maintained as unused space for index data updates.
Extents Alloc	Number of extents used by Oracle to store the index.
Blocks	Alloc ——Number of Oracle data blocks currently allocated. Leaf — Number of leaf blocks in the index. Depth — Depth of index from root block to leaf blocks.
Keys	Unique — Unique or Non-Unique. Num Distinct —Number of distinct indexed values. Leaf/Key — Average number of leaf blocks in which each distinct value in the index appears. Blocks/Key — Average number of Oracle data blocks that point to a distinct value in the index.

Database Space Parameters reports

Admin provides two Database Space Parameters reports, one for tables and one for indexes.

The following figure shows the Database Space Parameters — Tables report for the DERIVATION_AUDIT table in the CTSDD account:

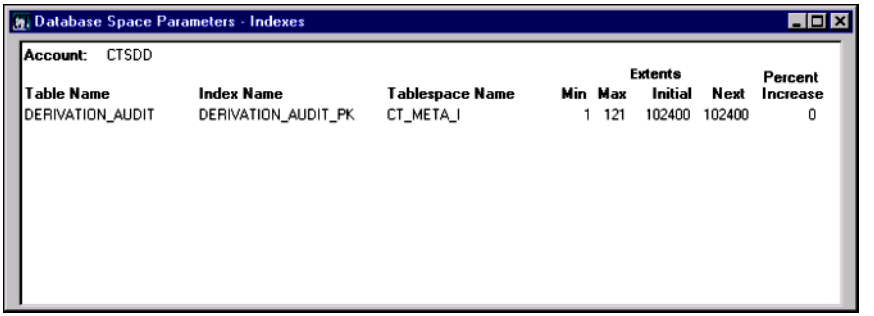


You can select one or more tables for a single account for inclusion in the report.

The Database Space Parameters — Tables report provides the following information:

Report column:	Description:
Table Name	Name of the table selected.
Tablespace Name	Name of the Oracle tablespace for the table.
Extents	Min — Total number of extents to be allocated when the segment is created. Max — Total number of extents, including the first, that can be allocated for the segment. Initial — Size, in blocks, of the first extent allocated when a segment is created. Next — Size, in bytes, of the next incremental extent to be allocated for a segment. Percent Increase — Percent by which each incremental extent grows over the last incremental extent allocated for a segment.

The following figure shows the Database Space Parameters — Indexes report for the DERIVATION_AUDIT index in the CTSDD account:



You can select one or more tables for a single account for inclusion in the report. All indexes for the selected tables are included.

The Database Space Parameters — Indexes report provides the following information:

Report column:	**Description:**
Table Name	Name of the table selected.
Index Name	The name of the index for the table.
Tablespace Name	Name of the Oracle tablespace.
Extents	Min — Total number of extents to be allocated when the segment is created. Max — Total number of extents, including the first, that can be allocated for the segment. Initial — Size, in blocks, of the first extent allocated when a segment is created. Next — Size, in bytes, of the next incremental extent to be allocated for a segment. Percent Increase — Percent by which each incremental extent grows over the last incremental extent allocated for a segment.

User Procedures reports

Admin provides reports about customized site-specific and protocol-specific functions for maintenance or archiving purposes. These reports display information about user-created functions that are stored in accounts specified by the PROC_SITE_ACCOUNT and PROC_ACCOUNT parameters. Two types of reports are available:

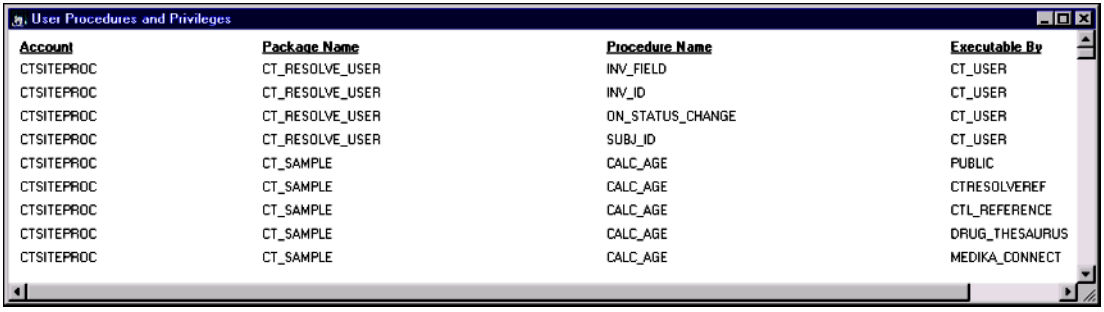
- User Procedures and Privileges
- User Procedures Usage

For more information on site-specific and protocol-specific functions, see the *Programming* section of the *Reference Guide*.

User Procedures and Privileges Report

The User Procedures and Privileges Report displays account and execute privilege information for user-created customized functions.

The following figure shows the User Procedures and Privileges Report in Admin:



The User Procedures and Privileges Report provides the following information:

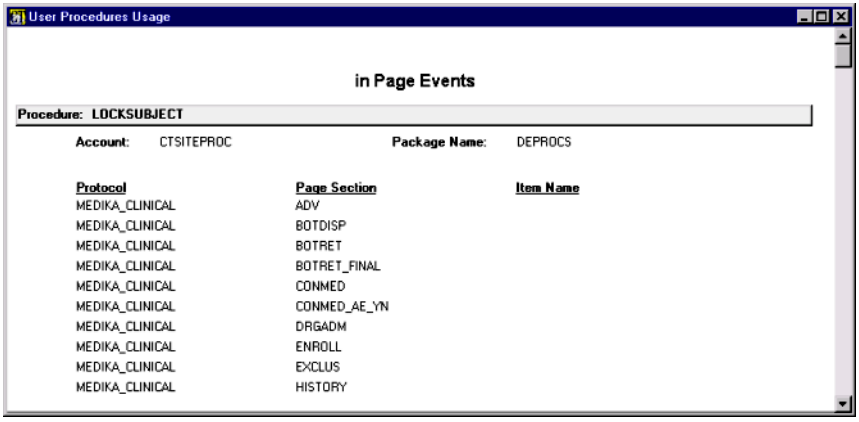
Report column:	Description:
Account	Name of the account that owns the procedure.
Package Name	Name of the package that contains the procedure (if it is a packaged procedure).
Procedure Name	Name of the procedure.

Report column:	Description:
Executable By	The EXECUTE privilege on the package or procedure. This privilege determines which protocols can call the package or procedure.

User Procedures Usage Report

The User Procedures Usage Report displays customized function identification information and lists the objects that use these functions. This report lists Procedures used in Page Events and in Value Changed Procedures.

The following figure shows the User Procedures Usage Report in Admin:



The User Procedures Usage Report provides the following information:

Report column:	Description:
Header Information	The top of each page indicates whether the procedures are used in Page Events or Value Changed Procedures. Procedure — Name of the procedure. Account — Name of the account that owns the procedure. Package Name — Name of the package that contains the procedure.

Report column:	Description:
Protocol	Name of the protocol containing the item or page template that uses the procedure.
Page Template (Page Events)	Name of the page template that uses the procedure.
Page Section (Value Changed Procedures)	Name of the page section containing the item that uses the procedure.
Item Name	Name of the item that uses the procedure.

Using SQL

You can use interactive SQL from within Admin. For details about using interactive SQL, see the *Reference Guide*.

Registering a site

At installation, you are prompted to name your local site (database instance). The local site name is the name by which your database instance is registered. You cannot change the installed name for the current database instance.

How to register a site

The Clintrial software registers sites automatically at installation and import. However, you can manually register additional sites if necessary by adding their names to the list of registered databases. Registering databases is useful when you perform protocol or codelist import from those databases.

Sites may also be registered using Multisite. Neither Multisite sites nor the local site can be deleted using Admin; however, the descriptions of these sites can be modified.

System Activity log

The *System Activity log* monitors menu commands that are used within any of the Clintrial software modules. If the System Activity log is running, menu selections are tracked in an Oracle table.

How to set the System Activity log

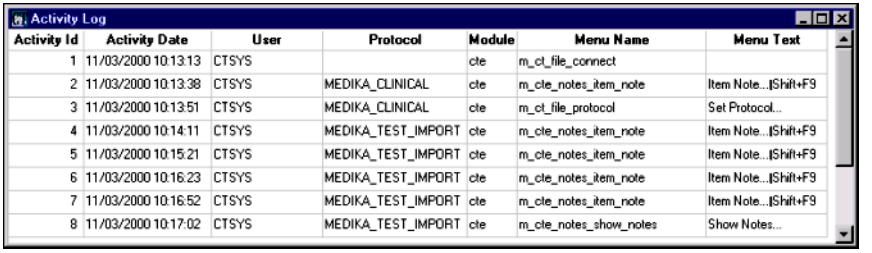
The System Activity log is activated by setting the following system parameters in Admin:

- ACTIV_LOG_START — The date at which you want system activity logging to activate
- ACTIV_LOG_END — The date at which you want system activity logging to deactivate

If you set ACTIV_LOG_START but not ACTIV_LOG_END, the System Activity log remains activated.

How to view or purge the System Activity log

Admin provides an Activity Log report, as shown in the following figure:


Activity Log

Activity Id	Activity Date	User	Protocol	Module	Menu Name	Menu Text
1	11/03/2000 10:13:13	CTSYS		cte	m_ct_file_connect	
2	11/03/2000 10:13:38	CTSYS	MEDIKA_CLINICAL	cte	m_cte_notes_item_note	Item Note...|Shift+F9
3	11/03/2000 10:13:51	CTSYS	MEDIKA_CLINICAL	cte	m_ct_file_protocol	Set Protocol...
4	11/03/2000 10:14:11	CTSYS	MEDIKA_TEST_IMPORT	cte	m_cte_notes_item_note	Item Note...|Shift+F9
5	11/03/2000 10:15:21	CTSYS	MEDIKA_TEST_IMPORT	cte	m_cte_notes_item_note	Item Note...|Shift+F9
6	11/03/2000 10:16:23	CTSYS	MEDIKA_TEST_IMPORT	cte	m_cte_notes_item_note	Item Note...|Shift+F9
7	11/03/2000 10:16:52	CTSYS	MEDIKA_TEST_IMPORT	cte	m_cte_notes_item_note	Item Note...|Shift+F9
8	11/03/2000 10:17:02	CTSYS	MEDIKA_TEST_IMPORT	cte	m_cte_notes_show_notes	Show Notes...

The Activity Log report includes the following information:

Report column:	**Description:**
Activity Id	Unique identifier for the activity being logged.
Activity Date	Date of insertion into the System Activity log.
User	User account name under which the activity occurred.

Report column:	Description:
Protocol	Protocol under which the activity occurred, if specific to a protocol.
Module	Module in which the activity occurred.
Menu Name	Internal name of the menu.
Menu Text	Text of menu command.

You can also purge all entries or selected entries from the System Activity log, using the Admin System Activity Log Purge function.

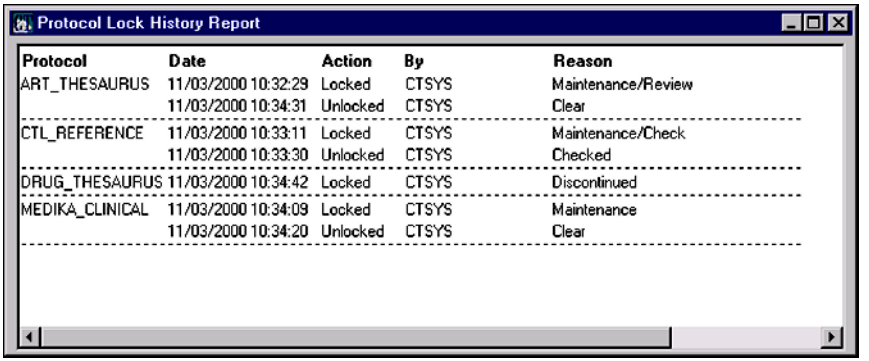
Protocol Lock History Report

The *Protocol Lock History Report* lists all instances of protocol locking and unlocking on the system. Locking a protocol ensures that the data and metadata in the protocol cannot be modified by allowing read-only access. Unlocking a protocol allows the data to be modified. These instances occur in Design typically when maintenance is being performed or when a study is complete.

The Protocol Lock History Report is created automatically when its window opens. When the report displays, you can:

- Filter the log to show only selected records.
- Print the report.
- Save the report data to a file.

The following figure shows a Protocol Lock History Report:



The Protocol Lock History Report provides the following information:

Report column:	**Description:**
Protocol	Name of the protocol on which the action was taken.
Date	Date of the action.
Action	Action taken (whether locked or unlocked).
By	User who performed the action.
Reason	Reason given for the action.

Section 2: Design

8 Introduction to Design

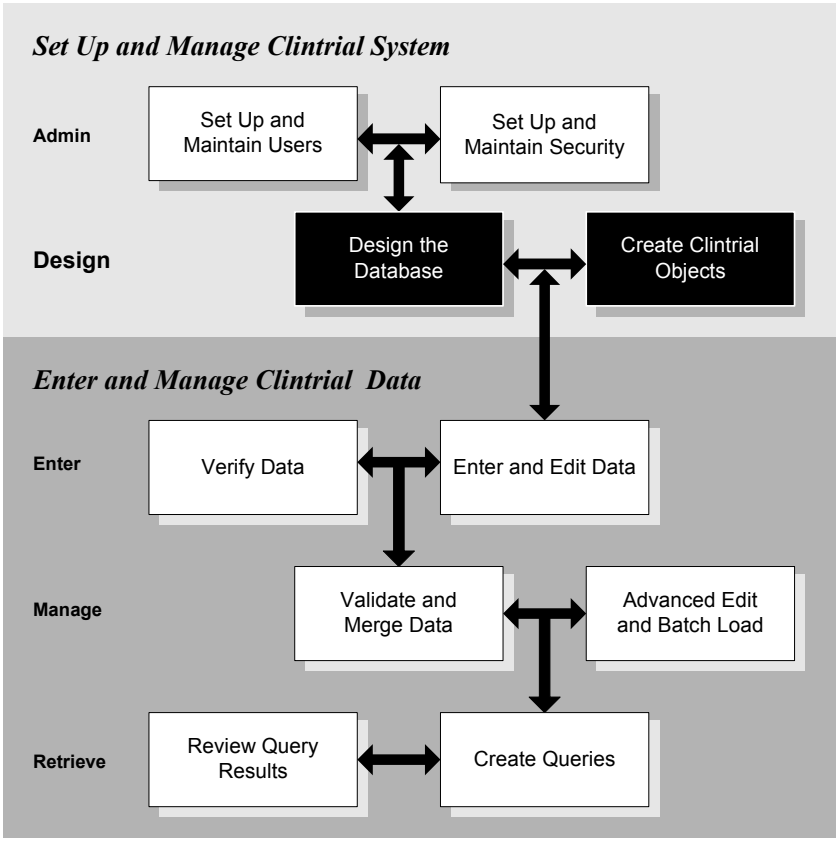
Overview

The Clintrial software Design module is the core module for designing a clinical trial database, a process that includes the following tasks:

- Defining how clinical data is entered and stored in the Oracle database
- Defining how clinical data is managed and retrieved from the Oracle database

How Design fits in the Clintrial software workflow

The following figure shows how Design fits in the Clintrial software workflow:



What are the Design tasks?

Using Design, you can:

- Prepare the Oracle database to store the clinical data collected in a clinical study.
- Create an online representation of a case report form (CRF) for data entry and editing.
- Define how data will be entered and managed; for example, by specifying which records require verification or when records are audited.
- Maintain metadata, such as standard items, panels, and study page sections, that can be shared by multiple studies.
- Define data, such as flags, notes, codelists, and coding thesauruses, that can be shared by multiple studies.

Who are Design users?

Designers and programmers are the main users of Design:

- Designers use Design primarily to define how the Clintrial software stores data, and to create data-entry screens.
- Programmers use Design to build rules, derivations, and data-entry processing procedures for checking or validating the logical consistency of clinical data, or for performing a particular action related to a study page, page section or item.

Clintrial software and Oracle database tables

How the Clintrial software uses database tables and accounts

The Clintrial software stores the Clintrial software objects you create in Design in Oracle database tables. Access to Oracle databases in the Clintrial software is through Oracle accounts. The Clintrial software handles the creation and use of most of these accounts transparently.

For a summary of Oracle database tables created through the Clintrial software, see the following section. For a complete description of the database tables, see the *Reference Guide.*

For more information on the use of Oracle accounts in the Clintrial software, see Chapter 3.

How database tables are created

The Clintrial software creates Oracle database tables as follows:

- At installation — The Clintrial software creates database tables to store data that defines Clintrial software objects, and other Clintrial software information, such as parameters and user accounts.
- At protocol creation — When you create protocols in Design, the Clintrial software also creates additional protocol-specific database tables related to error handling, flags, and notes.
- At panel installation — When you install panels in Design, you can instruct the Clintrial software optionally to create protocol-specific database tables to store clinical data.
- At codelist creation — When you create codelists, the Clintrial software may create additional database tables (for example, for unaggregated codelists).

For more information on protocols, see Chapter 9.

For more information on panels, see Chapter 10.

What is Clintrial software metadata and clinical data?

Clintrial software *metadata* is data that defines:

- Clintrial software objects, such as protocols and codelists
- Clintrial software system information, such as system parameters and user accounts

Clinical data is data that is collected during a clinical trial, for example, data about a subject collected on a CRF.

In Design, you work with the metadata that defines Clintrial software objects. You do not work directly with clinical data or with metadata that defines system information.

In Enter, Manage, and Retrieve, you use the Clintrial software objects that you create in Design to enter, manage, and retrieve clinical data. In Admin, you work with system administration.

For a summary of Clintrial software objects, see "Clintrial software objects" on page 127.

How is metadata stored?

The database tables that store the metadata are called *data dictionary tables*.

Metadata for Clintrial software objects is stored in the data dictionary tables as follows:

1. When you create a protocol, the Clintrial software stores the metadata (the protocol attributes) in a data dictionary table that contains protocol metadata for all protocols in the database instance.

   At protocol creation, additional protocol-specific data dictionary tables are created to store error handling information and flag or note information for the protocol.
2. As you create other objects within the protocol, the Clintrial software similarly stores the metadata for the objects in data dictionary tables that were created at installation for those objects.
3. When you install the panels for a protocol, the Clintrial software stores additional panel and item metadata in data dictionary tables that were created at installation for those objects when they are in an installed state.

The Clintrial software system information, such as user accounts or system parameters, are also considered metadata. At installation, data dictionary tables are created to store this information as it is created.

For information on installing panels, see Chapter 10.

For information on Clintrial software system information, see Chapter 7.

For a complete description of the data dictionary tables, see the *Reference Guide*.

How is clinical data stored?

The tables that store clinical data are called *clinical data tables*.

When you install panels, you can optionally choose to create clinical data tables. Clintrial software creates the following clinical data tables for each installed panel (except the context panel) in the protocol:

- An update table for clinical data that is still being worked on and, thus, is not ready to be moved to the data table

- A data table for clinical data that has passed validation; that is, data that is clean
- An audit table for copies of clinical data as it was before modification or deletion

For information on the context panel, see Chapter 10.

For information on how to plan for auditing when creating a protocol, see Chapter 9.

For more information on validation and auditing, see *Manage, Classify, and Lab Loader.*

How are tablespaces used?

When you create a protocol, you can choose to accept the default tablespaces for the tables that store clinical data and indexes. Creation of additional tablespaces can be useful for the distribution of tables and indexes.

When you install panels and choose to create clinical data tables, the Clintrial software creates clinical data tables in the tablespace you designated when you created the protocol. The clinical data tables store the clinical data as it is entered in Enter.

For more information on specifying tablespaces, see Chapter 9.

Clintrial software objects

What are Clintrial software objects?

Clintrial software *objects* are the data structures that serve as the building blocks of the online representation of your clinical studies.

Categories of objects

You can loosely group Clintrial software objects into three categories, as shown in the following table:

Category:	Description:	Clintrial software objects:
Data storage objects	Organize and store clinical data.	Protocol, panel, item.
Data management objects	Assist in the entry, processing, and management of clinical data.	Flags, notes, codelists, rules, derivations and coding thesauruses.
Display objects	Define the graphical layout and organization of data-entry windows.	Page section, page template, study book, study page.

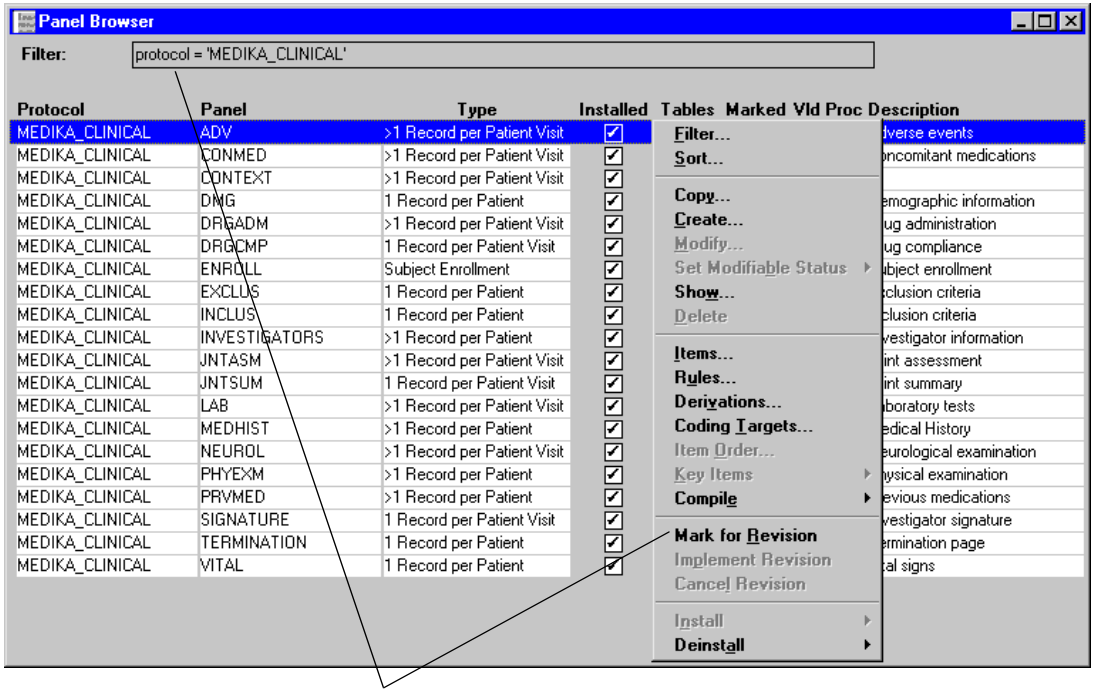
How to create an object

To create an object, you specify attributes for the object. *Attributes* are characteristics of an object that define how the object works with the data in your study.

For example, the attributes of a Clintrial software item include:

- Name
- Database format
- An indication of whether the item is required

In Design, you use object browsers to work with objects in the current protocol. The following figure shows a Panel Browser:



You can select an object, then click the right mouse button to view a list of tasks that you can perform for the object.

Data storage objects

The following Clintrial software objects are used for data storage:

- Protocols — Are parent containers for organizing all other Clintrial software objects.
- Panels — Contain logically or clinically related items.
- Items — Represent data collected by individual fields on a study page.

What is a protocol?

A *protocol* is one of the following:

- A *clinical protocol* describes the data needed for a particular clinical study and how that data is to be collected.
- A Clintrial software *protocol* is a logical container that organizes the metadata that defines the Clintrial software objects and the clinical data for a clinical study.
- A *protocol account* is an Oracle account that consists of tables that store the clinical data, flags, and notes associated with clinical data; audit and Error Log information; and one or more views of the clinical data stored in the protocol.

What is a panel?

A *panel* is a collection of logically related or clinically related items. For example, the Panel Browser shown in "How to use browsers" on page 129 lists the panels created for the MEDIKA_CLINICAL protocol.

What is an item?

An *item* is a Clintrial software object representing the data collected by a single field in a study page. Items are grouped into panels. For example, the PRVMED panel in the MEDIKA_CLINICAL protocol contains the items listed in the following Item Browser:

Item Browser

Filter: protocol = 'MEDIKA_CLINICAL' AND panel = 'PRVMED'

Protocol	Panel	Item	Rev State	DB Format	Description
MEDIKA_CLINICAL	PRVMED	ALGORITHM	IN	VARCHAR2(20)	Name of algorithm
MEDIKA_CLINICAL	PRVMED	CONTIN	IN	NUMBER(1)	Medication continuing?
MEDIKA_CLINICAL	PRVMED	DOSEUNIT	IN	VARCHAR2(20)	Dose unit
MEDIKA_CLINICAL	PRVMED	DRG_NUMMAT	IN	VARCHAR2(4)	Number of matches found
MEDIKA_CLINICAL	PRVMED	DRGCOD_DATE	IN	DATE	Date of coding
MEDIKA_CLINICAL	PRVMED	DRGCODE	IN	VARCHAR2(10)	Code 1 item
MEDIKA_CLINICAL	PRVMED	DRGCODER	IN	VARCHAR2(20)	User account that performe
MEDIKA_CLINICAL	PRVMED	DRGCONF	IN	VARCHAR2(2)	Algorithm step
MEDIKA_CLINICAL	PRVMED	DRGNORM	IN	VARCHAR2(25)	Normalized
MEDIKA_CLINICAL	PRVMED	DRGWKFLOW	IN	VARCHAR2(5)	Workflow
MEDIKA_CLINICAL	PRVMED	INDICATION	IN	VARCHAR2(20)	Reason for medication
MEDIKA_CLINICAL	PRVMED	MAXDOSE	IN	VARCHAR2(20)	Maximum daily dose
MEDIKA_CLINICAL	PRVMED	MEDNAME	IN	VARCHAR2(30)	Name of concomitant medic
MEDIKA_CLINICAL	PRVMED	MEDSTART	IN	DATE	Medication start date - deriv
MEDIKA_CLINICAL	PRVMED	MEDSTOP	IN	DATE	Medication stop date - deriv
MEDIKA_CLINICAL	PRVMED	STARTDD	IN	VARCHAR2(2)	Start day - DD
MEDIKA_CLINICAL	PRVMED	STARTMMM	IN	VARCHAR2(3)	Start month - MMM
MEDIKA_CLINICAL	PRVMED	STARTYYYY	IN	VARCHAR2(4)	Start year YYYY
MEDIKA_CLINICAL	PRVMED	STOPDD	IN	VARCHAR2(2)	Stop date - DD
MEDIKA_CLINICAL	PRVMED	STOPMMM	IN	VARCHAR2(3)	Stop month - MMM
MEDIKA_CLINICAL	PRVMED	STOPYYYY	IN	VARCHAR2(4)	Stop year - YYYY

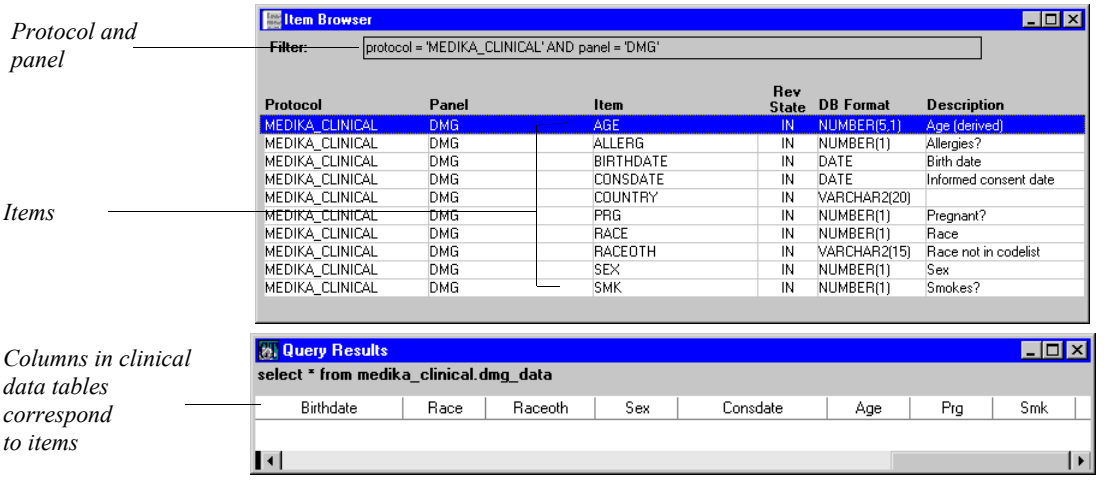
Order for creating data storage objects

You create protocols, panels, and items in the following order:

1. Create a protocol.
2. Create panels within the protocol.
3. Create items within each panel.

How panels relate to clinical data tables

Panels define the Oracle database tables that store the clinical data. The following figure shows the relationship of the panel and its items to the underlying clinical data tables:



When are clinical data tables created?

After you create a panel, you can install it. When a panel is installed, you can optionally create the three clinical data tables: the update table, data table, and audit table.

For more information

For more information on protocols, see Chapter 9.

For more information on panels and items, see Chapter 10.

For more information on creating and installing panels and items, see the Design Help.

Objects for managing clinical data

The following Clintrial software objects are used for managing clinical data:

- Codelists — Sets of code values linked to an item
- Derivations and rules — PL/SQL statements used to validate data in Manage
- Notes and flags — Information about clinical data that is attached to items, records, or observations during data entry or validation
- Coding thesauruses — Dictionaries containing standard codes for a particular type of clinical data

What is a codelist?

A *codelist* links a set of codes to a corresponding set of values. A codelist is associated with an item. For example, a codelist for sex might include the codes 1 and 2, associated with the values M and F.

The codelists you create are not study-specific; you can use the same codelists with multiple studies throughout the Clintrial software database instance.

What is a derivation?

A *derivation* is a PL/SQL statement that calculates the value of an item or a temporary variable when the record is validated in Manage. A derivation is attached to a panel.

What is a rule?

A *rule* is part of a PL/SQL statement that evaluates to TRUE or FALSE when the record is validated. A rule is associated with a panel and can refer to any item in the panel or to any temporary variable calculated by a derivation attached to a panel. Rules check that your clinical data meets the requirements of the clinical protocol.

What is a flag?

A *flag* is a statement about clinical data, consisting of the following parts:

- *Flag category* — Characterizes the clinical data. Data-entry operators can attach a flag category to an observation, record, or specific item.
- *Flag name* — Describes the action taken about the clinical data. Data-entry operators can attach a flag name within a flag category to an observation, record, or specific item.

What is a note?

A *note* is an annotation about clinical data, consisting of the following parts:

- *Note category* — Characterizes the person who annotates the clinical data
- *Note name* — Further characterizes the person who is the source of the note

For more information on creating flags and notes, see Chapter 13.

For information on using flags and notes in Enter, see *Enter, Resolve, and Retrieve*.

What is a coding thesaurus?

A *coding thesaurus* is a dictionary that contains standard codes for a particular type of clinical data, such as WHOART or COSTART. The Clintrial software supports coding in multiple languages.

In order to match clinical data (verbatim text) against standard dictionary codes, you may either:

- Use Clintrial Manage, and optionally, the Clintrial Classify extended module.
- Use the Oracle Central Coding application.

The decision to use either application may be made on a protocol basis or you may standardize for your organization. For example, existing protocols may continue to use the Classify module for coding, while new protocols may use Central Coding. Of course, existing protocols may be updated to use Central Coding if you want to use it as a standard within your organization.

For more information about using coding thesauruses in Clintrial software, see Chapter 14.

Data Display Objects

After you have created the protocols, panels, and items that define how to store and manage the data you are collecting, you can create objects for entering, editing, and displaying data.

In Enter, you enter and edit clinical data in a data-entry window that looks similar to a page in the CRF for your study. The following sections describe the Clintrial software objects that comprise the online representation of the CRF.

Types of data-entry and display objects

The types of objects you can create for entering, editing, and displaying clinical data are:

- Page sections
- Page templates
- Study books
- Blocks
- Study pages

What is a page section and a context section?

A *page section* is part of a study page that corresponds to sections of a CRF page. A *context section* is a special type of page section that contains the unique identifying items for the record or records that you enter in the page.

What is a page section layout and a field?

A *page section layout* defines how a page section looks and how data is entered in fields. A *field* is the data-entry area in which values for an item can be entered in a page section on a study page.

What is a page template?

A *page template* defines the order of page sections in a study page. A page template can consist of one or more page sections.

What is a study book?

A *study book* is an ordered list of related study pages that corresponds to a CRF. Study books are used to access and enter subject data, providing the data-entry interface to the underlying Oracle database tables.

What is a block?

A *block* is a grouping of related pages in a study book. Blocks usually represent subject evaluation checkpoints, such as subject visits.

What is a study page?

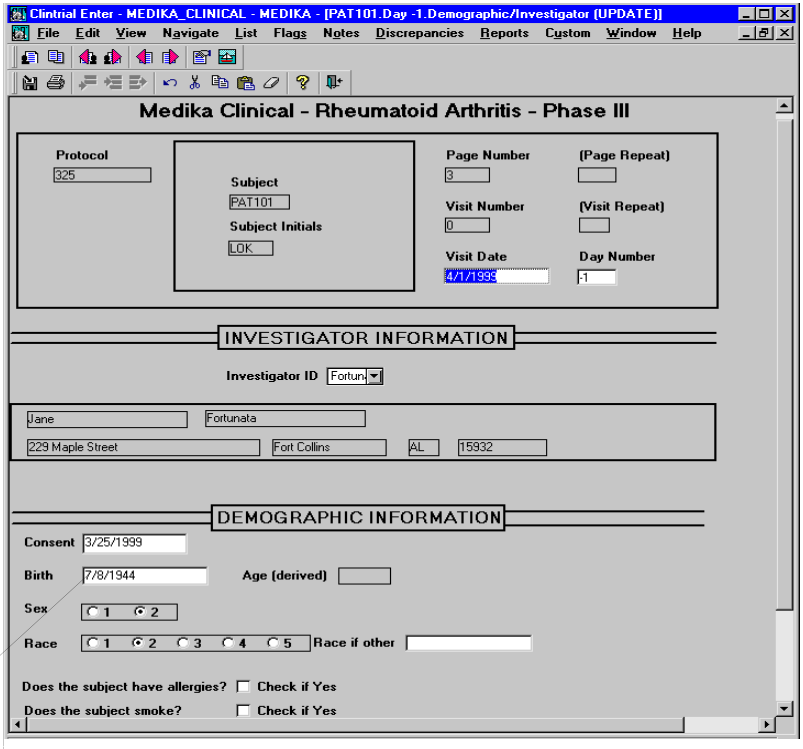
A *study page* is an online representation of a CRF page. Study pages are based on page templates, but they also contain information that defines the position of the study page within a block and a study book.

How data-entry objects are related

The following example shows the elements of a study page:

The context page section contains identifying items for the record.

This page contains three page sections.



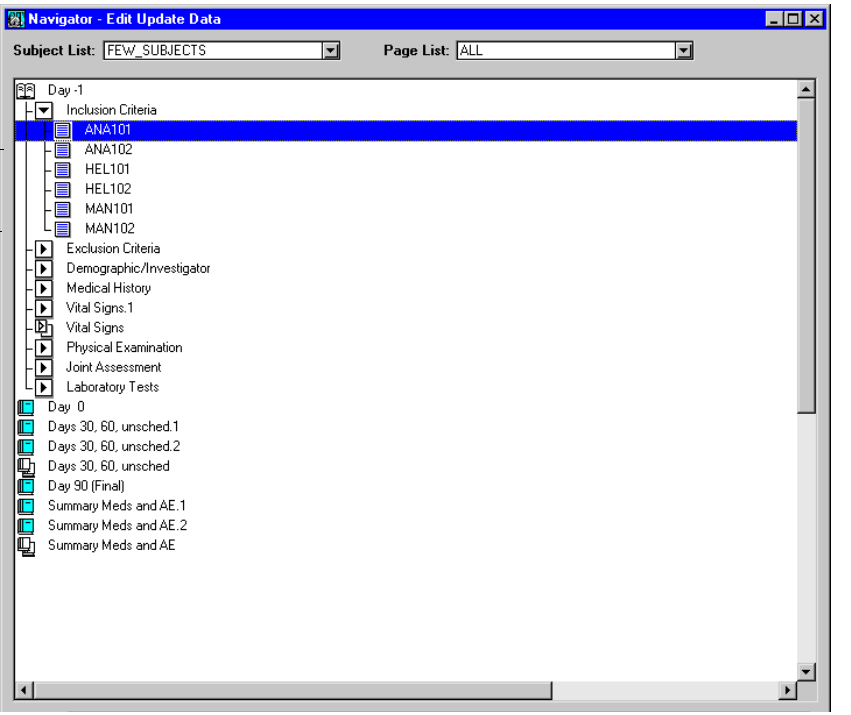
This is a field in a study page section, corresponding to one item in the clinical data tables.

The following example shows how the MEDIKA study book is ordered. This hierarchy of block, study page, and subject is displayed in the Navigator in Enter:

The blocks in this study book correspond to subject visits.

The subject ID appears in the Navigator.

The study pages correspond to pages in the CRF book.



Order for creating

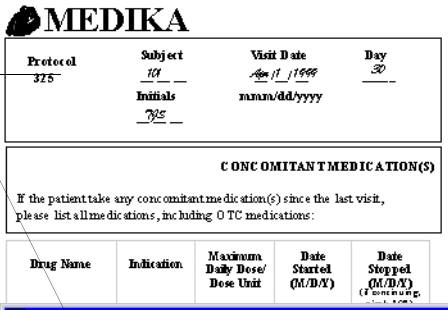
After creating and installing panels, you create the layout and study book objects in the following order:

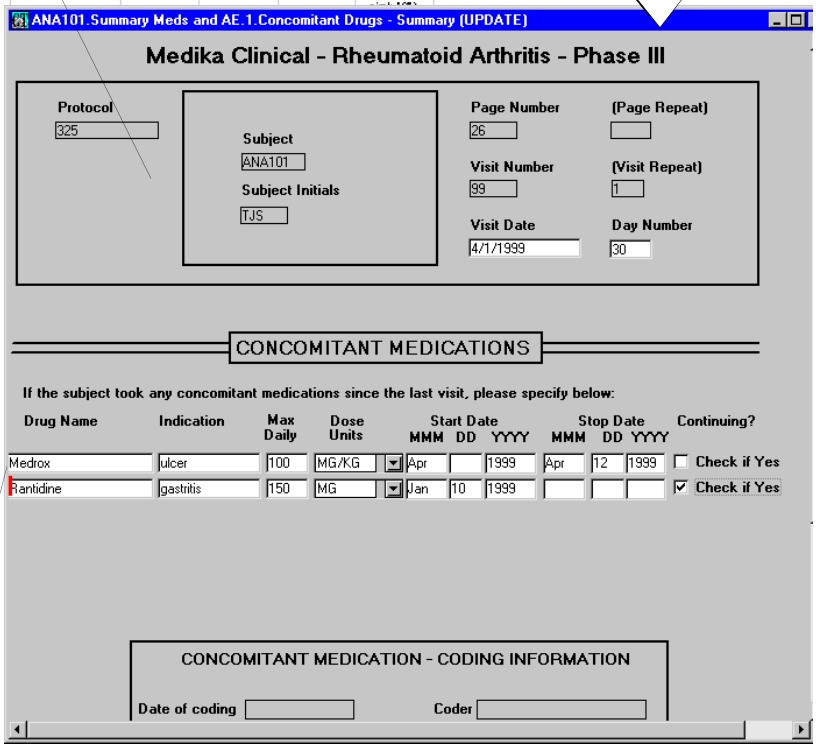
1. Create the objects that define the page layouts:
   - Create page sections.
   - Using a graphical layout editor, define the page section layout.
   - Create page templates.
2. Create the objects that build a study book:
   - Create the study book.
   - Create blocks.
   - Using your predefined page templates, create study pages.

Sample study page

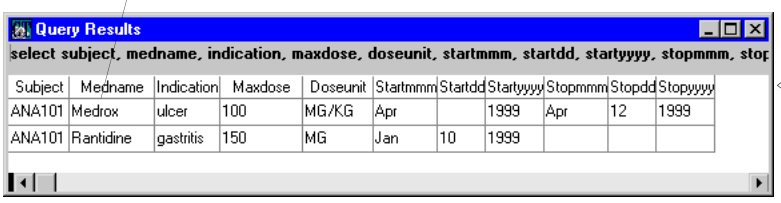
The following example shows a study page from the MEDIKA study book and its relationship to the CRF and clinical data tables:

Sections in the CRF correspond to page sections in the study page.





Data you enter in the study page is stored in clinical data tables.



Standardizing Clintrial software data

What is standardization?

Standardization is a process that enforces consistency among:

- Data definitions across multiple projects
- Clinical data from multiple sources

In Design, you can standardize metadata by creating common Clintrial software objects that you can use with multiple studies. With standardization, you can efficiently share data definitions and enforce consistency among studies. For example, you can define the following objects to use in multiple studies:

- A generic panel DEMOG, that contains information about a subject
- A generic codelist RACE, that assigns a one-digit code to race

You can also standardize clinical data by translating terms into codes from standard coding thesauruses.

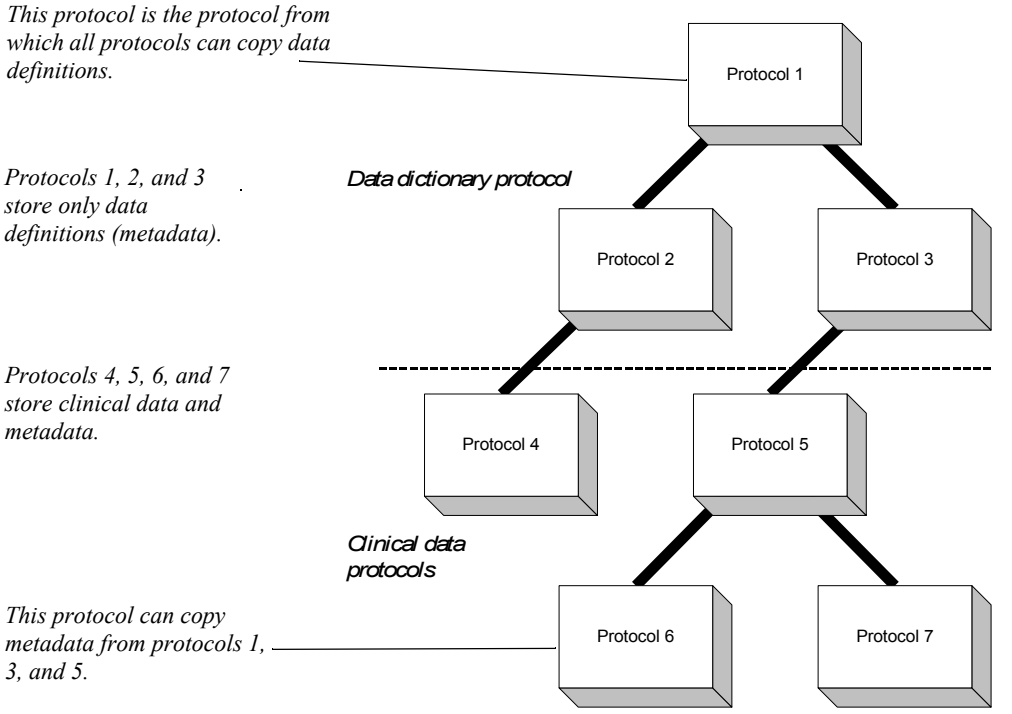
Tasks to help standardization

The following tasks promote the standardization of clinical data and Clintrial software metadata:

- Defining protocol hierarchies
- Using PL/SQL stored procedures and functions
- Using coding thesauruses
- Using revision control
- Distribution of codelists and protocols

Why use protocol hierarchies?

A *protocol hierarchy* defines a relationship among a set of protocols. Protocol hierarchy encourages the copying of objects from standardized protocols, as shown in the following figure:



For more information on protocol hierarchies, including the protocol attributes that set hierarchy, see Chapter 9.

Why use PL/SQL procedures?

PL/SQL is the Oracle procedural extension language to SQL. You can use PL/SQL with Clintrial software as follows:

- When you create derivations and rules for a panel, you can enter PL/SQL directly, or you can call any valid PL/SQL function, procedure, or package that you have stored elsewhere.
- A data-entry processing procedure is a PL/SQL procedure that runs when the user performs a specific action on a study page, page section, or item. When you attach a data-entry processing procedure to an object, you specify the name of the procedure and store the data-entry processing procedure elsewhere in valid PL/SQL.

By using stored procedures and functions, you make standardized PL/SQL available to multiple Clintrial software protocols, and to multiple derivations, rules, and data-entry processing procedures within a protocol.

For more information on using PL/SQL with the Clintrial software, see the *Reference Guide*.

Why use a coding thesaurus?

A *coding thesaurus* is a thesaurus that contains standard codes for a particular type of clinical data. In Design, you can:

- Load data from an industry-standard coding thesaurus; for example:
  - Coding Symbols for a Thesaurus of Adverse Reaction Terminology (COSTART)
  - World Health Organization Adverse Reaction Terminology dictionary (WHOART)
- Refine the thesaurus or design your own thesaurus.

You then associate the coding thesaurus protocol with specific items that you want to code. Users of the Clintrial application may decide to use either the Clintrial software extended module Classify or the Oracle application Central Coding to do this.

For more information on using coding thesauruses in the Clintrial software, see Chapter 14.

For more information on using Oracle's Central Coding application with the Clintrial software, see Chapter 15 of this document.

Why use revision control?

Revision control enables you to manage and track changes to your metadata. In Design, you can use revision control to track changes to objects. When you copy an object you can specify that the copied object (the destination object) is connected to the object from which it is copied (the source object). Then, you can view changes to the metadata, see where objects with a connection were copied from or to, and propagate changes of the source object to the destination object.

For more information on revision control, see Chapter 19.

Why export and import codelists and protocols?

The Clintrial software enables you to transfer protocols within and between Clintrial software database instances and to transfer codelists between database instances. Using Design you can:

- Export and import codelists.
- Export and import protocols containing only metadata.
- Export and import protocols containing both metadata and clinical data.

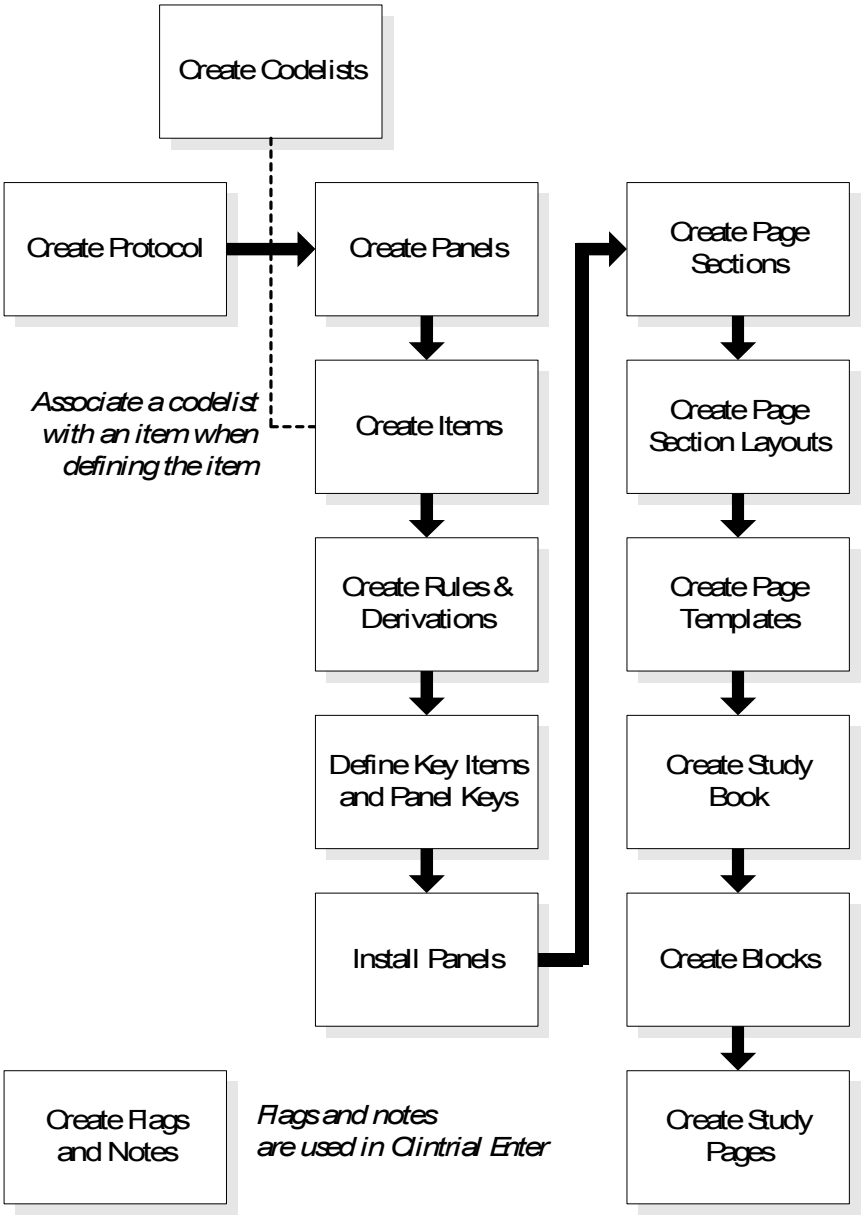
Exporting and importing protocols helps you to:

- Facilitate work with contract research organizations (CROs).
- Archive studies.

For more information on exporting and importing protocols and codelists, see Chapter 20.

Sample workflow in Design

The following figure shows a typical workflow within Design:



Required access rights and access levels

What is a non-protocol access right?

A *non-protocol access right* pertains to Clintrial software activities that are not associated with a particular protocol, but only with a usergroup or a user.

What is a protocol access right?

A *protocol access right* pertains to Clintrial software activities that require access to specific protocols. A protocol access right must be associated with a protocol as well as with a user or usergroup.

List of Design non-protocol access rights

The following table lists the Design non-protocol access rights and access levels:

Access right:	User can:	Access levels:
Global	Work on codelists, flags, and notes.	Full — Can perform all valid actions on these objects, including Creating, Modifying and Deleting objects. Read — Can view these objects, but not edit them. None — Cannot access these objects.

Access right:	User can:	Access levels:
System	Work on protocols and thesaurus objects.	Full — Can perform all valid actions on the protocol objects, including Locking, Unlocking, Creating, Modifying and Deleting protocol objects. Read — Can view these objects, but not edit them. None — Cannot access these objects.

List of Design protocol access rights

The following table lists the Design protocol access rights and access levels:

Access right:	User can:	Access levels:
Database	Work on panels, items, rules, derivations, and coding targets.	Full — Can perform all valid actions on these objects. Read — Can view object attributes, but not work on these objects. None — Cannot view these objects.
Data Entry	Work on page sections, page templates, study pages, and study books.	Full — Can perform all valid actions on these objects. Read — Can view object attributes, but not work on these objects. None — Cannot view these objects.

Setting your protocol parameters and user preferences

What are protocol parameters?

Protocol parameters are parameters that tailor the working environment for particular protocols. Protocol parameters take their default values from corresponding system parameters. The study designer sets protocol parameters.

For more information, see Chapter 9 and the Design Help.

What are user preferences?

User preferences are parameters that tailor the working environment for individual users. User preferences take their default values from corresponding system parameters. Not all system parameters have corresponding user preferences.

The following table describes the user preferences that are available in Design:

User Preference:	Default Value:	Description:
DD_DBFMT_FIXED	NUMBER(5)	Default database format for items of the data type FIXED.
DD_DBFMT_FLOAT	NUMBER(6,2)	Default database format for items of the data type FLOAT.
DD_DBFMT_TEXT	VARCHAR2(20)	Default database format for items of the data type TEXT.
DD_SASNAME_UPDATE	No	If Yes, the default SAS name associated with an item is automatically updated when the item name is modified. If No, the SAS name is not automatically updated.

User Preference:	Default Value:	Description:
FRM_CODEENTRY	No	If Yes, Enter as Code is checked by default in the Define Item Style window for a field to which a codelist is attached. If No, Enter as Code is not checked by default.
FRM_OVERRIDE	No	If Yes, Override is checked by default in the Modify Object Attributes window for a field to which a checklist is attached. If No, Override is not checked by default.
FRM_USE_DESCRIP	No	If Yes, the item description displays as the default text label for an item in the page section layout. If No, the item name displays as the default text label.

Part I: Objects for Storing Clinical Data

9 Protocols

Overview

Part I describes the Clintrial software objects that you create in Design to define how the Clintrial software clinical data is stored. The objects that define clinical data storage are:

- Protocols

  For more information, see this chapter.

- Panels and items

  For more information, see Chapter 10.

Introduction to protocols

What is a protocol?

The Clintrial software protocol is central to the representation of your CRFs and the data collected in them. Keep in mind that a Clintrial software protocol differs from a clinical protocol:

- A *clinical protocol* describes the data needed for a particular clinical study and how that data is to be collected.
- A Clintrial software *protocol* is a logical container that organizes the metadata that defines the Clintrial software objects and the clinical data for a clinical study.
- A *protocol account* is an Oracle account that consists of tables that store the clinical data, flags, and notes associated with clinical data; audit and error log information; and one or more views of the clinical data stored in the protocol.

How do protocols use Oracle tablespaces?

When you create a protocol, you can choose to accept the default tablespaces for the database tables that store clinical data and indexes. Creation of additional tablespaces can be useful for the distribution of database tables and indexes.

When you install panels and choose to create clinical data tables, the Clintrial software creates clinical data tables in the tablespace you designated when you created the protocol. The clinical data tables store the clinical data as it is entered in Enter.

For more information on how the Clintrial software uses Oracle tablespaces, see "Tablespaces for specific data space usage" on page 176.

What is protocol design?

Protocol design consists primarily of planning how to:

- Create and organize a database.

  The Oracle database tables are created when the Clintrial software is installed, when you create a protocol, and when you install panels within a protocol.

- Access and manage the database.

  When you create a protocol, you specify attributes to control that protocol's access to the database tables that store definitions for the Clintrial software objects. Protocols also specify the default auditing behavior for all panels you create within the protocol. You also can create a protocol that limits the end user's view of the data.

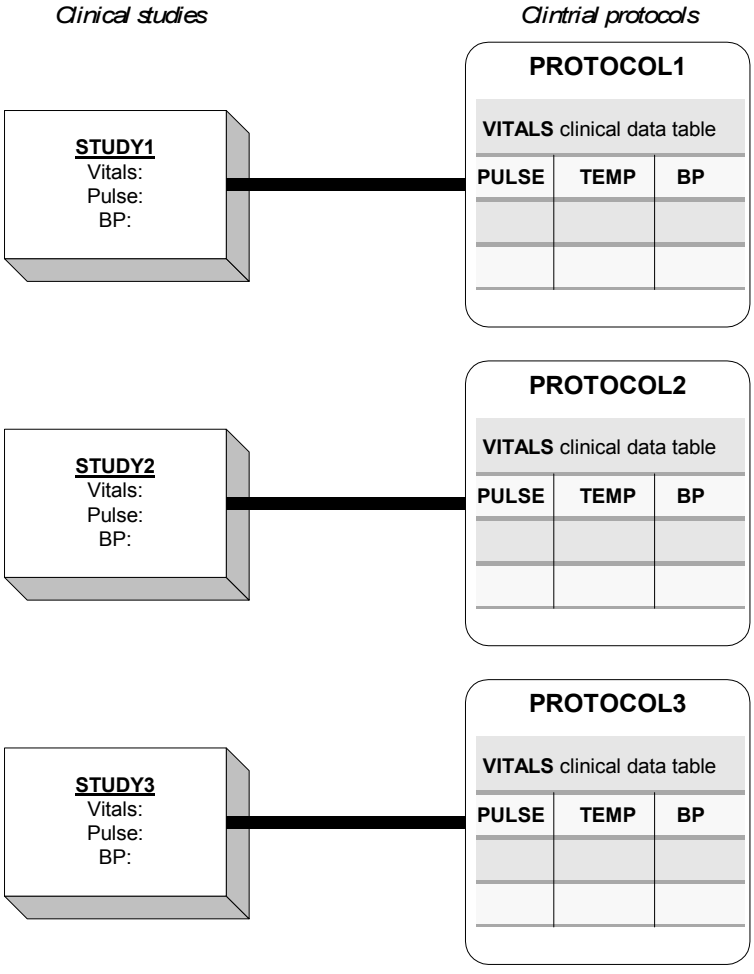
Understanding clinical studies and the Clintrial software protocols

A Clintrial software protocol determines how to store and access the metadata and clinical data for one or more clinical studies in Oracle database tables. This section describes the relationship between clinical studies and Clintrial software protocols.

How does a protocol relate to a clinical study?

Typically, you create a Clintrial software protocol for each clinical study, so that one protocol controls the clinical data tables for one clinical study.

The following figure shows the typical relationship between clinical studies and protocols:



In this figure, clinical studies STUDY1, STUDY2, and STUDY3 each are represented in a separate protocol. The Clintrial software creates clinical data tables for each installed panel in the protocol. All the data from a similar section of the CRF page (Vitals) is stored in the three separate but similar sets of clinical data tables (the VITALS tables for each protocol).

For more information on panels, see Chapter 10.

Can multiple clinical studies use one protocol?

Although the typical relationship between clinical studies and protocols is one-to-one, you can also create a protocol that contains information for more than one clinical study. If you have multiple clinical studies that collect very similar data, you may find it useful to group the clinical studies into one Clintrial software protocol.

For example, suppose you create a single protocol for the three clinical studies shown in the previous figure (STUDY1, STUDY2, and STUDY3). You could then create one panel to define a single set of clinical data tables for all sections in the CRF pages for the three clinical studies that contained similar information, such as Vitals.

Considerations for one protocol to one clinical study

The possible advantages of creating one protocol for each clinical study are:

- Ability to limit access to clinical data on a study-by-study basis.
- Ease of access and retrieval of data on a study-by-study basis.
- Potentially smaller size of tables than when clinical studies are grouped in one protocol, so that processing is easier.
- Ability to minimize the amount of transferred metadata and data when transferring a protocol specific to a single clinical study.

The possible disadvantages are:

- Pooling data for cross-protocol studies may require more work if the items are not standardized across protocols.
- More protocols require more tables, which are more overhead.
- There are more Clintrial software objects to create because the number is related to the number of protocols.

For more information

For planning considerations when you are ready to create a protocol within the Clintrial software, see "Protocol attributes" on page 173.

Planning protocol hierarchy

What is protocol hierarchy?

A *protocol hierarchy* is a description of the relationships among protocols. Protocol hierarchy is a standardization tool that you can use to control the creation of Clintrial software objects.

Protocol hierarchy facilitates protocol design by enabling you to set up standardized metadata for use in multiple clinical data protocols. As you decide whether to set up a protocol hierarchy and plan how to structure it, you must consider the following:

- Protocol type — There are five protocol types depending on the function for which the protocol is created.
- Dictionary attribute — Used to specify certain protocol types as data dictionaries.

For more information on protocol type, see "What is a data dictionary protocol?" on this page.

For more information on data dictionary protocols, see below.

Hierarchy and types of protocols

Protocol hierarchy is used primarily to differentiate between data dictionary protocols and clinical data protocols without the Dictionary attribute checked.

What is a data dictionary protocol?

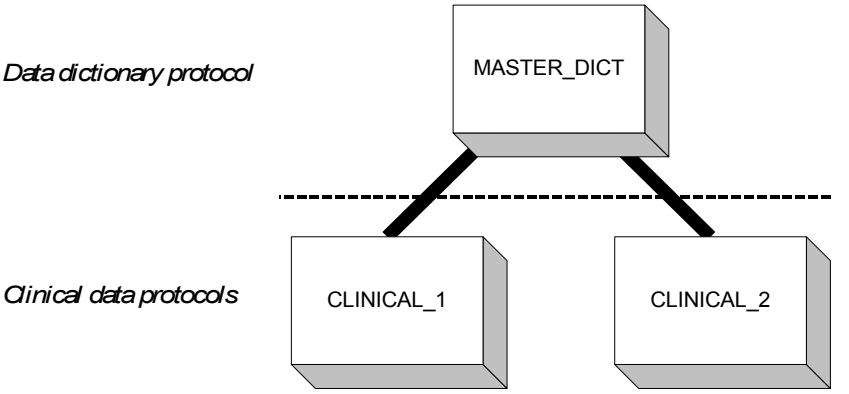
You can define certain types of protocols as *data dictionary protocols* by setting the Dictionary attribute when you create them. A data dictionary protocol contains only metadata definitions of Clintrial software objects created within the protocol. A data dictionary protocol cannot create clinical data tables and cannot store clinical data. For example, when you set the Data Dictionary attribute even for a Type 1 (Clinical Data) protocol, it is then not used to store actual clinical data.

For more information on using the Dictionary attribute to specify a protocol as a data dictionary, see page 174.

What is a clinical data protocol?

A *clinical data protocol* also contains the metadata definitions of Clintrial software objects created within the protocol. Optionally, if it does not have the Dictionary attribute set, a clinical data protocol also includes database tables that store clinical data.

The following figure shows a simple protocol hierarchy:



Typically, data dictionary protocols serve as master dictionaries, storing standardized metadata for objects defined in the protocol. The designer of a clinical data protocol can create a protocol and copy object definitions from the standardized metadata in data dictionary protocols. Whether or not you create a protocol hierarchy, you can also allow designers of objects in clinical data protocols to define their own objects without reference to the standardized metadata.

Why create protocol hierarchy?

Protocol hierarchy allows for:

- Standardization of metadata

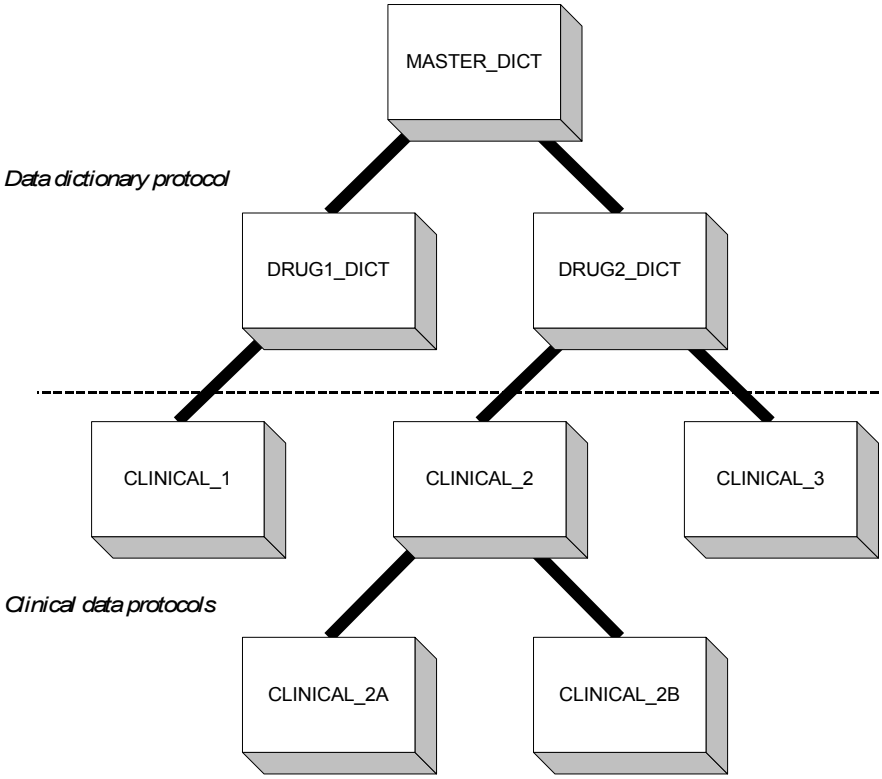
  By creating protocols that contain only metadata and protocols that also contain clinical data, you can separate those Clintrial software activities associated with management of standardized metadata from those associated with data collection, validation, and retrieval.

- Separation of related groups of protocols from unrelated groups

  By creating different hierarchies for protocols, you allow protocols to copy objects from shared common protocols (typically data dictionary protocols). When you create branching hierarchies below shared common protocols, you

also ensure that protocols only have access to and knowledge of those protocols from which they need to copy objects.

Such a protocol hierarchy can have multiple levels, as the following figure shows:



This example assumes that:

- The Modify Searchlist protocol attribute is cleared, so that the searchlists are fixed.
- The CREATE_OBJ_DD protocol parameter is set to No, so that items can be created only by copying other items (or panels, which contain items).

This figure shows how:

- Protocol hierarchy separates standardized dictionary protocol metadata item definitions from clinical data item definitions.

  Separate dictionary protocols for each drug project store definitions of those items that are specific only to clinical protocols concerned with those drugs. For example, you can copy items from DRUG1_DICT when you need to create them in CLINICAL_1. You can copy from DRUG2_DICT to both

CLINICAL_2 and CLINICAL_3. MASTER_DICT is available to all three clinical protocols.

- Protocols can share common protocols for copying objects and yet remain insulated from each other.

  For example, items in CLINICAL_2 are available to CLINICAL_2A and CLINICAL_2B, but not to other clinical data protocols. Conversely, you cannot copy items within CLINICAL_2A and CLINICAL_2B based on items in CLINICAL_3, even when they share a dictionary protocol such as DRUG2_DICT.

In other words, you can only copy downwards along one or more lines; that is, from an ancestor.

Note: When copying objects from one protocol to another, you also need to be aware of how a copy operation can create either connected or unconnected copies of objects. For more information on connected and unconnected copies, see Chapter 19.

Factors affecting the protocol searchlist

The protocol searchlist is determined by the following protocol attributes:

- Parent Protocol attribute

  A *parent protocol* is a protocol that defines the current protocol's place in the protocol hierarchy. It is also defines the default searchlist for the current protocol.

  When you create a view protocol, the parent protocol is automatically set to the base protocol and cannot be changed.

- Searchlist attribute

  A *searchlist* is a list of protocols from which the current protocol can copy Clintrial software object definitions (metadata). A protocol's searchlist can consist of protocols of any type.

  The setting of the protocol parameter DD_STRICT_SEARCH deter-mines which protocols are available to be included in the searchlist. If the value is Yes, only direct ancestors (the parent, the parent of the parent, and so forth) and siblings (protocols that have the same parent) can appear in the searchlist. If the value is No, any protocol in the database instance can be added to the searchlist.

  If the current protocol was created to allow searchlist modification, the designer of objects in the protocol can use the Protocol Browser to change the searchlist.

- Modify Searchlist attribute

  This attribute allows you, as the designer of objects within the protocol, to modify the searchlist by including other protocols that are in the same database. If this attribute is cleared, the searchlist cannot be modified, therefore, the default searchlist must be used. If this attribute is checked, the searchlist can be modified. The protocol parameter DD_STRICT_SEARCH determines the protocols available to be included in the modifiable searchlist.

  For a view protocol, the Modify Searchlist attribute is cleared and cannot be changed.

  Allowing searchlist modification for a protocol with a parent protocol defeats the enforcement of a protocol hierarchy, but increases flexibility.

What is the default searchlist?

The default searchlist for a protocol is defined as follows:

- For a view protocol, the parent protocol is always the base protocol and the searchlist consists only of the parent protocol (and cannot be modified).

  Note: A protocol's searchlist implicitly includes the protocol itself. For example, you can copy and rename a Clintrial software object within the current protocol, or you can copy an object from another protocol in the searchlist and optionally rename it. However, with a view protocol, you can only copy objects from the base protocol and you cannot rename them.
- If a protocol has no parent protocol, then the default searchlist contains no protocols.
- If a parent protocol is specified and DD_STRICT_SEARCH is set to Yes (the default value), then the default searchlist consists of the parent protocol and siblings (all children of that parent protocol). Otherwise, the default searchlist consists of the parent protocol.

Modifying the searchlist

If the protocol's Modify Searchlist attribute is checked, you can modify the protocol's searchlist. To modify the Searchlist, from the Protocol Browser, select a protocol, and then, from the **Protocol** menu, select **Search List**.

The following conditions apply to searchlist modification:

- DD_STRICT_SEARCH set to No — Any protocols in the database instance can be added to or removed from the searchlist.
- No parent protocol specified and DD_STRICT_SEARCH set to Yes — No protocols can be included in the searchlist, since DD_STRICT_SEARCH

requires that the searchlist be restricted according to the protocol hierarchy, but this protocol is not in a protocol hierarchy.

- Parent protocol specified and DD_STRICT_SEARCH set to Yes — The only protocols that can be added to the searchlist are this protocol's direct ancestors (the parent protocol, the parent of the parent, and so forth) and siblings (protocols that have the same parent). Any protocol can be removed from the searchlist.

Note: If you clear the Modify Searchlist attribute, then the **Search List** command on the **Protocol** menu is deactivated, and the searchlist is always the default searchlist.

How to control object creation and modification

In Design, to encourage standardization, you can limit the modification of any Clintrial software objects that are created in a protocol, or that are copied within a protocol or to another protocol. You can limit the modification of all objects in a protocol, or on an object-specific level by setting the following:

- The protocol's Can Modify Objects attribute

  If this attribute is checked, an object's Modifiable attribute determines whether it is modifiable. If this attribute is cleared, no objects in the protocol can be modified, regardless of the setting of an individual object's Modifiable attribute.

- An object's Modifiable attribute

  If a protocol's Can Modify Objects attribute is checked, each object's Modifiable attribute can be switched on or off to allow or disallow modification of the object. However, the Modifiable attribute is switched off and not available for toggling if the protocol's Can Modify Objects attribute is cleared.

  You can switch the Modifiable attribute by opening an object browser, and, from the object's menu, selecting **Set Modifiable Status**.

- The CREATE_OBJ_DD, CREATE_OBJ_FRM, and CREATE_OBJ_VLD protocol parameters

  If the CREATE_OBJ_DD protocol parameter is set to Yes, you can create schema objects like panels and items in the protocol. If the protocol parameter is set to No, you are limited to copying these objects from other protocols in the protocol's searchlist. Similarly, the CREATE_OBJ_FRM controls the creation of display objects like page sections, and the CREATE_OBJ_VLD controls the creation of validation objects like derivations.

For copied objects, the settings for the Modifiable attribute depend upon whether the copies are under revision control (connected or unconnected). For more information on connected and unconnected copies, see Chapter 19.

For more information on the protocol attributes Modify Searchlist and Can Modify Objects, see "Protocol attributes" on page 173.

For more information on the Modifiable attribute, see each object's attributes section.

For information on protocol parameters, see Chapter 5.

Why create protocols with no hierarchy?

You can create protocols without a hierarchy. These protocols would have either of two extreme organizations:

- All protocols have access to all other protocols — Objects in any protocol are based on objects in any other protocol. In this case all protocols appear in each others' searchlists.
- No protocol has access to another protocol — No protocol can copy an object from another protocol. In this case no protocol has a searchlist, and the designer of the objects in the protocol is not authorized to change searchlists. However, the designer must be authorized to create items, because items cannot be copied from another protocol.

Locking a protocol

You can lock a protocol so that no data can be entered, modified, or managed, and no metadata can be created or modified.

To lock a protocol:

1. In the Protocol Browser, click on the protocol.
2. From the **Protocol** menu, select **Protocol Lock**.
3. In the Change Protocol Lock State dialog box, specify a Reason for Change.
4. Save the changes.

View protocols

The Clintrial software allows you to determine which portion of the clinical data stored in a base protocol is available to users through the use of view protocols.

What is a view protocol?

A *view protocol* is a protocol that does not contain actual data but provides a view onto another protocol (that is, a base protocol). A view protocol can display a view of records based on either of the following:

- A checkpoint date that recreates the state of the clinical data as of midnight of a specified date.
- A view restriction, in the form of a SQL restriction clause, that selects records based on a condition.

You can create a view protocol with a checkpoint date or a view restriction, with both, or with neither. When you create a view protocol, Design asks if you want to copy the base protocol metadata into the view protocol. Then,

- If you choose to copy the metadata, the view protocol creates view panels based on all the base panels.
- If you choose not to copy the metadata, no view panels are created and you copy them individually after the view protocol is created.

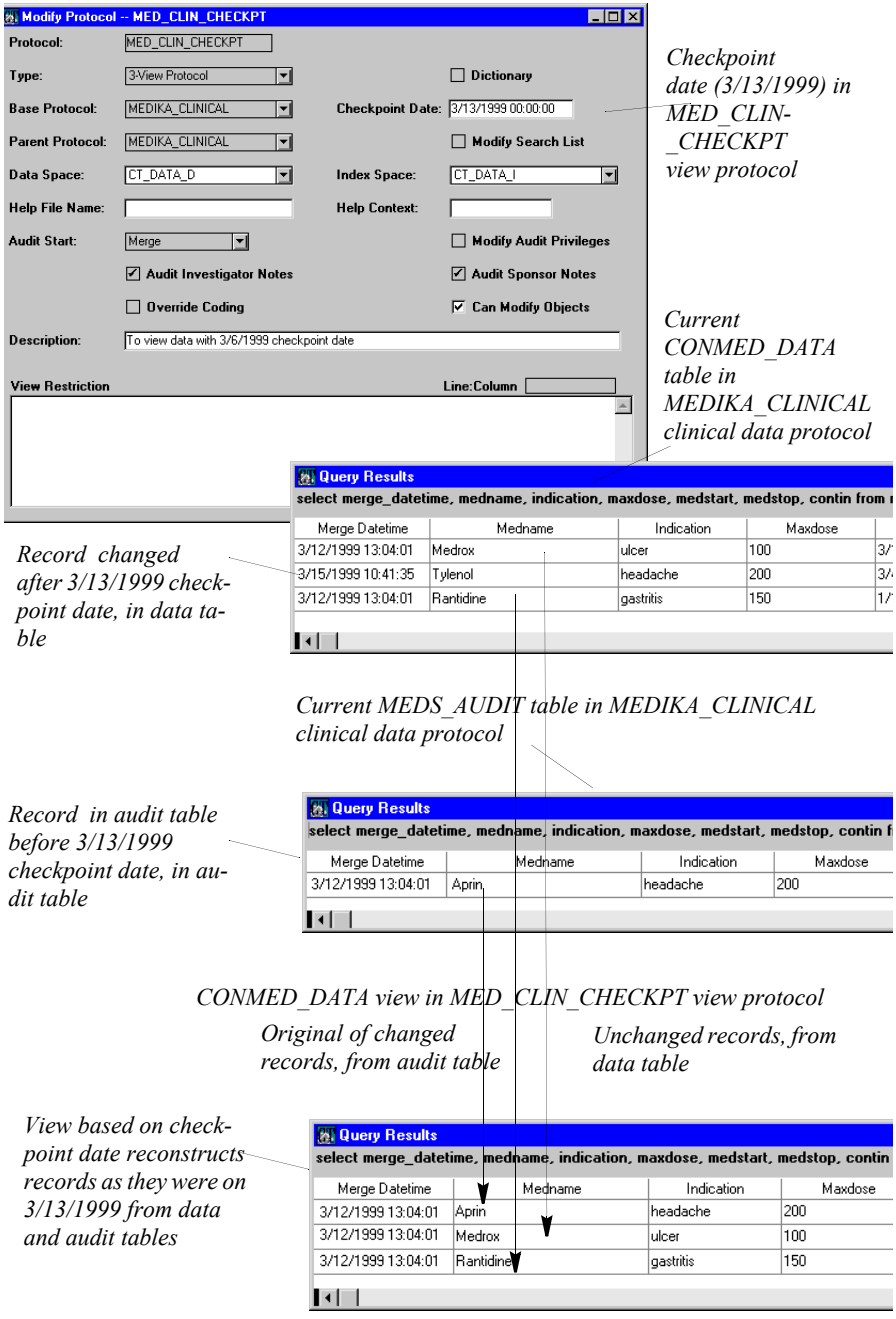
Note: You cannot import or export view protocols.

How does a checkpoint date work?

A *checkpoint date* in a view protocol allows you to create the appearance of the clinical data stored in a base protocol's database tables at a particular time. Checkpoint dates can be useful in the following situations:

- At certain points in a large, ongoing study, you need to conduct interim analyses to monitor the efficacy of a test compound.
- A regulatory agency requests an analysis of how the clinical data appeared at an earlier time (for example, when the NDA was filed), rather than an analysis of how the clinical data appears currently.
- As you gather more clinical data, you want to conduct periodic safety updates to monitor the adverse events and laboratory data for a test compound.

The following example shows how a checkpoint date in a view panel in a view protocol selects data for display:


Modify Protocol -- MED_CLIN_CHECKPT
Protocol: MED_CLIN_CHECKPT
Type: 3-View Protocol
Dictionary
Base Protocol: MEDIKA_CLINICAL
Checkpoint Date: 3/13/1999 00:00:00
Parent Protocol: MEDIKA_CLINICAL
Modify Search List
Data Space: CT_DATA_D
Index Space: CT_DATA_I
Help File Name:
Help Context:
Audit Start: Merge
Modify Audit Privileges
Audit Investigator Notes
Audit Sponsor Notes
Override Coding
Can Modify Objects
Description: To view data with 3/6/1999 checkpoint date
View Restriction
Line:Column
Checkpoint date (3/13/1999) in MED_CLIN-_CHECKPT view protocol
Current CONMED_DATA table in MEDIKA_CLINICAL clinical data protocol
Query Results
select merge_datetime, medname, indication, maxdose, medstart, medstop, contin from
Merge Datetime | Medname | Indication | Maxdose
3/12/1999 13:04:01 | Medrox | ulcer | 100
3/15/1999 10:41:35 | Tylenol | headache | 200
3/12/1999 13:04:01 | Rantidine | gastritis | 150
Record changed after 3/13/1999 checkpoint date, in data table
Current MEDS_AUDIT table in MEDIKA_CLINICAL clinical data protocol
Query Results
select merge_datetime, medname, indication, maxdose, medstart, medstop, contin f
Merge Datetime | Medname | Indication | Maxdose
3/12/1999 13:04:01 | Aprin | headache | 200
Record in audit table before 3/13/1999 checkpoint date, in audit table
CONMED_DATA view in MED_CLIN_CHECKPT view protocol
Original of changed records, from audit table
Unchanged records, from data table
Query Results
select merge_datetime, medname, indication, maxdose, medstart, medstop, contin
Merge Datetime | Medname | Indication | Maxdose
3/12/1999 13:04:01 | Aprin | headache | 200
3/12/1999 13:04:01 | Medrox | ulcer | 100
3/12/1999 13:04:01 | Rantidine | gastritis | 150
View based on checkpoint date reconstructs records as they were on 3/13/1999 from data and audit tables

Current data is stored in the data tables and copies of modified or deleted data are stored in the audit tables. Clintrial uses information stored in both the data tables and the audit tables of the base protocol to determine how the clinical data in the data tables appeared as of the checkpoint date. The view protocol uses:

- The data from the data tables for any panels whose data has not changed since the checkpoint date
- The data from the audit tables for any panels whose data has changed since the checkpoint date

In this example, the clinical data tables for the view protocol's view panel show:

- CONMED_DATA — Current records

  The DATA table contains any records that were entered before the March 13, 1999 checkpoint date and that were not changed, as well as the record that was changed and merged after the checkpoint date, on March 15, 1999.
- CONMED_AUDIT — Previous version of the changed record

  The AUDIT table contains the record as it appeared on the checkpoint date of March 13, 1999, before it was changed.
- MED_CLIN_CHECKPT — Data as displayed by the view panel

  The view protocol's CONMED_DATA view panel displays the data as it existed on the checkpoint date of March 13, 1999, two records from the CONMED_DATA table, and one record from the CONMED-_AUDIT table.

For information on the data and audit tables, see "What are update, data, and audit tables?" on page 189.

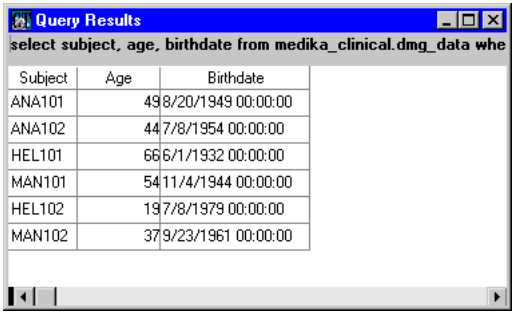
How does a view restriction work?

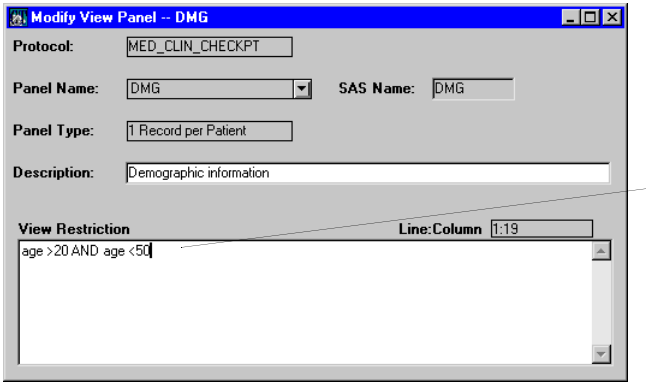
A SQL *view restriction* allows you to select records based upon a condition described using SQL. SQL restriction clauses in view protocols can be useful in the following situations:

- Users need access to clinical data based on its collection points (for example, only the clinical data collected for subjects in a given country or at a given center).
- Users need access only to the clinical data that meets certain criteria (for example, the clinical data for males over the age of 55, or for all geriatric subjects who have used a specified concomitant medication).
- Clinical data managers want to safeguard confidential subject information and, at the same time, provide access to the other clinical data in a base protocol.

The following figure shows how a restriction clause in a view protocol selects data for display:

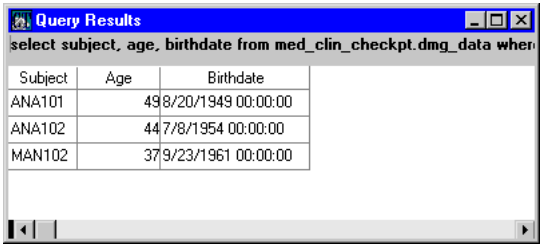
Records for a set of subjects from the DMG_DATA panel of the MEDIKA_CLINICAL clinical data protocol





View restriction in the DMG_DATA view of the MEDS_CLIN_CHECKPT view protocol

Records for subjects ages 21 to 49 from the DMG_DATA view of the MEDS_CLIN_CHECKPT view protocol, based on the view restriction



In this example, the SQL restriction clause used as a view restriction presents only those demographic records for subjects aged 21 through 49.

For information on SQL restriction clauses, see the *Reference Guide*.

You specify the checkpoint date or the restriction clause for all view panels in the view protocol when you create or modify the view protocol. After you create the view protocol, you create view panels for each set of tables to be accessed by a view protocol. If you create a view protocol without a checkpoint date or a restriction clause, the view panels include all the base panel data from the data tables.

The restriction clause that you specify in the view protocol is the default for all view panels in the view protocol. However, if you do not specify a restriction clause in the view protocol, you can set a restriction clause for one or more view panels in the view protocol. You can also set a restriction clause in the view panel that is different from the restriction clause set in the view protocol. If specified in the view panel, the restriction clause overrides the view protocol's restriction clause for that view panel.

However, you cannot override the view protocol's checkpoint date for individual view panels.

Note: When you save modifications to the checkpoint date, all view panels in the protocol are automatically updated to use the new date. The same is true when you save modifications to the default restriction clause, except that view panels which have their own explicit restriction clause are not updated, since they are not using the default restriction.

For more information about view panels, see Chapter 10.

Setting the audit start points

The Clintrial software allows for the auditing of data that is modified or deleted. The clinical data for a protocol is stored in three Oracle database tables per panel: update, data, and audit. If you enable auditing, you ensure that when a record in the update table or data table for a particular panel is modified, a copy of the unmodified record is placed in the audit table of that panel.

The auditing functions of Clintrial software protocols and panels are closely related to the collection, management, and analysis of the clinical data. Therefore, you must carefully consider your company's specific data management needs and the corresponding Clintrial software functionality before you set or modify these functions.

For information on data collection and management, including auditing, see *Manage, Classify, and Lab Loader.*

List of audit start points

Clintrial software auditing requires that you set an audit start point.

You can set an audit start point to one of five values: Entry, Verification, Validation, Validity, or Merge.

The most restrictive audit start point is Entry, because it results in auditing of all changes and deletions after the first data entry for a record. You might establish an audit start point of Entry if your company considers the discrepancy rate between first entry and verification entry to be valuable management information. However, such an approach uses substantial Oracle and disk resources.

The least restrictive audit start point is Merge. You might choose an audit start point of Merge if your company only needs to capture data changes occurring after the data is declared to be clean data. The Merge audit start point is the default start point unless you use the system parameter AUDIT_START_DEFAULT to set it differently.

Although you can change the audit start point at any time, you should be aware that any such changes can make the audit trail difficult to interpret.

For more information on the audit start point, see "Protocol attributes" on page 173.

For more information on panels, see Chapter 10.

Options for setting audit start points

Audit start points are specified for each panel as follows:

- Set a database-wide default audit start point for all panels in all protocols by setting the system parameter AUDIT_START_DEFAULT. The database-wide default is set through Admin.
- Set a protocol-specific audit start point for all panels in the protocol by setting the Audit Start attribute when you create or modify the protocol (if the system parameter AUDIT_PROTOCOL_MOD allows you to set protocol-specific audit start points).
- Set a panel-specific audit start point if the Modify Audit Privileges protocol attribute was enabled when the protocol was created.

Factors in creating audit start points

Consider the following in deciding whether to set the audit start point as database-wide, protocol-specific, or panel-specific:

- Ease of administration — A database-wide audit start point for all panels in all protocols is simpler to set up.
- Consistency — Database-wide or protocol-specific audit start points enforce more consistency of the types of information collected in the audit trail than do panel-specific audit start points. In particular, if you allow differing panel-specific audit start points, then panels that collect similar types of information may have inconsistent audit records that make the audit trails difficult to interpret.
- Granularity — Some panels in a protocol may require more rigorous auditing (that is, a more restrictive audit start point) than others. If so, panel-specific audit start points offer greater flexibility than database-wide or protocol-specific audit start points.

For more information on how to set the AUDIT_START_DEFAULT and AUDIT_PROTOCOL_MOD system parameters, see Chapter 5.

Protocol attributes

This section describes the specific attributes of a protocol. Some of these attributes are set for you; others you must set. Before planning and creating the logical design of your protocols, you must consider the functionality represented by the protocol attributes. For example, to design the protocol hierarchy most effectively, you must understand the specifics of searchlist functionality.

When creating or modifying a protocol, you set the protocol attributes, which determines in part how the constituent objects within the protocol (such as panels and items) are created. The designer of the protocol may or may not be the same as the designer of the constituent objects within the protocol.

The following sections name and describe the protocol attributes.

Protocol

The name of the protocol.

Type

You must specify a protocol type to categorize the protocol by its purpose or properties. The purpose of protocol types helps enforce standardization, categorization, and the separation of metadata and data in the protocol hierarchy. The types of protocols are:

- Clinical Data (Type 1) — The most general protocol type, for protocols that collect clinical data and store it in database tables. However, you can also use the clinical data type for a protocol that stores only metadata.
- Coding Thesaurus (Type 2) — For protocols created to store coding thesauruses.
- View Protocol (Type 3) — For protocols that do not contain actual clinical data, but which provide views of base protocols.
- Lab Loader (Type 4) — For protocols created to store lab normal data and clinical lab data, and to transfer records using Lab Loader.

Note: In addition, when creating a protocol, you can specify the Dictionary attribute for any protocol type except a view protocol. Specifying the Dictionary attribute creates the protocol as a data dictionary protocol. For more information on the Dictionary attribute and data dictionary protocols, see page 159 and the following section.

For information on coding thesaurus protocols, see Chapter 14.

For information on Lab Loader, see *Manage, Classify, and Lab Loader*.

Dictionary

Specifies whether the protocol is a dictionary protocol. You typically use a dictionary protocol to store standard metadata. A protocol with the dictionary attribute cannot create associated clinical data tables.

You can define the following protocol types as dictionary protocols:

- Clinical Data (Type 1)
- Coding Thesaurus (Type 2)
- Lab Loader (Type 4)

Note: To create a protocol as a dictionary, you must check the Dictionary attribute when creating the protocol. You cannot modify this attribute (either to activate or deactivate it) after you have created the protocol.

Base Protocol

Select a protocol upon which to base the view protocol. (Applies to view protocols only.)

Parent Protocol

A protocol that defines the current protocol's place in the protocol hierarchy. It is also used to define the default searchlist.

A *searchlist* is a list of protocols from which you can later copy Clintrial software object definitions into the protocol you are creating. Specify the searchlist from the Protocol Browser.

Specify a parent protocol when you want to encourage the use of a protocol hierarchy that uses data dictionary protocols for the definition of metadata.

Note: When you create a view protocol, the parent protocol is automatically set to the base protocol and cannot be changed.

For more information on the default searchlist, see page 163.

Checkpoint Date

Specify a checkpoint date. The view panels you subsequently create for this view protocol will display the data from the base protocol's tables as it appeared on midnight of that date. (Applies to view protocols only.)

Modify Searchlist

This attribute allows the designer of objects within the protocol to modify the searchlist by including other protocols that are in the same database. If this attribute is cleared, the searchlist cannot be modified and so the default searchlist must be used. If this attribute is checked, the searchlist can be modified as described in the section "Modifying the searchlist" on page 163. The protocol parameter DD_STRICT_SEARCH determines the protocols available to be included in the modifiable searchlist.

Note: For a view protocol, the Modify Searchlist attribute is cleared and cannot be changed.

The tablespace to use for clinical data tables (for example, CT_DATA_D). Installing an Oracle database for use by Clintrial software creates four tablespaces in addition to the SYSTEM tablespace required by Oracle. The tablespaces and their contents are:

- CT_META_D — Clintrial software metadata tables
- CT_DATA_D — Clintrial software clinical data tables
- CT_META_I — Clintrial software metadata indexes
- CT_DATA_I — Clintrial software clinical data indexes

The tablespaces CT_META_D and CT_META_I are mandatory. They are used to store Clintrial software data dictionary tables and indexes, respectively. The other tablespaces (CT_DATA_D and CT_DATA_I) are the default tablespaces for the tables and indexes used to store Clintrial software data. Your choices are all existing Oracle tablespaces except for:

- The SYSTEM, CT_META_D, and CT_META_I tablespaces
- Any tablespaces ending with _I (reserved for indexes)

You can choose to use the default CT_DATA_D tablespace to store clinical data, or you can choose to create additional tablespaces for the creation of tables within a project. Doing the latter provides the following benefits:

- Input/Output (I/O) optimization

  If two distinct tablespaces are set up for each project (one for data storage tables and one for indexes), and if these tablespaces are placed on different devices, I/O is improved. Oracle is now able to fetch the current row while simultaneously looking in the index for the next row to fetch. Also, projects with heavy I/O use can be put on separate disk drives.
- Allocation of tablespace based on project size

  Using different tablespaces allows for the allocation of appropriately sized tablespaces based on project size.
- Project backup and archiving

  Tablespaces can be taken off-line while leaving the rest of the database available. Therefore, you can group protocols into project-level tablespaces for backup and archiving purposes without affecting the use of protocols that use other tablespaces.

For more information on tablespaces, see the Oracle documentation.

Index Space

The tablespace to use for indexes (for example, CT_DATA_I). Your choices are all existing Oracle tablespaces except for:

- The SYSTEM, CT_META_D, and CT_META_I tablespaces
- Any tablespaces ending with _D (reserved for data)

You can choose to use the default CT_DATA_I tablespace to store clinical data indexes, or you can choose to create additional tablespaces, particularly for the creation of indexes on a project level.

Help File Name

The name of a Windows Help file that you can optionally create to associate with this protocol — for use in Enter when, from the **Help** menu, the user selects the **About Protocol**.

The default context point for your Help file is the first topic in your Help file.

If you do not specify your own Windows Help file here, when the user selects **About Protocol**, Clintrial software Help appears, describing how to create your own Windows Help file.

Help Context

A context number for the topic in your Windows Help file — for use in Enter when the user selects the **About Protocol** menu choice, if you want that topic to replace the default topic in your Windows Help file.

You can also set context numbers for individual study pages, to override the default set by Help File Name, or the context point set by Help Context.

For more information on setting Help context numbers for a study page, see Chapter 18.

Audit Start

Check this attribute to set a default point at which auditing begins in each panel created in this protocol. The start points are:

- Entry — Auditing begins after data has been entered interactively in a record. The record is in the update table.

  Note: Batch-loaded records cannot be audited until they pass screening; that is, until their record status becomes 1.
- Verification — Auditing begins after data has been reentered to check the accuracy of the entry, or, for batch-loaded records, after data has been screened. The record is in the update table.

  Note: No changes to data are stored during blind verification, so there is nothing to audit.
- Validation — Auditing begins after a record has been validated, regardless of whether the record passed or failed validation. The record is in the update table.
- Validity — Auditing begins after a record has been validated and has passed the validation. The record is in the update table.
- Merge — Auditing begins after data has moved from the update table to the data table. The record is in the data table.

Audit Investigator Notes

Check this attribute if you want to audit notes recorded by the investigator (that is, people at the data collection site).

Override Coding

This attribute is only available if you are using the Clintrial application to do your coding. This is handled by the Oracle Central Coding application if the protocol parameter USE_CENTRAL_CODING is Yes.

When you use Clintrial for coding, this attribute specifies whether, during automatic coding, Manage can override a value for an interactively coded verbatim text item, as follows:

- If you clear this attribute, no coding occurs during automatic coding for an item whose last coding was interactive.
- If you check this attribute, coding occurs during automatic coding for an item whose last coding was interactive.

During interactive coding, the Manage user can always code the item, regardless of the setting of this attribute.

For information on interactive coding, see *Manage, Classify, and Lab Loader*.

Modify Audit Privileges

Check this attribute to allow the setting of the audit start point on a panel-by-panel basis when you create panels for this protocol.

Audit Sponsor Notes

Check this attribute if you want to audit notes recorded by the sponsor (that is, users at your company).

Can Modify Objects

Check this attribute to allow modification of objects in this protocol. Clear to disallow modifications.

Description

An optional description of the protocol.

View Restriction

Specify a SQL restriction clause for a view protocol. The view panels you subsequently create for this view protocol will display the current data from the base protocol's tables based on the restriction clause. (Applies to view protocols only.)

Read-only attributes

When you select the **Show** command from the **Protocol** menu, the Clintrial software displays the previously described protocol attributes as read-only, and also displays the following additional read-only attributes:

- Creator — Name of the user account that created the protocol.
- Create Date — Date and time of the protocol creation.
- Version — Clintrial software version number.
- Export Site — If this is an imported protocol, the database registration name for the site that exported the protocol.

- Status — One of the following:
  - CREATING — During protocol creation, this status is temporarily set while the protocol is being created to ensure that only one protocol is created at a time. When you try saving a newly created protocol, the Clintrial software server uses this status to determine if any other protocols are currently being created. If so, an error occurs and the creation request fails. This feature prevents locking conflicts that cause system performance to deteriorate.
  - NORMAL — The normal status for a created protocol.
  - PT_DD_INSTALLED —Only data dictionary metadata is created (protocol import status).
  - PT_EMPTY — For an imported protocol, ready to create local copy (protocol import status).
  - PT_ERROR — For an imported protocol, after creation of a local protocol, need to do reconciliation of codelists, items, flags, or notes (protocol import status).
  - PT_CHKD — For an imported protocol, after creation of local protocol and reconciliation, ready to install (protocol import status).
  - PT_INSTALLED — Data dictionary metadata and empty clinical data tables are created (protocol import status).
  - PT_OK — During metadata installation, this status is temporarily set after copying the metadata but before installing the panels. If a failure occurs during metadata installation, the process can start where it was interrupted (protocol import status).
  - PT_LOADED — During the Release step, this status is temporarily set after all required panel tables have been installed. The Load Data step is prevented if the protocol has this status (protocol import status).
  - MD_EMPTY — For a distributed protocol, ready to create the protocol account during the protocol acceptance process (part of Multisite Metadata Distribution).
  - MD_CREATED — For a distributed protocol, ready to create data dictionary metadata during the protocol acceptance process (part of Multisite Metadata Distribution).
- Locked — Checked if the protocol is locked.
- Enrollment Panel — Name of a panel that stores subject enrollment data.
- Subject Item — Name of a subject-related context item that you specify in the context panel to uniquely identify the subject.
- Block Key Item — Name of a visit-related context item that is a key for identifying data by block for display in Enter. For a non-repeating block, the value of the Block Key Item uniquely identifies the block; for a repeating block, the values of the Block Key Item and the Block Repeat Key Item uniquely identify the block.

- Block Repeat Key Item — Name of a visit-related context item that is the key, with the Block Key Item, for identifying data by block for display in a repeating block in Enter.
- Page Key Item — Name of a page-related context item that is the key for identifying data by study page within a block for display in Enter. For a nonrepeating study page, the value of the Page Key Item uniquely identifies the page within the block; for a repeating page, the values of the Page Key Item and the Page Repeat Key Item uniquely identify the study page within the block.
- Page Repeat Key Item — Name of a page-related context item that is the key, with the Page Key Item, for identifying data by page for display in a repeating page in Enter.
- Error Log Item — Name of an optional item from the context panel that is displayed with the Error log to identify records.
- View Object ID — A unique identifier used internally to identify the view restriction text.
- Protocol Number — A unique internal identifier for the protocol.
- Protocol Searchlist — The list of protocols from which the designer of objects for this protocol can copy objects.

Note: With the Protocol Browser open, you can also select the **Show** command from the **Protocol** menu to display protocol attributes.

Note: An additional field called *Line:Column* appears above the View Restriction field. The Line:Column field dynamically tracks the position of the cursor in the view restriction (active for view protocols only) and is not a protocol attribute.

Deleting a protocol

When you delete a protocol, all data and metadata for that protocol are deleted.

You cannot delete a protocol that is locked. For more information, see Locking a protocol.

Caution: Clintrial Design has no way of knowing if a CIS protocol is currently being synched and therefore there is no check for this condition when deleting the protocol from the Clintrial server. All protocol data will be lost if an active protocol is deleted.

To delete protocols:

1. Refresh the Protocol Browser.
2. Select one or more protocols to delete.
3. From the Protocol menu, select Delete. You are prompted to confirm the deletion for each protocol you selected.
4. Click Yes to delete the protocol, or No to cancel the deletion.

10 Panels and Items

Overview

This chapter describes panels and items. Within a protocol, you:

1. Create one or more panels.
2. Create one or more items for the panel.
3. Install the panel.

What is a panel?

A *panel* is a Clintrial software object that groups together a set of logically or clinically related items.

A panel is defined by metadata that is stored in data dictionary tables. When you install a panel, three additional database tables are created (the update table, data table, and audit table) for the storage of clinical data.

What is an item?

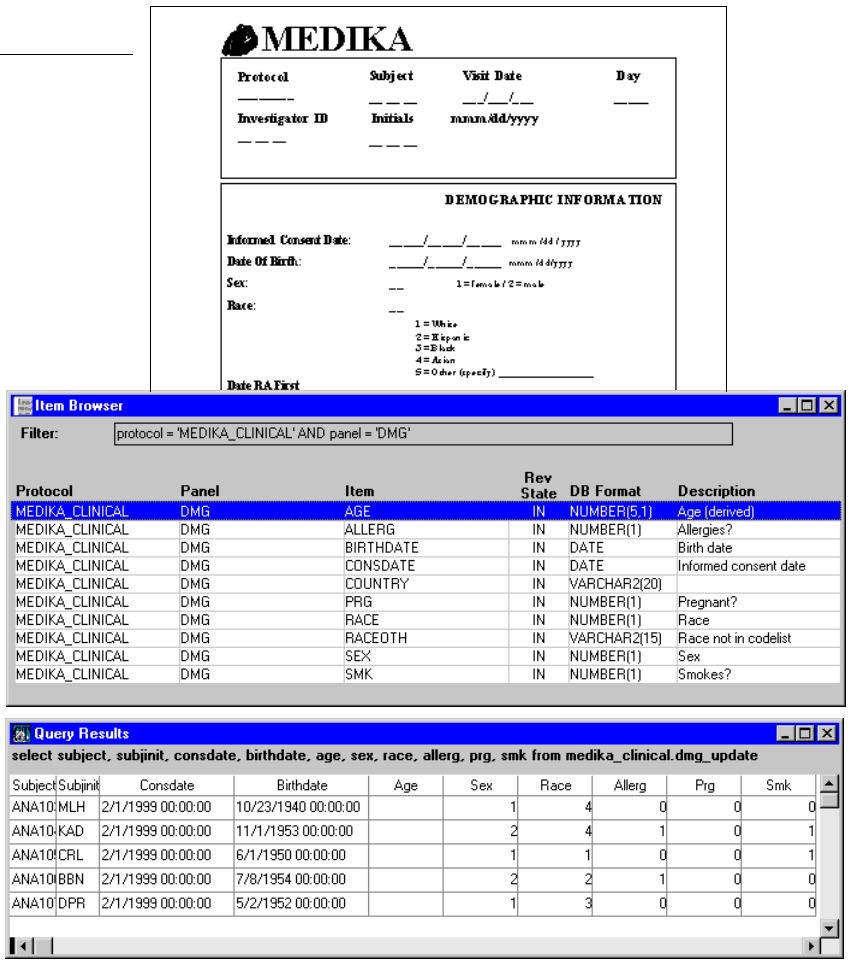
An *item* is a Clintrial software object that stores a piece of data, for example, the data collected by a single field in a study page, or a single field in a batch-loaded file. Items are defined within panels, and each corresponds to one column in a clinical data table. For example, the DMG panel used by the MEDIKA study book contains the items AGE and SEX.

The following figure shows the relationship between a section of a CRF page used to enter clinical data, the DMG panel and its items, and the data tables that store the clinical data.

CRF page contains data to be entered in Enter.

DMG panel and items that define clinical data tables.

Clinical data table DMG_UPDATE stores entered data.



Database tables for panels and items

Oracle database tables store two types of information for panels and items:

- Metadata — Panel and item definitions as specified in Design by panel and item attributes.

  When you create or modify a panel or an item, and when you install a panel, the Clintrial software stores the metadata definition of the panel or item in data dictionary tables.
- Clinical data—For the clinical data collected for each item in the panel.

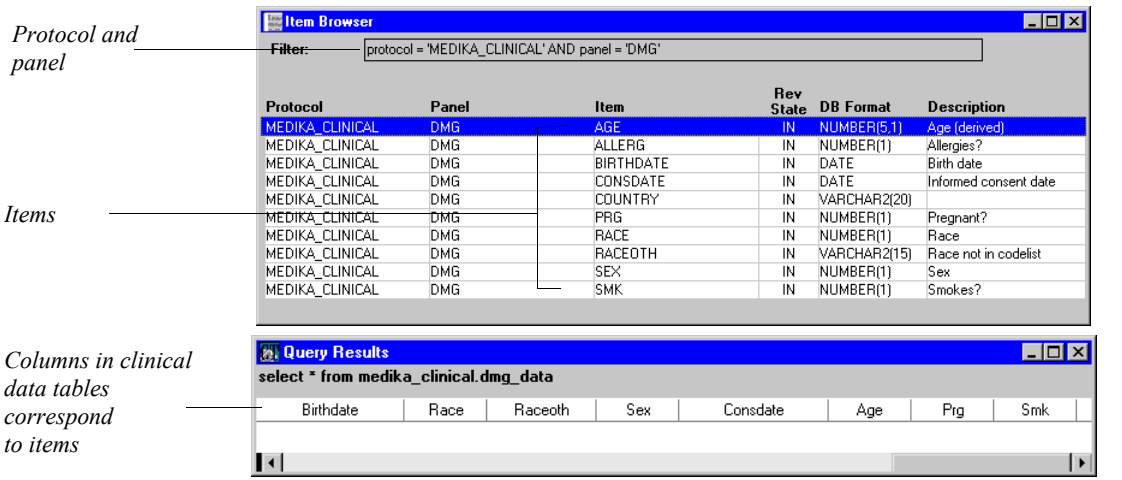
  When you install a panel you can optionally create clinical data tables to store data. If you do not choose to create clinical data tables, the installation creates only the metadata for the panels and items.

For more information on creating and installing panels, see "Creating and installing a panel and its items" on page 221.

For more information on panel and item attributes, see "Panel attributes" on page 226 and "Item attributes" on page 231.

How panels and items relate to clinical data tables

Panels define the Oracle database tables that store the clinical data. The following figure shows the relationship of a panel and its items to the underlying clinical data tables:



When are clinical data tables created?

After you create a panel, you can install it. When a panel is installed, you can optionally create the three clinical data tables: the update table, data table, and audit table.

What are update, data, and audit tables?

When you install panels you can optionally choose to create clinical data tables. The Clintrial software creates the following clinical data tables for each installed panel (except for the context panel) in the protocol:

- An update table for clinical data that is still being worked on (for example, in the process of being validated), and is not ready to be moved to the data table.

  The data is stored here as it is saved in Enter or batch-loaded in Manage.
- A data table only for clinical data that has passed validation.

  The data is stored here after it has been merged in Manage.
- An audit table for copies of clinical data as it was before modification if it was entered in Enter, or batch-loaded in Manage and then modified in Enter.

  The point at which data is written to the audit table depends on where the audit start point is set for the panel.

For information on the context panel, see "Context panels and context items" on page 199.

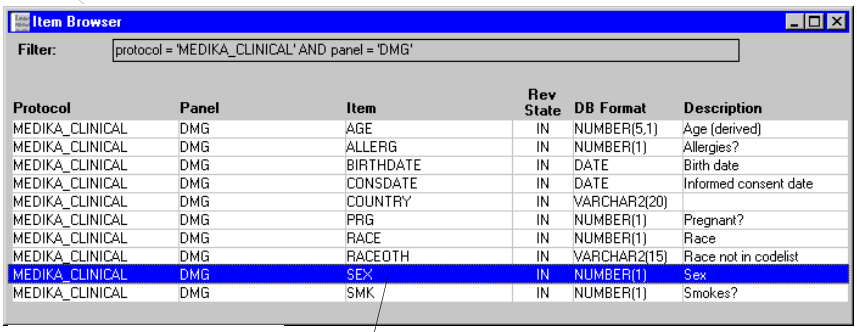
For information on merging data, see *Manage, Classify, and Lab Loader*.

For information on setting the audit start point, see "Setting the audit start points" on page 171.

Example

The following example shows the relationship of a panel and its items to the data dictionary table and the clinical data table:

Panel and items

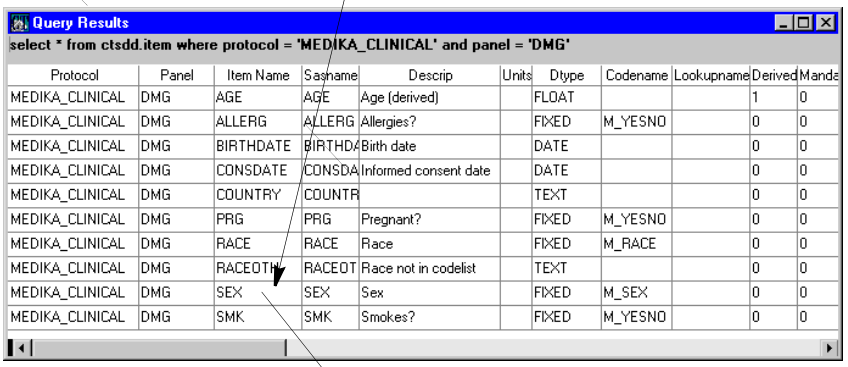


Item Browser

Filter: protocol = 'MEDIKA_CLINICAL' AND panel = 'DMG'

Protocol	Panel	Item	Rev State	DB Format	Description
MEDIKA_CLINICAL	DMG	AGE	IN	NUMBER(5,1)	Age (derived)
MEDIKA_CLINICAL	DMG	ALLERG	IN	NUMBER(1)	Allergies?
MEDIKA_CLINICAL	DMG	BIRTHDATE	IN	DATE	Birth date
MEDIKA_CLINICAL	DMG	CONSDATE	IN	DATE	Informed consent date
MEDIKA_CLINICAL	DMG	COUNTRY	IN	VARCHAR2(20)	
MEDIKA_CLINICAL	DMG	PRG	IN	NUMBER(1)	Pregnant?
MEDIKA_CLINICAL	DMG	RACE	IN	NUMBER(1)	Race
MEDIKA_CLINICAL	DMG	RACEOTH	IN	VARCHAR2(15)	Race not in codelist
MEDIKA_CLINICAL	DMG	SEX	IN	NUMBER(1)	Sex
MEDIKA_CLINICAL	DMG	SMK	IN	NUMBER(1)	Smokes?

Data dictionary table

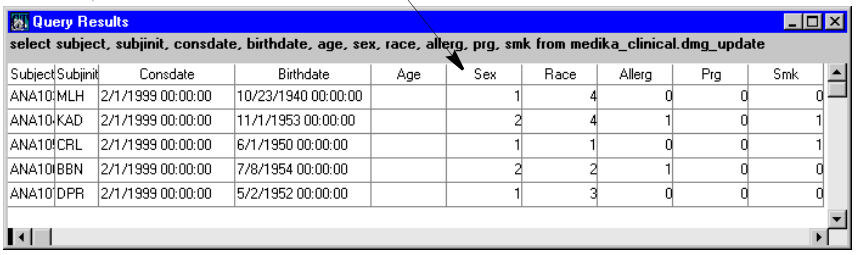


Query Results

select * from ctsdd.item where protocol = 'MEDIKA_CLINICAL' and panel = 'DMG'

Protocol	Panel	Item Name	Sasname	Descrip	Units	Dtype	Codename	Lookupname	Derived	Manda
MEDIKA_CLINICAL	DMG	AGE	AGE	Age (derived)		FLOAT			1	0
MEDIKA_CLINICAL	DMG	ALLERG	ALLERG	Allergies?		FIXED	M_YESNO		0	0
MEDIKA_CLINICAL	DMG	BIRTHDATE	BIRTHDA	Birth date		DATE			0	0
MEDIKA_CLINICAL	DMG	CONSDATE	CONSDA	Informed consent date		DATE			0	0
MEDIKA_CLINICAL	DMG	COUNTRY	COUNTR			TEXT			0	0
MEDIKA_CLINICAL	DMG	PRG	PRG	Pregnant?		FIXED	M_YESNO		0	0
MEDIKA_CLINICAL	DMG	RACE	RACE	Race		FIXED	M_RACE		0	0
MEDIKA_CLINICAL	DMG	RACEOTH	RACEOT	Race not in codelist		TEXT			0	0
MEDIKA_CLINICAL	DMG	SEX	SEX	Sex		FIXED	M_SEX		0	0
MEDIKA_CLINICAL	DMG	SMK	SMK	Smokes?		FIXED	M_YESNO		0	0

Clinical data table



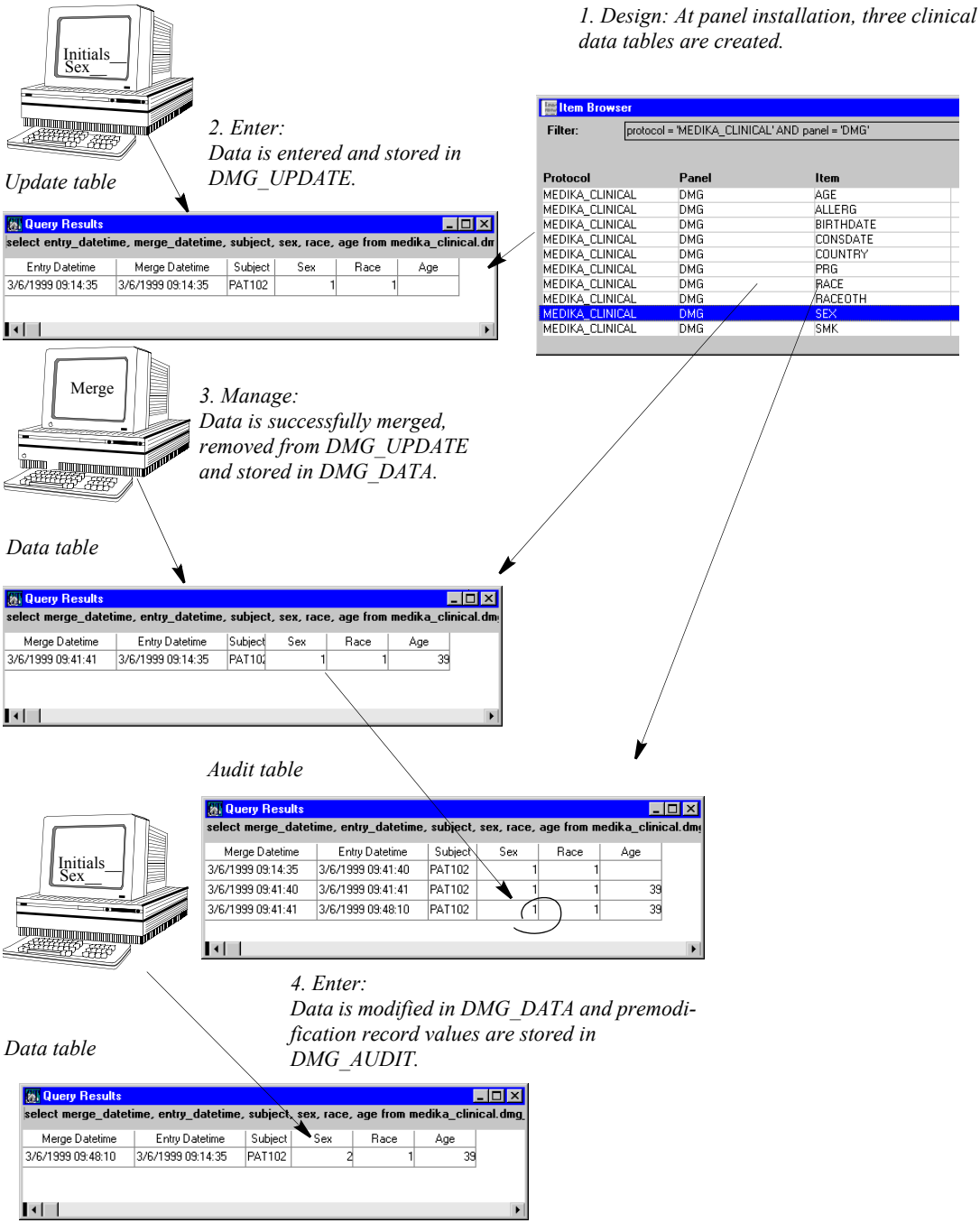
Query Results

select subject, subjinit, consdate, birthdate, age, sex, race, allerg, prg, smk from medika_clinical.dmg_update

Subject	Subjinit	Consdate	Birthdate	Age	Sex	Race	Allerg	Prg	Smk
ANA10	MLH	2/1/1999 00:00:00	10/23/1940 00:00:00		1	4	0	0	0
ANA10	KAD	2/1/1999 00:00:00	11/1/1953 00:00:00		2	4	1	0	1
ANA10	CRL	2/1/1999 00:00:00	6/1/1950 00:00:00		1	1	0	0	1
ANA10	BBN	2/1/1999 00:00:00	7/8/1954 00:00:00		2	2	1	0	0
ANA10	DPR	2/1/1999 00:00:00	5/2/1952 00:00:00		1	3	0	0	0

In this example, the panel has been installed. The item attributes that define the metadata for the SEX item are stored in a data dictionary table. The Clintrial software uses this metadata to create the three clinical data tables for each panel. (Only the update table is shown here.) In the clinical data tables, SEX is a column, as defined by the metadata.

The following example shows the use of the three clinical data tables associated with the DMG panel:



In this example, the audit start point is set to Merge (the default). The subject with the PATINIT value of ARX12 had an incorrect SEX value of 2 (female). The data was merged, so that record is no longer stored in the update table. After merging, the data was edited to change the SEX value to 1 (male). This resulted in the premodification record (SEX=2) being moved from the data table to the audit table.

For complete descriptions of the data dictionary tables, see the *Reference Guide.*

Panel types

This section describes the six Clintrial software panel types and which panel types allow repeating items.

What is a panel type?

A *panel type* is a number (0 – 5) that defines how the database tables associated with the panel (the update, data, and audit tables) are structured; that is, whether there can be one or multiple records for each subject item, or subject item and block key item combination.

The following table summarizes this information for the different panel types:

Panel type:	**Description:**	**Examples from MEDIKA_CLINICAL protocol:**
Non-Patient Data (Type 0)	Data is not related to a specific subject.	LAB_NORMS (Normal ranges for tests)
1 Record per Patient (Type 1)	One record can be collected only once during the clinical study for each subject.	DMG (Demographic Information)
>1 Record per Patient (Type 2)	Multiple records can be collected once in the clinical study for each subject.	MEDHIST (Medical History)

Panel type:	Description:	Examples from MEDIKA_CLINICAL protocol:
1 Record per Patient Visit (Type 3)	One record can be collected for each subject visit.	VITAL (Vital Signs)
>1 Record per Patient Visit (Type 4)	Multiple records can be collected for each subject visit.	ADV (Adverse Experience(s))
Subject Enrollment (Type 5)	Contains one record for each enrolled subject.	ENROLL

Caution: Design does not enforce this limitation on what data you can store in a panel's clinical data tables. However, other Clintrial software modules, such as Manage, may report errors when you carry out actions on data that does not conform to these limitations. Therefore, it is strongly recommended that you design your panels to conform to these limitations.

Non-Patient Data (Type 0)

A Type 0 panel stores data that is not related to a particular subject or visit, such as standard coding thesauruses, view codelists, or laboratory normal ranges. Neither a subject identifier nor a visit identifier is required for a Type 0 panel.

The following sample clinical data table contains laboratory normal ranges for a Type 0 panel called LAB_NORMS:

Query Results

select labtestname, hi_range, low_range, lab_type from medika_clinical.lab_n

Labtestname	Hi Range	Low Range	Lab Type
ALAT(SGPT)	100	50	Chemistry
ASAT(SGOT)	60	30	Chemistry
hematocrit	50	35	Hematology
hemoglobin	12	8	Hematology
white cell count	10	5	Hematology
color	yellow	amber	Urinalysis
glucose	TRACE	1% or 3+	Urinalysis

Note: Panel keys are strongly suggested for Type 0 panels. (During verification, the panel key values are filled in for all records, and are not editable.)

1 Record per Patient (Type 1)

A Type 1 panel stores data that is collected only once for each subject. The data is not related to a particular visit. For example, demographic data or eligibility data might be collected by a Type 1 panel. A subject identifier is required for a Type 1 panel.

The following example shows part of a clinical data table for a Type 1 panel called DMG_DATA:

Query Results

select subject, visno, visdate, consdate, birthdate, sex, race, age from medika_clinical.dmg_data

Subject	Visno	Visdate	Consdate	Birthdate	Sex	Race	Age
ANA101	0	2/1/1999 00:00:00	2/1/1999 00:00:00	8/20/1949 00:00:00	1	1	49
ANA102	0	2/1/1999 00:00:00	2/1/1999 00:00:00	7/8/1954 00:00:00	2	2	44
HEL101	0	2/1/1999 00:00:00	2/1/1999 00:00:00	6/1/1932 00:00:00	1	1	66
MAN101	0	2/1/1999 00:00:00	2/1/1999 00:00:00	11/4/1944 00:00:00	1	1	54
HEL102	0	2/1/1999 00:00:00	2/1/1999 00:00:00	7/8/1979 00:00:00	2	2	19
MAN102	0	2/1/1999 00:00:00	2/1/1999 00:00:00	9/23/1961 00:00:00	2	2	37
PAT102	0	2/12/1999 00:00:00	2/12/1999 00:00:00	5/11/1959 00:00:00	2	1	39

In this example, there is one record for each subject, identified by the SUBJECT item. The data is collected only once for a Type 1 panel, so the VISNO item has the same value for each record.

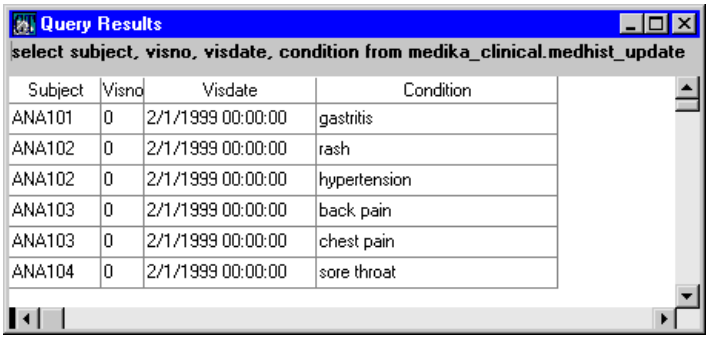
Caution: Design does not enforce this limitation on what data you can store in a panel's clinical data tables. However, other Clintrial software modules, such as Manage, may report errors when you carry out actions on data that does not conform to these limitations. Therefore, it is strongly recommended that you design your panels to conform to these limitations.

> 1 Record per Patient (Type 2)

A Type 2 panel stores data that is collected once in a clinical study, but possibly multiple times for each subject (that is, an item in the panel might have multiple values on the single occasion that data is collected for that item). The data is not related to a particular visit. For example, previous medications data might be collected by a Type 2 panel. A subject identifier is required for a Type 2 panel.

Panel keys are strongly suggested for Type 2 panels.

The following figure shows part of a clinical data table for a Type 2 panel MEDHIST_UPDATE:



Subject	Visno	Visdate	Condition
ANA101	0	2/1/1999 00:00:00	gastritis
ANA102	0	2/1/1999 00:00:00	rash
ANA102	0	2/1/1999 00:00:00	hypertension
ANA103	0	2/1/1999 00:00:00	back pain
ANA103	0	2/1/1999 00:00:00	chest pain
ANA104	0	2/1/1999 00:00:00	sore throat

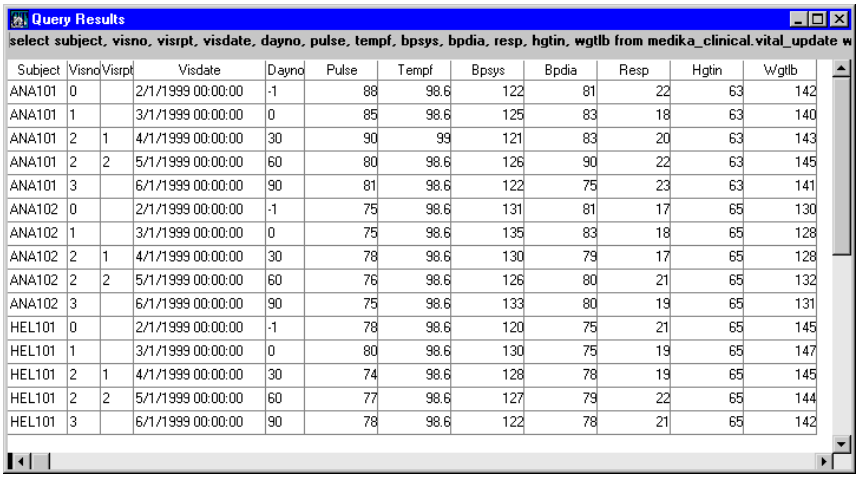
In this example, there are multiple records for subjects with the SUBJECT values ANA102 and ANA103. Each of these subjects has multiple records storing data about previous medical conditions. The data is collected only once for a Type 2 panel, so the VISNO item has the same value for each record.

Caution: Design does not enforce this limitation on what data you can store in a panel's clinical data tables. However, other Clintrial software modules, such as Manage, may report errors when you carry out actions on data that does not conform to these limitations. Therefore, it is strongly recommended that you design your panels to conform to these limitations.

1 Record per Visit (Type 3)

A Type 3 panel stores data that is collected once each for multiple subject visits (although not necessarily for all visits). For example, if a subject's vital signs are collected each visit, you might collect them in a Type 3 panel. A subject identifier and visit identifier are required for a Type 3 panel.

The following example shows part of a clinical data table for a Type 3 panel called VITAL_UPDATE:



Query Results

select subject, visno, visrpt, visdate, dayno, pulse, tempf, bpsys, bpdia, resp, hgtin, wgtlb from medika_clinical.vital_update w

Subject	Visno	Visrpt	Visdate	Dayno	Pulse	Tempf	Bpsys	Bpdia	Resp	Hgtin	Wgtlb
ANA101	0		2/1/1999 00:00:00	-1	88	98.6	122	81	22	63	142
ANA101	1		3/1/1999 00:00:00	0	85	98.6	125	83	18	63	140
ANA101	2	1	4/1/1999 00:00:00	30	90	99	121	83	20	63	143
ANA101	2	2	5/1/1999 00:00:00	60	80	98.6	126	90	22	63	145
ANA101	3		6/1/1999 00:00:00	90	81	98.6	122	75	23	63	141
ANA102	0		2/1/1999 00:00:00	-1	75	98.6	131	81	17	65	130
ANA102	1		3/1/1999 00:00:00	0	75	98.6	135	83	18	65	128
ANA102	2	1	4/1/1999 00:00:00	30	78	98.6	130	79	17	65	128
ANA102	2	2	5/1/1999 00:00:00	60	76	98.6	126	80	21	65	132
ANA102	3		6/1/1999 00:00:00	90	75	98.6	133	80	19	65	131
HEL101	0		2/1/1999 00:00:00	-1	78	98.6	120	75	21	65	145
HEL101	1		3/1/1999 00:00:00	0	80	98.6	130	75	19	65	147
HEL101	2	1	4/1/1999 00:00:00	30	74	98.6	128	78	19	65	145
HEL101	2	2	5/1/1999 00:00:00	60	77	98.6	127	79	22	65	144
HEL101	3		6/1/1999 00:00:00	90	78	98.6	122	78	21	65	142

The vital signs data is collected once during each of the subject's visits. For example, each subject with in this example has a single record for each of the visits in the study (where VISNO = 0, 1, and 3, and the repeating block combination of VISNO and VISRPT = 2/1 and 2/2).

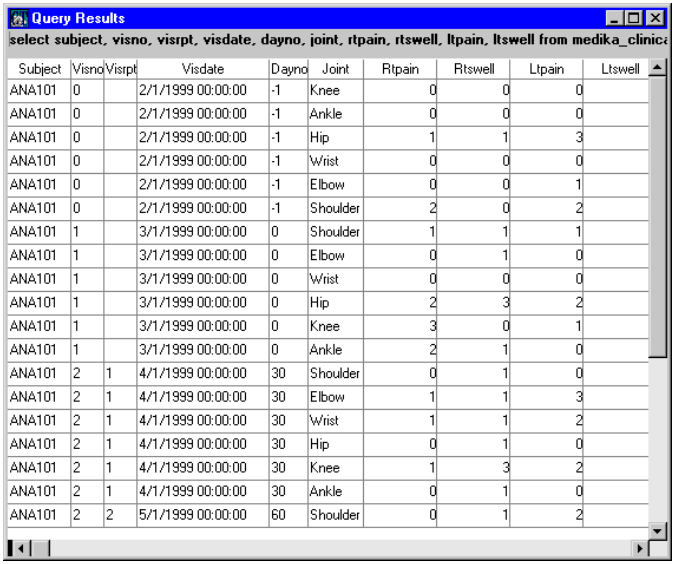
Caution: Design does not enforce this limitation on what data you can store in a panel's clinical data tables. However, other Clintrial software modules, such as Manage, may report errors when you carry out actions on data that does not conform to these limitations. Therefore, it is strongly recommended that you design your panels to conform to these limitations.

> 1 Record per Visit (Type 4)

A Type 4 panel stores data that is collected multiple times during a subject visit (that is, an item in the panel might have multiple values entered on multiple visits). The data need not be collected on every visit. For example, concomitant medications data might be collected by a
Type 4 panel called MEDS.

Panel keys are strongly suggested for Type 4 panels.

The following example shows part of the clinical data table JNTASM_UPDATE, based on a Type 4 panel JNTASM:



Query Results

select subject, visno, visrpt, visdate, dayno, joint, rtpain, rtswell, ltpain, ltswell from medika_clinica

Subject	Visno	Visrpt	Visdate	Dayno	Joint	Rtpain	Rtswell	Ltpain	Ltswell
ANA101	0		2/1/1999 00:00:00	-1	Knee	0	0	0	
ANA101	0		2/1/1999 00:00:00	-1	Ankle	0	0	0	
ANA101	0		2/1/1999 00:00:00	-1	Hip	1	1	3	
ANA101	0		2/1/1999 00:00:00	-1	Wrist	0	0	0	
ANA101	0		2/1/1999 00:00:00	-1	Elbow	0	0	1	
ANA101	0		2/1/1999 00:00:00	-1	Shoulder	2	0	2	
ANA101	1		3/1/1999 00:00:00	0	Shoulder	1	1	1	
ANA101	1		3/1/1999 00:00:00	0	Elbow	0	1	0	
ANA101	1		3/1/1999 00:00:00	0	Wrist	0	0	0	
ANA101	1		3/1/1999 00:00:00	0	Hip	2	3	2	
ANA101	1		3/1/1999 00:00:00	0	Knee	3	0	1	
ANA101	1		3/1/1999 00:00:00	0	Ankle	2	1	0	
ANA101	2	1	4/1/1999 00:00:00	30	Shoulder	0	1	0	
ANA101	2	1	4/1/1999 00:00:00	30	Elbow	1	1	3	
ANA101	2	1	4/1/1999 00:00:00	30	Wrist	1	1	2	
ANA101	2	1	4/1/1999 00:00:00	30	Hip	0	1	0	
ANA101	2	1	4/1/1999 00:00:00	30	Knee	1	3	2	
ANA101	2	1	4/1/1999 00:00:00	30	Ankle	0	1	0	
ANA101	2	2	5/1/1999 00:00:00	60	Shoulder	0	1	2	

In this example, subjects ANA101 has multiple records stored for each visit (where VISNO = 0 and the repeating block combination of VISNO and VISRPT = 2/1 and 2/2). Each record stores data about the level of pain or swelling in a single joint at this visit.

Subject Enrollment (Type 5)

A Type 5 panel stores subject enrollment data. Design automatically includes all context items in the enrollment panel. You can optionally add other items.

For information on how to create an enrollment panel, see "Enrollment panel" on page 207.

How do panel types relate to repeating items?

Panel Types 0, 2, and 4 allow *repeating items,* items for which multiple values can be entered. A panel can contain either repeating items or nonrepeating items, but not both.

For more information, see "Repeating items" on page 483.

Context panels and context items

What are context panels and context items?

A *context panel* is a special panel containing *context items* that are associated with each record in a clinical data table defined by a panel of Types 1, 2, 3, 4, or 5.

The context panel is always named CONTEXT. There is one context panel for each protocol. The context panel requires a corresponding context page section that appears once on each study page, if that study page contains page sections based on panel types 1, 2, 3, or 4. The context panel is automatically set to Type 4. The type cannot be changed.

Certain context items, called *special context items*, are used as keys (either alone, in combination with each other, or in combination with items in the non-context panels) to uniquely identify a record for display in Enter. Before you install the context panel, you must specify three special context items used for the display of data in Enter:

- Subject Item — A subject-related item
- Block key item — A visit-related item
- Page key item — A page-related item

Optionally, you can specify the following special context items:

- Block repeat key item — A visit-related item
- Page repeat key item — A page-related item
- Error Log item — An item which holds an additional identifier for records in the Error Log.

For more information on the use of special context items and other panel items as keys, see "Special context items" on page 203, and "Panel keys" on page 210.

For more information on page sections, study books, and study pages, see Chapter 22 and Chapter Chapter 18.

For a description of the panel types, see "Panel types" on page 193.

Types of context items

There are four types of context items:

- Subject-related — Items that identify the subject, such as a subject identifier
- Visit-related — Items that identify the visit, such as a visit number or a date
- Page-related — Items that identify a study page, such as a page name or page number
- Other — Items not related to the subject, the visit, or the page, such as the investigator name

How do context items get initial values?

Any subject-related context items, including the subject item, are initialized based on the values supplied when the subject is enrolled.

Note: Data for subject-related context items is enterable only in the enrollment panel or by batch-loading. Subject-related context items are read-only in Enter.

You must initialize the block key value and the page key value when you create the block and the study page in the study book. You can also initialize the block repeat key value and the page repeat key value, if you identified these as special context items before you installed the panel.

Optionally, you can initialize values for any other visit-related context items or page-related context items by modifying the block values or page values in the study book.

For information on subject enrollment, see "Enrollment panel" on page 207.

For more information on panel types, see "Panel types" on page 193.

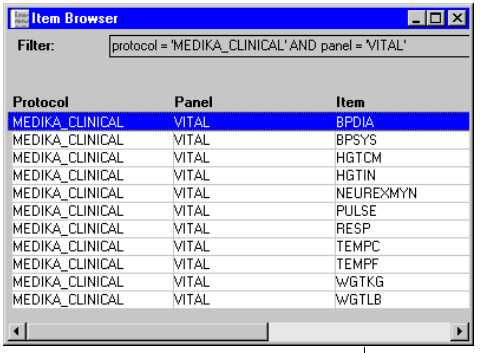
For information on initializing values in the study book for block-related context items and page-related context items, see Chapter 18.

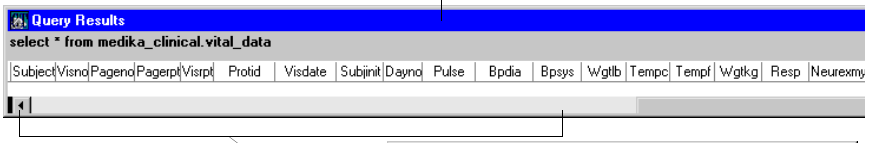
How do context panels relate to data tables?

Unlike other panels, the context panel cannot be deleted, and it has no data tables of its own. When you install panels of Type 1 through Type 5 in the protocol, all items in the context panel are attached automatically as columns in the clinical data tables for those other panels.

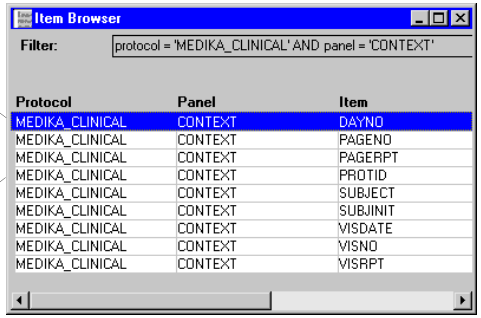
The following figure shows how the items in a context panel appear in the clinical data tables for all the other panels in the protocol:

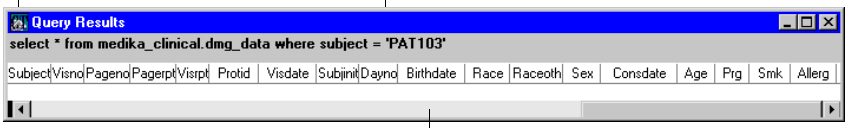
Items from the VITAL panel are included as columns only in the three VITAL clinical data tables (update, data, and audit).



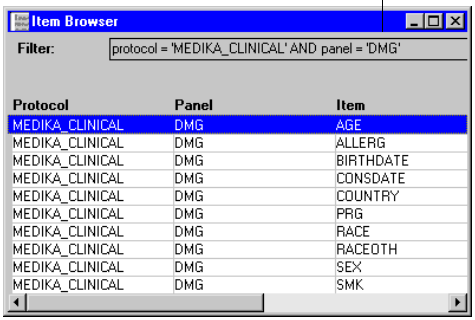


Items from the CONTEXT panel are included with all clinical data tables.





Items from the DMG panel are included as columns only in the three DMG clinical data tables (update, data, and audit).



In this example, the items from the CONTEXT panel are prepended to the VITAL_UPDATE and DMG_UPDATE tables.

Note: System items are also prepended to clinical data tables when they are created at panel installation. For more information on system items, see "List of system items" on page 222.

Special context items

What are special context items?

Special context items are items defined in the context panel whose values are used by the Clintrial software to uniquely identify a subject, visit, or study page. The special context items are:

- Subject item
- Block key item
- Page key item
- Block repeat key item
- Page repeat key item
- Error Log item

Before you install other panels, you can optionally specify additional key items to combine with the special context items to uniquely identify individual records for display in Enter.

Note: You cannot use an item of data type DATE or DATETIME as a subject item, block key item, block repeat key item, page key item, or page repeat key item.

The following sections describe the special context items.

What is a subject item?

A *subject item* is a subject-related context item used in enrollment to create a unique identifier for the subject. The Clintrial software associates the subject item with the system item SUBJECT_ID, for which the Clintrial software generates a value. The SUBJECT_ID is the key for identifying data by subject in the clinical data tables. The designer must specify the subject item as a protocol attribute for a clinical data protocol.

Typically, the subject item is the subject ID (the SUBJECT item in the MEDIKA_CLINICAL protocol) in the context panel, or a derived item consisting of several items in the enrollment panel.

Note: The subject item itself is not the key value for displaying subject data, because the subject item value may change during the course of your clinical study. The SUBJECT_ID system item is the actual key, and remains associated with the subject data even if the subject item value changes.

For information on the Derived item attribute, see "Derived" on page 232.

Why create a subject item as a derived item?

You may want to create the subject item as a derived item. For example, you may want your subject item to be the concatenation of a patient identifier and a site identifier. To create a subject item as a derived item, do the following:

1. In the context panel, create the context items, including a subject-related item for use as the subject item. Check the Derived attribute for this item, but do not attach a derivation to the context panel.
2. Install the context panel.
3. In the enrollment panel, create any additional items, and attach the derivation that calculates the value of the derived item.
4. Install the enrollment panel.

For information on the enrollment panel, see "Enrollment panel" on page 207.

For information on data-entry processing procedures, see Chapter 16 and the *Reference Guide*.

For more information on panel types, see "Panel types" on page 193.

What is a block key item?

A *block key item* is a visit-related context item that is a key for identifying data by block. For a nonrepeating block, the value of the block key item uniquely identifies the block; for a repeating block, the values of the block key item and the block repeat key item uniquely identify the block. Typically, the block key item is the context item for identifying the visit.

For example, the MEDIKA study book collects painful or swollen joint information in each visit (including any unscheduled visit). All data is stored in the clinical data tables defined by the JNTASM panel. In the MEDIKA study

book the VISNO item in the context panel is designated as the block key item. When the data-entry operator selects a subject in the Joint Summary study page in the Day 0 visit, Enter displays the joint summary information only for that visit and that subject, if the table stores such information.

What is a block repeat key item?

A *block repeat key item* is a visit-related context item that is the key, with the block key item, for uniquely identifying data in a repeating block.

For example, assume that a subject is expected to return for multiple visits with exactly the same set of study pages. You can set up the study book to contain repeating blocks, either as a set of repeats whose number is predefined by values supplied when you create the repeating block in the study book, or as an undefined set of repeats whose number can be defined by values supplied in Enter, based on the actual visits recorded in the CRF. In either case, you can also set a maximum number of allowable repeats.

For example, the MEDIKA study book contains two repeating blocks, for Day 30 and Day 60, and allows for additional unscheduled repeating blocks that the data-entry operator can create.

What is a page key item?

A *page key item* is a page-related context item that is a key for identifying data by study page within a block. For a nonrepeating study page, the value of the page key item uniquely identifies the page within the block; for a repeating page, the values of the page key item and the page repeat key item uniquely identify the study page within the block. Typically, the page key item is the context item for identifying the page number.

What is a page repeat key item?

A *page repeat key item* is a page-related context item that is the key, with the page key item, for uniquely identifying data in a repeating page within a block.

For example, the MEDIKA study book includes a repeating page for vital signs data. This allows for the expected entry of a single set of vital signs at each visit. However, the use of a repeating page also allows for additional sets of unscheduled vital signs data in the same visit, if necessary.

The block key would not be sufficient to distinguish between the vital signs data entered in a repeating page at the two different times in the same visit. The combination of the block key item, the page key item, and the page key repeat item ensures a unique key for the vital signs data collected multiple times in one visit.

Example

For example, suppose that the study designer knows that a subject must return for four consecutive monthly visits that include exactly the same study pages. The study designer could create a repeating block with four block repeat key item values: A, B, C, and D. But, anticipating possible unscheduled visits, the study designer creates the block to allow for additional block repeat key item values to be determined as needed in Enter. In Enter, these could be added as needed, perhaps as A-1, B-1, B-2, and so forth.

How are block and page keys specified?

To specify a block key item, block repeat key item, page key item, and page repeat key item from the Panel Browser do the following:

1. Create and save any items that you require for the context panel, including visit-related items for the block key item and block repeat key item, and page-related items for the page key item and the page repeat key item.
2. In the Panel Browser, click on the context panel.
3. From the **Panel** menu, select **Key Items** >> **Special Context Items**.

   The Define Special Context Items dialog box opens.
4. Select the following:
   - A subject-related item as the Subject Item (required)
   - An item that stores a unique identifier for records in the Error Log for the Error Log Item (optional)
   - A visit-related item as the Block Key Item (required)
   - Another visit-related item as the Block Repeat Key Item (if you are planning for block repeats)
   - A page-related item as the Page Key Item (required)
   - Another page-related item as the Page Repeat Key Item (if you are planning for page repeats)

Note: If you do not specify the subject item, block key item, or page key item before you install the context panel, you are prompted to do so when you install the panel. At that time, you can also specify the block repeat key item and page repeat key item.

5. Install the context panel.
6. Create the context page section layout.

When you create the blocks and study pages in the study book, you specify values for the block key item and the page key item.

Optionally, when you create blocks and study pages in the study book, if repeating blocks or study pages have been set up in Design, you can specify that a block or a study page can have repeats. When you specify a repeating block or study page, you can also specify multiple block repeat values for a repeating block and page repeat values for a repeating page. These values differentiate repeating blocks or repeating study pages in Enter. Or, you can allow the Enter user to specify these values for blocks and study pages that you have defined as repeats.

For information on specifying the block key value and the page key value, as well as block repeat key values and page repeat key values in the study book, see "Block values and page values" on page 407.

For information on specifying the block repeat key values and page repeat key values in Enter, see *Enter, Resolve, and Retrieve*.

Enrollment panel

What is an enrollment panel?

An *enrollment panel* is a panel that stores enrollment data that uniquely identifies each subject in a study. An enrollment panel contains all context items (including one item that is designated as the subject item), and optionally, other subject-related items defined for the panel.

Enter requires subject enrollment for access to clinical data tables based on panels of types 1, 2, 3 and 4. Therefore, a clinical data protocol must have an enrollment panel.

Steps for creating and installing an enrollment panel

When you create and install an enrollment panel, Design automatically creates enrollment panel items based on any context items for this protocol. Therefore, you must create and install the context panel first. Then, do the following:

1. Create the enrollment panel.

   An enrollment panel is a Type 5 panel.

2. Create any additional items that you require for the enrollment panel.

   You may want to create items to store some information in the enrollment panel, rather than in the study book's panels. For example, you could use items specific only to the enrollment panel to derive a unique subject item from a subject number and a site identifier.

   Note: You should create the subject item in the context panel as a required item. Likewise, create any other items used to create a derived subject item as required items (in the context panel or the enrollment panel).

3. Install the panel.

   Design automatically creates a single page section, study book, block, and study page based only on the enrollment panel. The study book is used in Enter to enroll subjects. You cannot add additional blocks or study pages. However, you can delete the study book and recreate it with the same name as the enrollment panel.

   The enrollment data entered in Enter is stored as one record in an enrollment panel. Data for the subject-related context items is displayed in the context section of any study page in which data can be entered for this subject.

4. Create the page section layout.

   Note: Although the tables for the enrollment panel contain all the context items, the page section layout contains only the subject-related context items, and any additional items you create in the enrollment panel.

   Note: Data for subject-related context items is enterable only in the enrollment panel or by batch-loading. Subject-related context items are read-only in Enter.

5. From the Study Book Browser, click on any study book, select **Enrollment Panel**, and specify the enrollment panel.

Note: Although all the context items are stored in the enrollment panel table, you cannot use the visit-related and page-related items in the enrollment panel itself. Subject-related context items are included automatically in the enrollment panel. You can add additional items to store enrollment data.

Example of derived subject

For example, when you enroll a subject in the MEDIKA_CLINICAL protocol's ENROLL study book, the result is a unique subject identifier consisting of a concatenation of the site identifier and the patient number. Create such a subject identifier as follows:

1. Create a context panel containing the required context items, including subject-related items (in the MEDIKA study book, SUBJECT for the subject item, and SUBJINIT).
2. Check the proposed subject item as derived, but do not attach a derivation to the context panel.
3. Specify SUBJECT as Subject Item.
4. Install the context panel.
5. Create an enrollment panel (ENROLL, in MEDIKA_CLINICAL).
6. Create items specific to the enrollment panel to contain the site identifier (CENTER) and a site-specific patient identifier (PAT_NUM).

   Note: Instead of creating CENTER and PAT_NUM in the enrollment panel, you could create them in the context panel, as subject-related context items. If you create them in the enrollment panel, the data is stored only in the clinical data table for the enrollment panel; if you create them in the context panel, the data is stored in the clinical data tables for all panels.
7. Create a derivation attached to the enrollment panel that derives the value or the subject item (SUBJECT) as a concatenation of the values for the center and the patient identifier (CENTER and PAT_NUM).
8. Install the enrollment panel.

For more information

See "Context panels and context items" on page 199 for a description of context panels.

See "Panel types" on page 193 for a description of the panel types.

See "Creating and installing a panel and its items" on page 221 for a description of how to create and install panels.

See Chapter 22 for a description of page sections, Chapter 17 for a description of page section layouts, and Chapter 18 for a description of study books, blocks, and study pages.

For more information about data-entry processing procedures, see Chapter 16 of this guide and the *Reference Guide*.

View panels

What is a view panel?

A *view panel* is view of the data in a panel that is part of a view protocol. The view protocol checkpoint date, if any, is used for all panels in the protocol. The view restriction, if any, is used as the default value of the restriction for each panel.

You can set a view restriction in the view panel that is different from the restriction clause set in the view protocol. If specified in the view panel, the restriction clause overrides the view protocol's restriction clause for that view panel. However, you cannot override the view protocol's checkpoint date for individual view panels.

After you create a view protocol, you can copy one or more view panels from the base protocol into the view protocol. You must explicitly copy a view panel for each required table view, unless you choose to copy all panels during view protocol creation. You can modify view panels and define a panel-specific restriction if you decide that you do not want the same restriction applied to all the panels in the view protocol (or if the view protocol does not have a restriction clause and you want one for a particular panel or panels).

You can copy a view panel for every panel defined by the base protocol for which clinical data tables have been created (therefore, not for the context panel).

For information on view protocols, see "View protocols" on page 166.

For information on SQL view restriction clauses, see the *Reference Guide*.

Panel keys

What are panel keys?

Panel keys are items that, in combination with special context items (subject item, block key item, block repeat key item, page key item and page repeat key item) uniquely identify each record in the clinical data table.

The special context items are not always sufficient to uniquely identify each record in the clinical data table. Using Design, you can identify additional items in each panel to create a unique key.

The following table summarizes the unique key requirements for each panel type. From the Panel Browser, select a panel and define keys by specifying items.

Panel type:	**Unique key consists of:**
Non-patient data (Type 0)	Any item or items that the designer specifies as the panel keys and that uniquely identify the record for the panel
1 Record per Patient (Type 1)	SUBJECT_ID, a Clintrial software-generated system item that corresponds to the unique subject identifier specified by the designer as the Subject Item context item
>1 Record per Patient (Type 2)	SUBJECT_ID and value of the page key item, (and page repeat key item for a repeating page) and any designer-specified item or items in the panel that combine to uniquely distinguish individual repeating records
1 Record per Patient Visit (Type 3)	SUBJECT_ID and value of the block key item (and block repeat key item, for a repeating block), that combine to uniquely identify a single nonrepeating record in a subject visit
>1 Record per Patient Visit (Type 4)	SUBJECT_ID, value of the block key item (and block repeat key item, for a repeating block), page key item, (and page repeat key item for a repeating page), and any designer-specified item or items in the panel that combine to uniquely identify individual repeating records

Caution: It is strongly recommended that you specify a panel key for a Type 0, 2, or 4 panel. If you do not, you may encounter merge problems in Manage.

All panel key items must have the Required attribute checked.

Panels with master-detail keys

What is a master-detail relationship?

A *master-detail relationship* is a relationship between two page sections on a study page, in which each record in one page section (the master page section) can have zero or more associated records in the other page section (the detail page section). During data entry, the displayed records in the detail page section are associated with the selected record in the master page section.

If you have multiple records in a master page section, you can access different sets of records in the detail page section.

There are two types of master-detail relationships, as follows:

- Cross-panel

  In a cross-panel master-detail relationship, the detail page section is based on a different panel than the master page section. You set up a cross-panel master-detail relationship by specifying a master key item in one panel, and a detail key item in another panel.

  For example, in the MEDIKA_CLINICAL protocol, a master-detail relationship exists between records in the PHYEXM and NEUROL panels. In Enter, you can enter data in the subject's record in the PHYEXM panel. If the subject requires a neurological exam, one or more detail records in the NEUROL panel are associated with a master record for the subject in the PHYEXM panel.

- Within-panel

  In a within-panel master-detail relationship, the detail page section is based on the same panel as the master page section. You set up a within-panel master-detail relationship by placing on a page template two separate page sections that are based on the same panel. Master key items and detail key items are not needed for within-panel master-detail relationships.

  For example, in the MEDIKA_CLINICAL protocol, a master-detail relationship exists between records in the ADV (master) and ADVCOMM (detail) page sections, both based on the ADVERSE panel. In Enter, this relationship enables you to add additional details for each adverse even that is recorded.

  Note: A detail page section in a within-panel relationship cannot serve as the master page section in another within-panel relationship.

For more information on creating a cross-panel master-detail relationship among records, see the following sections and "Creating page templates with master and detail page sections" on page 370.

For more information on creating a within-panel master-detail relationship, see "Repeating items in master-detail relationships" on page 381.

What is a master key item?

A *master key item* is an item in the master record that is the key for uniquely identifying the detail records associated with the master record. When a user enters a value for the master key item, that value is propagated to the associated detail records.

What is a detail key item?

A *detail key item* is an item in the detail record that is the key for uniquely identifying the master record associated with the detail record. Users do not enter the value for the detail key item; the value is propagated from the master key item in the associated master record.

Requirements for cross-panel master-detail key

For a cross-panel master-detail relationship, master key items and detail key items must meet the following requirements:

- Both items must be of the data type TEXT or FIXED.
- Both items must be of the *same* data type (that is, both TEXT or both FIXED).
- The detail key item must be required.
- Neither item can be a context item.
- Neither item can be defined as a subset key item.

Note: You can set up the master key item so that Enter automatically generates a unique value for each record. To do this, specify the data type of the master key item as FIXED, and clear the master page section's Enterable attribute for the master key item.

Requirements for panel type

You can set up a cross-panel master-detail relationship involving panel types 1-4, but not Type 0. (This limitation is necessary because all page sections on a Type 0 page must access the same panel. However, you can use a Type 0 panel for a within-panel master-detail relationship, since all page sections access the same panel.)

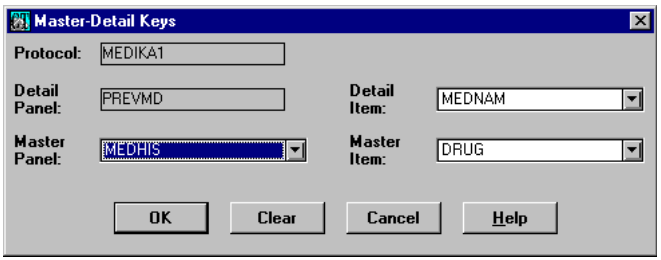
How to define a master-detail relationship

For a page template to contain page sections that are in a cross-panel master-detail relationship, you must first define this relationship in the Panel Browser. You can then place a master page section and one or more detail page sections on a page template.

To define a master-detail relationship across panels:

1. Ensure that:
   - The data dictionary component of the master panel is installed.
   - The data tables component of the master panel is not installed.
   - The detail panel is not installed.
   - All key items (both master and detail) are required.
2. In the Panel Browser, select the panel you want to be the detail panel, and then from the **Panel** menu, select **Key Items >> Master Detail Keys**.

   The Master-Detail Keys dialog box opens:



3. From the drop-down lists, select the required information for the following fields:
   - Master Panel — Name of the panel you want to be the master. This panel must be different from the current panel displayed in the Detail Panel field.
   - Master Item — Name of an item belonging to the master panel. In this example, this item is DRUG, the value of which will be propagated to the field associated with the MEDNAM (detail) key.

- Detail Item — Name of an item belonging to the detail panel. In this example, this item is MEDNAM. It has the same range of values as the master item.

4. Click **OK** to save the panel keys.

You can now create a master page section and detail page sections, and place them on a page template.

In the example from the Medika sample study, the master panel is named MEDHIST and the detail panel is named PREVMD. The master key is the item on MEDHIST named DRUG, and the detail key is MEDNAM.

Master-detail page sections and page templates

After you create and install the panel or panels that you plan to use in a master-detail relationship, do the following:

1. Create a master page section and a detail page section.

   For information on creating master page sections, detail page sections, and subset page sections, see "Master page sections, detail page sections, and subset page sections" on page 485.

2. Place the master page section and detail page section on a page template.

   For information on creating page templates containing master and detail page sections, see "Creating page templates with master and detail page sections" on page 370 and "Page template attributes" on page 383.

How to modify or delete master-detail panels

You can modify or delete a master panel or detail panel only as follows:

- For any panel referenced as a master panel, you cannot delete this panel or any master key item it contains.
- For any panel referenced as a master panel, you cannot modify the data type of any master key item.
- For any panel referenced as a detail panel, you cannot delete the detail key item or change the data type of that item. If you delete the detail panel itself, the master-detail relationship is deleted.
- You must deinstall the detail panel(s) before you can deinstall the master panel.
- A single master panel may have multiple detail panels. You can associate each detail panel with the master panel using a different key item.

Panels with subset key items

What is a subset key item?

A *subset key item* is an item that is the key for uniquely identifying the records in a subset page section. The value of the subset key item is determined when the subset page section is placed on a page template; this value cannot be modified during data entry or editing.

You can use a subset key item to place multiple page sections, based on a single panel containing the subset key, on the same page template. These page sections, which can be repeating or nonrepeating, each contain distinct rows (subsets) of data.

A subset key is a panel item that must meet the following requirements:

- It must be of the data type TEXT or FIXED.
- It must be required.
- It cannot be a context item.
- It cannot already be defined as a cross-panel master key.
- It cannot have a sequence defined for it.

Example of subset key item usage

For example, suppose you want to set up data entry for a series of lab test results based on the same set of repeating items. You could group those results on a single study page according to the type of test. Using the item specifying test type as a subset key in the appropriate panel (LABS in the Medika Sample Study), you can create a series of repeating page sections, one for each test type (for example, blood chemistry, hematology, and urinalysis) based on this panel. You can now place all these page sections on the same page template by specifying unique subset key values for each page section in the Page Template editor.

For more information on setting up study pages with subset keys, see Chapter 16.

How to specify an item as a subset key

To specify an item as a subset key:

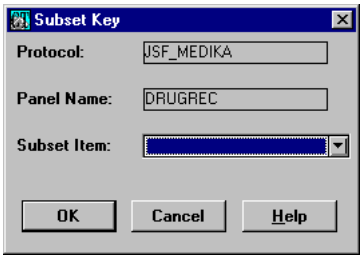
1. Create the panel and associated items you want to use for subset page sections.

   This step includes creating the item you want to specify as the subset key item.

   Note: You can change or remove the subset key when the panel is uninstalled or deinstalled. If you remove the subset key, the subset key values must be removed from the corresponding page sections. Changing or removing the subset key invalidates the page section layout.

2. In the Panel Browser, select the panel for which you want to specify a subset key, and then from the **Panel** menu, select **Key Items >> Subset Key**.

   The Subset Key dialog box opens:



3. From the Subset Item drop-down list, select the subset key for this panel, and then click **OK**.

Note: You can optionally display the subset key item in the page section layout editor. If the item appears on the page section, it is marked as not modifiable.

Note: You cannot delete an item that has been chosen as the subset key.

For information on creating subset page sections, see "Master page sections, detail page sections, and subset page sections" on page 485.

For information on placing subset page sections on page templates, see "Page template attributes" on page 383.

Grouping and sorting items in a panel

You can specify grouping items for Type 0 panels, and sorting items for Type 0, 2, and 4 panels. You should always specify a sort item and a grouping item.

Note: If neither type of item is specified, Enter sorts according to CT_RECID. In this scenario, however, the client sort order and server sort order differs, leading to possible confusion during verification. Clintrial 4.3 sorts in the sequence shown in the Character column (as defined by the regional settings on the client machine), while on the server, Oracle sorts using the ASCII code for each character. To avoid these discrepancies, always specify a sorting and grouping key.

Note: Sorting uses uppercase values of the items.

What is a sorting item?

A *sorting item* is an item used to sort records in a page section based on a Type 0, Type 2, or Type 4 panel. A sorting item can be used to sort records in ascending or descending order.

What is a grouping item?

A *grouping item* is a sorting item that also functions as a key for grouping multiple records into observations in a page section based on a Type 0 panel.

For more information on observations, see *Enter, Resolve, and Retrieve*.

Requirements for grouping items

Grouping items must meet the following requirements:

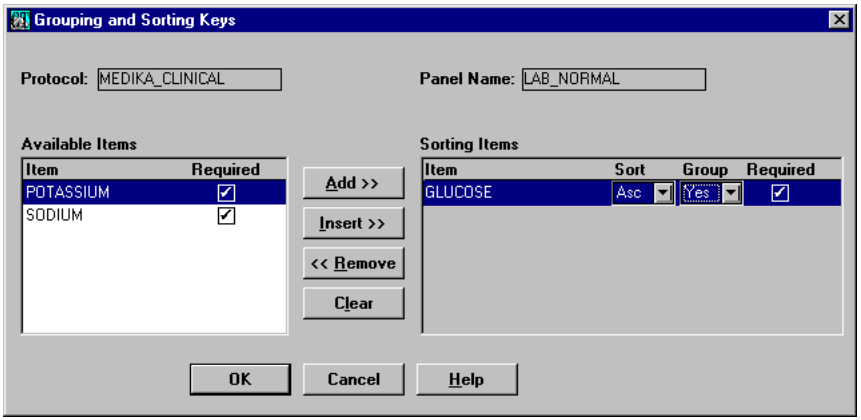
- They must be required items.
- They cannot be derived items.
- They can only be of the data type TEXT or FIXED.
- If grouping items are specified for a panel, the values of these items must never be updated after the record has passed screening.

How to specify grouping and sorting items

To specify grouping and sorting items:

1. From the **Panel** menu, select **Key Items >>Grouping and Sorting**.

   The Grouping and Sorting Keys dialog box opens:



2. Do one of the following:
   - To add an item from the Available Items list to the Sorting Items list, select the item and then click **Add >>** or **Insert >>**.
   - To remove an item from the Sorting Items list, select the item and then click **<< Remove**.
   - To remove all the items from the Sorting Items list, click **Clear**.
3. To specify sort order for a sorting item, from the **Sort** drop-down list, select **Asc** (ascending order) or **Desc** (descending order).
4. To specify that you want the item to be a grouping item, from the **Group** drop-down list, select **Yes**.

   Note: This option is displayed only when the items (as in this example) belong to a Type 0 (non-patient data) panel. Type 2 and Type 4 panels always use the default grouping item — subject item, block repeat item, page repeat item, and subset key item (if a repeating block, repeating page, or subset page section is specified).
5. Click **OK** to save.

The items sort in the order displayed. Grouping items must sort before nongrouping items. Therefore, grouping items must be higher in the list than nongrouping items.

For more information on grouping and sorting items, see "List of read-only attributes" on page 235.

Example of grouping key use for observations

An observation is a group of records, associated with a study page field, that functions as a single object during data management tasks. You can use grouping keys to enable the Enter user to add observations to fields in a repeating section of a Type 0 (non-patient data) study book. To set up a field so that observations can be added to it, simply create a required item for that field and then specify that item to be a grouping key, as described in the previous procedure.

For information on creating Type 0 study books, see "Study books for non-patient data" on page 406.

For information on observations, see *Enter, Resolve, and Retrieve*.

Checking data in panels and items

List of objects and attributes that control data entry

As the designer, you can use the Clintrial software to automate, control, or limit data entry. The following table lists Clintrial software objects or object attributes that can be associated with panels or items to calculate a value or to limit valid values:

Clintrial software object or attribute:	Associated with:	Description:	For more information:
Derivation (Clintrial software object)	Panel	A PL/SQL statement (or series of statements) that calculates the value of an item in a panel and assigns that value to the item in the database, or that calculates a temporary variable.	See Chapter 11.
Rule (Clintrial software object)	Panel	A part of a PL/SQL statement that evaluates to TRUE or FALSE.	See Chapter 11.

Clintrial software object or attribute:	Associated with:	Description:	For more information:
Coding target (Clintrial software object)	Item	A set of items in a clinical data protocol that correspond to items in the coding thesaurus protocol that is used to code data entered in the clinical data protocol.	See Chapter 14.
Codelist (Clintrial software object)	Item	A set of codes linked to a corresponding set of values.	See Chapter 12.
Min or Max value (item attribute)	Item	A minimum or maximum value that can be entered for the item.	See "Item attributes" on page 231.

Creating and installing a panel and its items

Steps for creating a panel and its items

To create a panel and its items:

1. Decide which group of items to include in the panel, based on the CRF page and other requirements for your protocol (for example, items required for use with a coding thesaurus protocol).
2. Specify the panel attributes and create the panel.
3. Add items to the panel, either by creating new items or by copying existing items. To create an item:
   a. Decide which piece of information to represent by an item.
   b. Create a codelist, if any, to be used by the item.
   c. Create the item, either by copying an item if a suitable item is in any protocols in the searchlist, or by creating a new item and specifying its attributes.
4. Add any necessary derivations and rules to the panels.
5. Set up coding targets (if the panel contains one or more coded items).

6. Specify key items for the panel.

   These include, optionally, special context items, panel keys, grouping and sorting items, subset keys and master-detail keys.

For information on installing panels and marking panels for revision, see the following section and "What is marking a panel for revision?" on page 224.

For information on panel and item attributes, see "Panel attributes" on page 226 and "Item attributes" on page 231.

For information on coding thesauruses, see Chapter 14.

What is panel installation?

After you create the panel, you must install it to populate data dictionary tables and create clinical data tables. First, install the context panel; then, install the other panels.

When you install a panel:

- Design stores the metadata (panel and item attributes) in data dictionary tables. The items are now available for copying from other protocols that contain this protocol in their searchlists.
- You can optionally choose to create clinical data tables to store clinical data. You install the context panel only in the data dictionary tables. When you create clinical data tables at the installation of other panels, the context items are included in all the clinical data tables that have subject-related data (that is, not in clinical data tables based on Type 0 panels).

You can perform the two steps simultaneously or you can create the data dictionary information first, and defer creation of the clinical data tables until later. If you are creating panels in a protocol with the Dictionary attribute checked, you cannot create clinical data tables. When you install clinical data tables for panels in a view protocol, no tables are created. Instead, views are created on the base panels.

List of system items

When you install a panel, the Clintrial software creates system items to identify and manage records. The system items are prepended to the three installed clinical data tables (Update, Data and Audit) for each panel. The following is a list of system items:

- MERGE_DATETIME — Date and time the record was last modified or deleted from the table
- STATUS — Numeric code indicating the status of the record
- ENTRY_ID — User account that entered or last modified the record
- ENTRY_DATETIME — Date and time the record was created or placed in the table (depending on the table)
- CT_RECID — Unique identifier that is automatically assigned to the record
- DB_ID — Unique identifier of the Clintrial software database instance that owns this record
- SUBJECT_ID — Unique identifier of the subject
- CTS$REASON — Reason that the record was changed or deleted

For a detailed description of these system items by table, see the *Reference Guide*.

Modifying a panel and its items

How are a panel and its items modified?

You can modify the contents of a panel, such as items, rules, derivations, and coding targets. The actions you take to modify the panel depend on the state of the panel, as summarized in the following table:

Panel state:	Steps to modify:	Notes:
Created, not installed	1. Modify items, rules, derivations, and coding targets. 2. Install.	Cannot modify the object names. Limited modifications of items are allowed.
Installed	1. Mark for revision (if not already marked). 2. Modify rules, derivations, coding targets, panel keys, and limited items. 3. Implement (or cancel) changes.	Cannot modify the object names or item data type.

Panel state:	Steps to modify:	Notes:
Deinstalled	1. Modify items, rules, derivations, and coding targets. 2. Reinstall.	No restriction on modifications, deinstallation removes the clinical data, and, optionally, the metadata. Cannot modify the object names.
Deleted	None.	Deletion removes all clinical data and metadata.

The following sections describe in more detail the issues related to panel modification.

What is marking a panel for revision?

By *marking a panel for revision*, you indicate that you intend to make changes in the panel. However, the changes will not actually take effect until you implement the revisions. When a panel is marked for revision, tasks such as data entry and retrieval can continue with the unchanged panel. When you implement the panel revisions, the changed panel replaces the previous version of the panel.

How can items, derivations, or rules be modified?

When you have marked a panel for revision, you can modify it in the following ways:

- Add new items.
- Modify items (with some restrictions).
- Add, change, or delete rules, derivations, and coding targets.
- Change the order of the items in the panel.

You must deinstall the panel to make any of the following modifications:

- Modify additional attributes of items without restrictions.
- Delete items.

- Modify keys (panel, subset, grouping, sorting, master, and detail).

How are revisions implemented?

After you revise the panel contents (items, derivations, rules, coding targets, and panel keys), you implement the pending revisions (or cancel them, as described in the following section). During the implementation process, no other users can use the panel.

When you implement revisions to a panel:

- Changes to the clinical data tables are carried out.
- The validation procedure is rebuilt.
- The coding procedure is rebuilt.

When all revisions have been implemented, the panels are no longer considered to be marked for revision.

You cannot implement changes to the context panel if any other panels are marked for revision. When you implement revisions to the context panel, the Clintrial software marks all Type 1 through Type 5 panels for revision and automatically implements the changes.

How are pending revisions canceled?

You can cancel pending revisions if you have not yet implemented them.

What is panel deinstallation?

Deinstalling a panel reverses the installation process. The deinstallation process has two stages:

- The first stage drops all clinical data tables (including data in the tables) associated with the panel.
- The second stage, which is optional, changes the item metadata definitions in the data dictionary tables to “uninstalled”.

You can perform the first stage of panel deinstallation without performing the second stage. For example, you might want to delete the clinical data tables without affecting the metadata in the data dictionary tables, so that items and panels are still available to be copied.

After you perform the first stage of panel deinstallation, the panel can no longer be used for data collection because its clinical data tables no longer exist.

You cannot deinstall a panel in the base protocol for which there is an associated view. You must drop the view before proceeding with the base panel deinstallation. The Clintrial software tells you if it cannot deinstall the panel because of a view.

Caution: If there is data in the clinical data tables, the Clintrial software warns you that if you drop the tables for the panel, the data will be lost.

How is a panel deleted?

You can delete a deinstalled or uninstalled panel from the current protocol. When you delete a panel:

- Data in the clinical data tables associated with the panel is deleted.
- Clinical data tables associated with the panel are dropped.
- Item metadata definitions are removed from the data dictionary tables if there are no other copies of the item in other panels.
- Rules and derivations attached to the panel are deleted.

You cannot delete a panel in the base protocol for which there is an associated view. You must delete the view before deleting the base panel. the Clintrial software tells you if it cannot delete the panel because of a view.

Panel attributes

With the Panel Browser open, when you select **Panel >> Modify**, the Clintrial software displays the following panel attributes:

Protocol

Name of the protocol containing the panel.

Panel Name

Name of the panel.

SAS Name

Name for this panel when data is sent to SAS through the SAS interface. This attribute must be no more than eight characters, and it must conform to SAS requirements. The default SAS name is the first eight characters of the panel name.

Panel Type

The panel type (0 through 5) that determines which context items are required, and whether one or multiple records are allowed for each set of context items. For more information see "Panel types" on page 193.

Audit Start

The Clintrial software allows for the auditing of data that is modified or deleted. This attribute sets a start point at which auditing begins for this panel. The start points are:

- Entry — Auditing begins after data has been entered interactively in a record. The record is in the update table.

  Note: Batch-loaded records cannot be audited until they pass screening; that is, until their record status becomes 1.

- Verification — Auditing begins after data has been reentered to check the accuracy of the entry, or, for batch-loaded records, after data has been screened. The record is in the update table.

  Note: No changes to data are stored during blind verification, so there is nothing to audit.

- Validation — Auditing begins after a record has been validated, regardless of whether the record passed or failed validation. The record is in the update table.
- Validity — Auditing begins after a record has been validated and has passed validation. The record is in the update table.
- Merge — Auditing begins after data has moved from the update table to the data table. The record is in the data table.

The default audit start value for the panel is set by the protocol. You can enter a different value for this panel with the Audit Start attribute.

Validation Priority

Specify a number to set a validation priority other than the default. By default, validation on multiple panels at the same time proceeds as follows:

- For interactive validation, the panels are validated in alphabetical order.
- For batch validation, the panels may or may not be validated in alphabetical order.

Use the Validation Priority attribute to specify that validation procedures process panels in an order other than the default order, so that you can process derivations or rules in that order. The Validation Priority attribute applies both to interactive validation and batch validation.

For example, a rule in one panel may look up the value of a derived item in a different panel. The panel with the derived item must be validated prior to the panel with the rule.

If you set the validation priority in one or more panels to 1, those panels validate in alphabetical order before the panels with no validation priority. If you set some panels to 1 and some to 2, the panels with a validation priority of 1 validate first, those with a value of 2 validate next, and those with no validation priority validate last.

Verify

Check to require reentry of data originally entered in the update table associated with this panel, to check the accuracy of the original entry.

Caution: If this attribute is changed after a panel and its page sections have been created, the page sections are marked invalid. Then you must open and save the page section layout for each page section to apply the changed attribute, or use the **Update** command.

Protected

Check to limit access rights to this panel. Access to a panel that has the Protected attribute set is limited. By default, a user with the following protocol access rights cannot exercise them in a protected panel:

- Enter — Merged and Unmerged
- Retrieve — Merged and Unmerged
- Manage — Coding, Global and Other

- Tools — SQL
- Resolve — Create, Propose, Manage and Produce.
- Lab Loader — Design and Transfer

By default, each of these access rights has the access level None if the panel is protected. You can override the default and allow these access rights in the same way that you allow access rights for users and usergroups for the protocol.

For a full description of each of these protocol access rights and levels of access, see **Chapter 4: Security**, in the section "Protocol access rights," in this manual.

Other protocol access rights, Enter - Enroll, and Retrieve - Library, work normally on both protected and non-protected panels.

In Admin, the Clintrial system administrator can specifically enable any of these access rights for a user of a protected panel.

Only panels to which a user has the proper rights will be displayed in dialogs.

Lab Loader Protected Panel access

In order to **create**, **edit**, **copy** and **delete maps**, the user must have **Full** protocol rights to the **Source protocol** and **Full Lab Loader Design** rights to protected panels in the Source protocol, as well as at least **Read** rights to the **Destination protocol** and at least **Read Lab Loader Design** rights to the protected panels.

A minimum of **Read** rights to the protocols are required to **show transfer maps**, if the map refers to protected panels, you need at least **Read Lab Loader Design** and **Transfer** rights to protected panels in the protocols.

If the panel in the Source protocol is protected, in order to actually **transfer data** the user must have at least **Read** rights to the **Source protocol** and at least **Read Lab Loader Design** rights to protected panels in the Source protocol. **Full** rights to the **Destination protocol** and **Full Lab Loader Transfer** rights to protected panels in the Destination protocol are also required.

This may be summarized in the following table:

Tasks	Protocol Rights	Panel Lab Loader Rights Required if Protected
Create, Edit, Copy or Delete Maps	Design: Full on Source Transfer: Read on Destination	Design: Full on Source Transfer: Read on Destination

Tasks	Protocol Rights	Panel Lab Loader Rights Required if Protected
Show Maps	Design: Read on Source	Design: Read on Source
	Transfer: Read on Destination	Transfer: Read on Destination
Transfer Data	Design: Read on Source	Design: Read on Source
	Transfer: Full on Destination	Transfer: Full on Destination

Manage Protected Panel access

Note: There is no restriction for Flags and Notes, since there are no access rights for Flags and Notes

Note: All logs and control files are not included. Some log files and control files can be saved as external files. You should protect them from the Windows level.

Resolve Protected Panel access

Note: Discrepancy count reports will not exclude data on protected panels since only counts, not actual data, can be seen. To help avoid confusion, the message "May include data for protected panels" will be put on any report run in a protocol that contains any protected panels.

Description

An optional description of the panel.

List of read-only attributes

When you select the **Show** command from the Panel Browser, the Clintrial software displays the previously described panel attributes as read-only, and also displays the following additional read-only attributes:

- Installed — Checked if the panel is already installed.
- Tables Created — Checked if clinical data tables currently exist.
- Marked for Revision —Checked if the panel is marked for revision.

  The changes do not take effect until the panel revision is implemented. Meanwhile, tasks such as data entry and retrieval can continue with the unchanged panel.

- Revising — Indicates if revisions to the panel are being implemented. During implementation of the revision, no one can use the panel.
- Modification Date — Date and time of the last modification to the panel.
- Subset Item — Name of the item specified as the subset key for subset page sections based on the panel. (Blank if no subset key is specified.)
- Modified By — Name of the user account that last modified the panel.
- View Object ID — A unique identifier used internally to identify the view restriction text.
- Modifiable Status — Modifiable if the panel can be modified, Not Modifiable if the panel cannot be modified.

In addition, the **Show** command displays the following information:

- Master-detail relationships
- User-defined panel keys
- Grouping and sorting items

Item attributes

With the Item Browser open, when you select the **Create** or **Modify** commands from the **Item** menu, the Clintrial software displays the following item attributes in the Create Item or Modify Item dialog boxes:

Protocol

Name of the protocol containing the panel that contains this item.

Panel

Name of the panel that contains this item.

Item Name

Name of the item.

SAS Name

Name of this item when data is sent to SAS through the SAS interface. This attribute must be no more than eight characters, and it must conform to SAS requirements. The default SAS name is the first eight characters of the item name.

Required

Check to require that a value for the item must be supplied (for example, by the data-entry operator, or as a derived item).

Note: A required item that is not displayed in a page section layout does not require a value. However, a required item that is batch-loaded does require a value, and will fail screening if a value is not supplied.

Derived

Check to specify that the value of the item is calculated from a derivation associated with the panel. You must attach a derivation for the item to the panel.

The subject item can be derived only in an enrollment panel (not in the context panel), and must also have a data-entry processing procedure defined for it in the page section layout for the enrollment panel.

By default, a page section does not display a derived item. However, you can add derived items to the page section. The derived items appear as read-only.

For more information on creating a derived subject item, see "Why create a subject item as a derived item?" on page 204.

Context Type

For a context item, select one of the following types:

- Subject-related context item
- Visit-related context item
- Page-related context item
- Other context item

Context type cannot be modified once the item has been created.

Note: You cannot use an item of data type DATE or DATETIME as a subject item, block key item, or page key item.

Data Type

The data type for values of this item. The data type can be TEXT, FIXED, FLOAT, DATE, or DATETIME. If the item is installed in the data dictionary tables, you cannot change this attribute.

TEXT items are limited to a length of 2000. FIXED items are limited to a length of 10. FLOAT items are limited to a length of 18 with at most 10 decimal places.

Note: In a Clintrial Unicode protocol, the setting of varchar2(2000) has a maximum length of 4000 bytes. In a language using Western characters (like English or French), where each character takes no more than 2 bytes, this sets the limit at 2000 characters. In Japanese, a character requires 3 bytes or more, so the limit could be as few as 1333 characters (4000 bytes divided by 3).

Units

A text description of the units of measurement for the item. This attribute controls coordination of nonhomogeneous data during data transfer.

DB Format

The format in which the Clintrial software stores values for the item in the clinical data tables. The user preferences DD_DBFMT_TEXT, DD_DBFMT_FIXED, and DD_DBFMT_FLOAT determine a default for this attribute, depending on the data type of the item. After installation of the item, you can increase the width of the DB format, but you cannot decrease the length or change it to conflict with the data type.

Min value

The minimum value that can be entered for the item. For example, you might set this value to 100 for the item WEIGHT if the Units attribute is pounds, and the clinical protocol requires that subjects weigh at least 100 pounds.

Max value

The maximum value that can be entered for the item. For example, you might set this value to 200 for the item WEIGHT if the Units attribute is pounds, and the clinical protocol requires that subjects weigh no more than 200 pounds. The Max value must be greater than the Min value.

Codelist

The name of a codelist associated with the item. A codelist encodes entered values. Only codes or values in the codelist can be entered. If you specify a codelist, you cannot specify a checklist or a coding thesaurus. The data type of the item and the codelist's Code Field Type attribute must be the same.

In an uninstalled panel, you can freely modify this item. In an installed panel marked for revision, the item's behavior depends on the setting of the protocol parameter DD_STRICT_CODE. If this parameter is set to Yes (the default), then you can only change the codelist to be a less restrictive subset. If DD_STRICT_CODE is set to No, you can freely modify the codelist. However, you must ensure that the new codelist is compatible with existing clinical data.

Checklist

The name of a codelist associated with the item. A checklist is a codelist that checks, but does not encode, values for the item. If you specify a checklist, you cannot specify a codelist or a coding thesaurus. The data type of the item and the codelist's Value Field Type attribute must be the same.

The Override attribute in the item's page section layout Design Attributes determines whether the data-entry operator is restricted to the checklist values, or can enter other values. For more information on the Override attribute, see "List of Attribute commands" on page 389.

Thesaurus

The name of the coding thesaurus protocol to be used for coding, if this item represents the primary code assigned by interactive or automatic coding. If you specify a coding thesaurus, you cannot specify a checklist or a codelist.

Description

A description of the item.

List of read-only attributes

With the Item Browser open, when you select the **Show** command from the **Item** menu, the Clintrial software displays, in the Show Item window, the previously described item attributes as read-only, and also displays the following additional read-only attributes:

- Modification Date — Date and time of the last modification to the panel.
- Modified By — Name of the user account that last modified the item.
- Rev State — Revision state of the panel to which the item belongs. Values are:

  UN — The panel the item belongs to is uninstalled.

  IN — The panel the item belongs to is installed.

  DE — The panel the item belongs to is deinstalled.

  PRE — The panel the item belongs to is marked for revision, and the item has not been modified.

  REV — The panel the item belongs to is marked for revision, and the item has been modified.
- Status — The item's status in the data dictionary table. Values are:

  Valid — The item is valid.

  Invalid — The item is invalid.

  A panel cannot be installed or have its revisions implemented if any of its items have an invalid status.
- Item Order — Number of the column containing the item in the update, data, and audit tables. This number defines the default order of the item in queries and reports, and in the default layout of a page section.
- Key Order — When nonzero, the item joins one or more context items in defining a unique key for the panel's tables (not in a Type 0 panel).
- Sort Key Order — If sort keys are specified, the order in which they are applied. Sorting items can be specified for types 0, 2 and 4 panels. Records within an observation will be sorted using these sorting items during regrouping.

  If sort keys are specified, then the Insert function in Enter cannot be used, since records will always be inserted in the proper place in the sort order. Enter sorts records first by grouping items, and then by sorting items. If

neither type of items are specified, Enter will sort according to CT_RECID. Sorting uses uppercase values of the items.

- Sort Descending — For each sorting item (defined by Sort Key Order), this attribute determines the sorting order. Values are:

  0 — Ascending order (the default).

  1 — Descending order.

- Grouping Item — Displays whether the item is a grouping item. Values are:

  0 — The item is not a grouping item.

  1 — The item is a grouping item.

  For information on grouping and sorting items, see "Grouping and sorting items in a panel" on page 217.

Part II: Objects for Managing Clinical Data

11 Derivations and Rules

Overview

Part II describes the Design objects used to manage clinical data or to help in the data-entry process. The objects that manage clinical data are:

- Derivations and rules

  For more information, see this chapter.
- Codelists

  For more information, see Chapter 12.
- Flags and notes

  For more information, see Chapter 13.
- Coding thesauruses

  For more information, see Chapter 14.

Validation procedures

What is validation?

Validation is the process of checking clinical data for logic and consistency with the clinical protocol by running a validation procedure on it. Data can either pass or fail validation. Records that pass validation can be merged; that is, moved to the data table.

Data can either pass or fail validation, and the record status of the record changes if it passes validation. Records must pass validation in the update table before they can be merged (moved to the data table).

For more information about validating data, see *Manage, Classify, and Lab Loader*.

What is a validation procedure?

Each panel has one *validation procedure* — a PL/SQL procedure that is automatically built at panel installation from the derivations and rules associated with the panel. The validation procedure runs when validation is run in Manage.

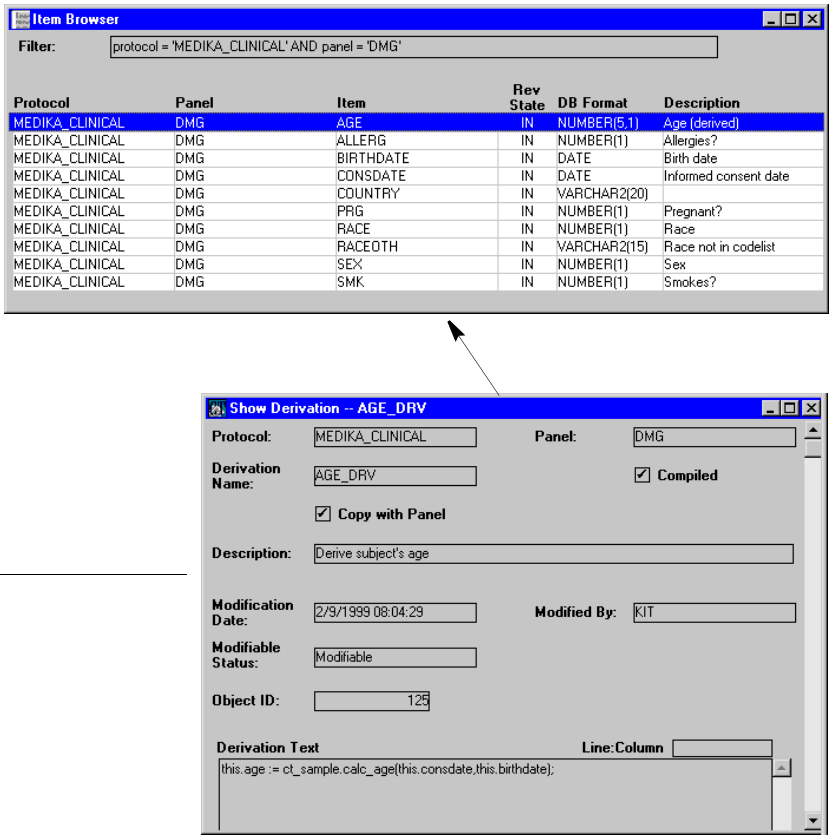
A *derivation* is a PL/SQL statement that is attached to a panel, and calculates the value of an item or a temporary variable for records during validation. Derivations are part of the validation procedure that checks clinical data before it is moved to the data table.

For example, you can create a derivation that calculates the subject's age as the difference between the subject's birth date and the informed consent date.

The following figure shows how a derivation relates to a panel and its items:

Panel and items

Derivation attached to DMG panel



In this example, item AGE is calculated from items BIRTHDATE and CONSDATE.

What is a rule?

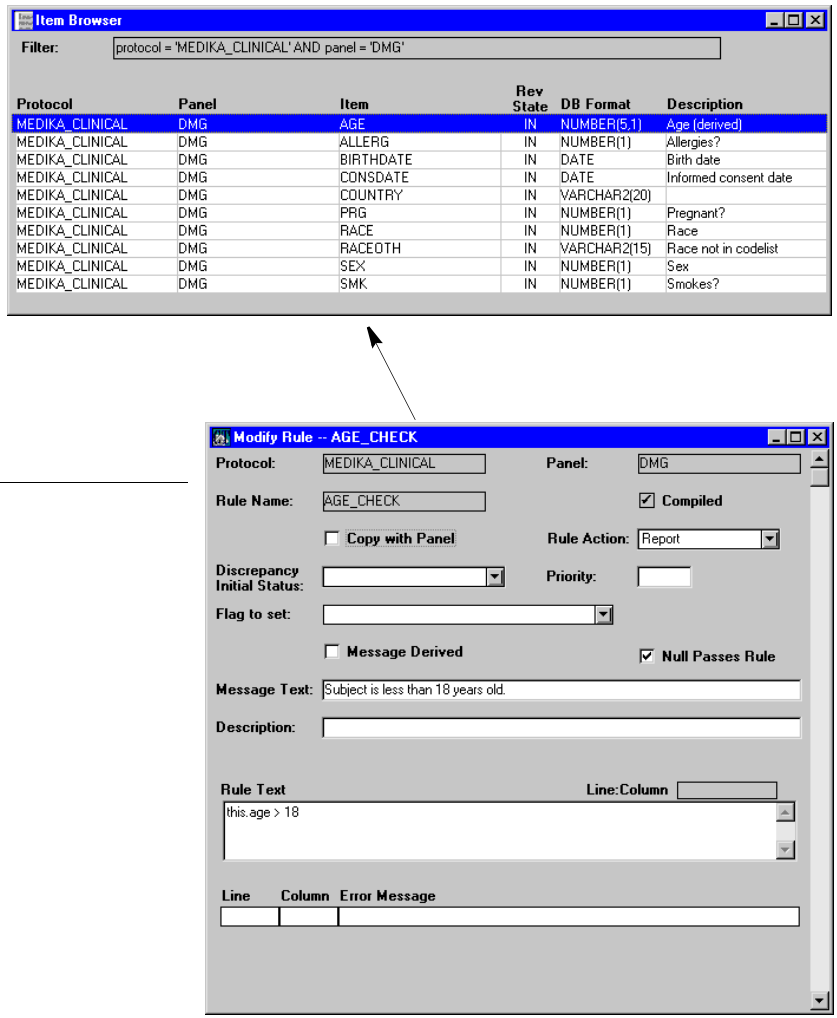
A *rule* is part of a PL/SQL statement that is attached to a panel, and used to confirm that clinical data meets the requirements of the clinical protocol. Rules become part of the validation procedure that validates clinical data before it is moved to the data table. During validation, rules evaluate to TRUE or FALSE.

For example, you can create a rule that checks that the subject's age is greater than eighteen, a criteria for inclusion in the protocol.

The following figure shows how a rule relates to a panel and its items:

Panel and items

Rule attached to DMG panel



In this example, the rule checks to make sure that the subject is over age 18.

Note: The value for the subject's age is obtained from a derivation that calculates age.

For more information on validating clinical data, see *Manage, Classify, and Lab Loader*.

How rules and derivations are created

Derivations and rules are Clintrial software objects. You set up derivations and rules in Design.

For more information on creating rules and derivations see "Process for creating rules and derivations" on page 252.

What is a validation procedure?

A *validation procedure* is a PL/SQL procedure that is built automatically from derivations and rules associated with a panel. There is one validation procedure for each panel.

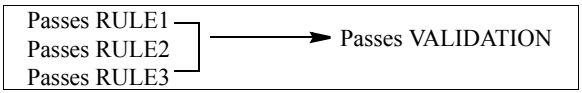
The Clintrial software automatically builds a validation procedure for a panel at the following times:

- When a panel is installed
- When revisions to a panel are implemented

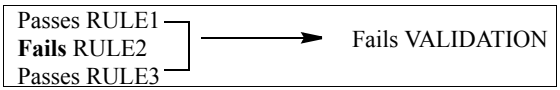
How the Clintrial software uses validation procedures

When data is validated in Manage, the Clintrial software runs the validation procedure for records in the panel being validated. Each record can either pass or fail validation, depending on whether the record passes the rules that are part of the validation.

If all rules for a record evaluate to TRUE, the record is considered to have passed validation, as shown in the following example:

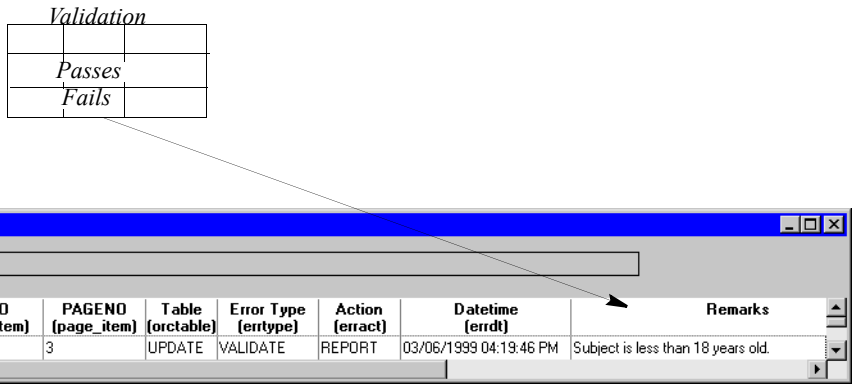


If one or more rules evaluate to FALSE for a record, the record is considered to have failed validation, as shown in the following example:



If a record fails validation in the update table, the record remains in that table with an error status, but the derived values resulting from the validation are stored in the database. If a record fails validation in the data table, the record remains in that table, but the derived values resulting from the validation are *not* stored in the database.

For each record that fails a rule, a message appears in the Remarks column of the Error Log:



In this example, the record for subject MAN113 in the DMG_UPDATE panel was rejected for failing the AGE_CHECK rule.

For information about validating data and reviewing the Error Log, see *Manage, Classify, and Lab Loader*.

Example: records before and after validation

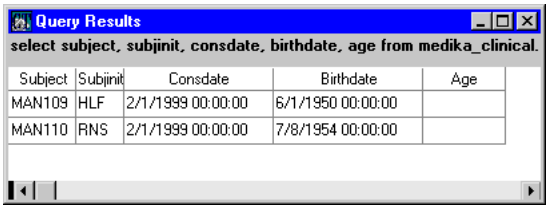
In the MEDIKA sample study book, the DMG panel includes the following items:

- CONSDATE — Date subject agreed to informed consent
- BIRTHDATE — Subject's date of birth
- AGE — Subject's age

The derivation AGE_DRV calculates the value of the AGE item as the difference between the CONSDATE item and the BIRTHDATE item.

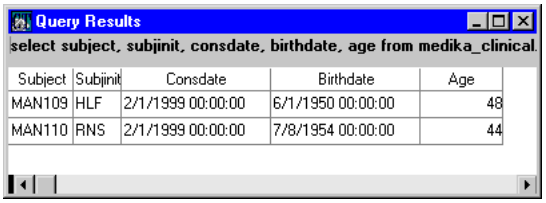
The rule AGE_CHECK checks whether the value of the derived item AGE is greater than 18. The rule AGE_INCLUSION checks that the INCLUSION panel's item for "Age > 18" has a Yes value.

Before the record is validated, there is no value for AGE, as shown in the following example:



Subject	Subjinit	Consdate	Birthdate	Age
MAN109	HLF	2/1/1999 00:00:00	6/1/1950 00:00:00	
MAN110	RNS	2/1/1999 00:00:00	7/8/1954 00:00:00	

When records for the DMG panel are validated, they can either pass or fail validation. If a record passes validation, a derived value for the AGE item is placed into the record. The following example shows the records in the previous example after validation:

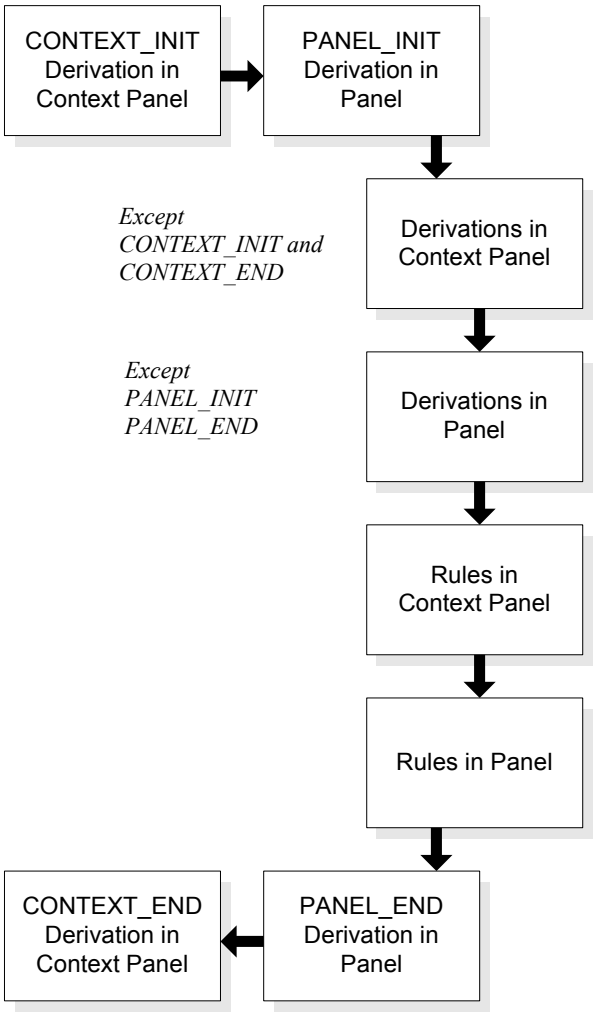


Subject	Subjinit	Consdate	Birthdate	Age
MAN109	HLF	2/1/1999 00:00:00	6/1/1950 00:00:00	48
MAN110	RNS	2/1/1999 00:00:00	7/8/1954 00:00:00	44

What is the processing order within a panel?

You can create one or more derivations and one or more rules for each panel. Additionally, you can create derivations and rules for the context panel.

When you validate records in any panel, the derivations and rules are applied in the order shown in the following figure:



Explanation of processing order

Before any records are processed, the Clintrial software processes the following special derivations:

1. The CONTEXT_INIT derivation, if any, that is attached to the context panel
2. The PANEL_INIT derivation, if any, that is attached to the panel

Then, the order of processing for each record is as follows:

1. All other derivations (other than CONTEXT_INIT and CONTEXT_END) attached to the context panel, in Oracle sort order by derivation name
2. All derivations attached to the panel (other than PANEL_INIT and PANEL_END), in Oracle processing order by derivation name
3. All rules attached to the context panel, in Oracle sort order by rule name
4. All rules attached to the panel, in Oracle sort order by rule name

Note: For more information on how Oracle sort order works with the Clintrial software, see the following section.

After all the records are processed, the Clintrial software processes the following derivations:

1. The PANEL_END derivation, if any, that is attached to the panel
2. The CONTEXT_END derivation, if any, that is attached to the context panel

For more information on special derivations, see "What is a special derivation?" on page 260.

How to ensure the correct processing order

The Oracle sort order determines the order in which rules and derivations are executed. Since the database administrator can configure this sort order (using the Oracle NLS_SORT parameter), the order in which rules and derivations are processed is not necessarily the same as the order in which they appear in the Derivation Browser or Rule Browser.

Caution: To ensure that your rules and derivations are processed in the correct order, you must find out what the Oracle sort order is on your Clintrial server, and then name your rules and derivations accordingly.

To minimize sort order problems, you can do the following:

- Minimize the use of the “underscore” character (_) in the names of rules and derivations. This character accounts for one of the main ways in which Oracle sort order can differ from the sort order of Clintrial software browsers.
- List the rules and derivations in their Oracle sort order using the appropriate SQL SELECT command to ensure that they are in the correct order.

To further ensure that the validation procedure processes a panel’s derivations and rules in the order that you want, you can:

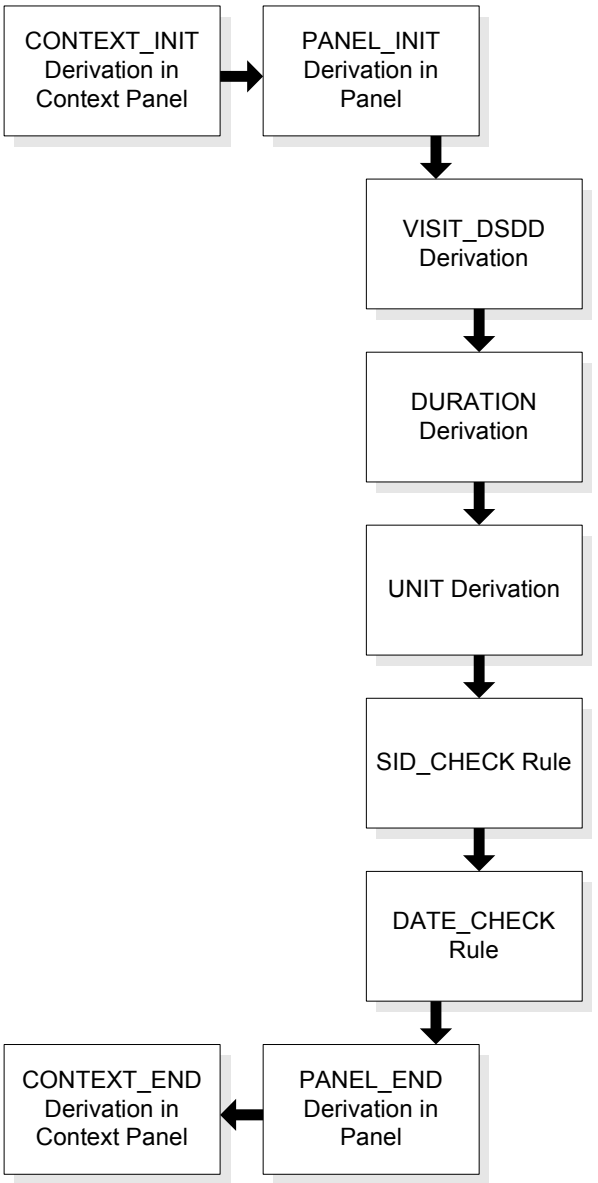
- Order the PL/SQL statements for multiple derivations within a single derivation. Handle rules similarly.
- Name the derivations and rules so that the alphabetical processing handles them in the order that you want.

Example of processing order

Suppose that the following derivations and rules exist for the CONTEXT panel and the PRVMED panel:

Panel:	Derivation or rule:
CONTEXT	CONTEXT_INIT derivation
CONTEXT	CONTEXT_END derivation
CONTEXT	SID_CHECK rule
CONTEXT	VISIT_DSDD derivation
PRVMED	PANEL_INIT derivation
PRVMED	PANEL_END derivation
PRVMED	DURATION derivation
PRVMED	UNIT derivation
PRVMED	DATE_CHECK rule

When records in the PRVMED panel are validated, derivations and rules in both the CONTEXT and PRVMED panels will be processed in the order shown in the following figure:



What is the validation order among panels?

Sometimes, derivations and rules in one panel are affected by the results of derivations and rules in other panels.

By default, panels are validated as follows:

- For interactive validation, the panels are processed in alphabetical order.
- For batch validation, the panels may or may not be processed in alphabetical order.

  The order is determined internally, based on the most efficient use of batch queue processing.

You can specify that a validation procedure processes panels in an order other than the default order, so that you can process derivations or rules in that order. To specify an order other than the default order, specify the Validation Priority attribute of any panel when you create or modify the panel. The Validation Priority attribute applies both to interactive validation and batch validation. For more information on how to set validation priority, see "Validation Priority" on page 228.

Compiling a validation procedure

When you install a panel or implement revisions in a panel, the Clintrial software creates a validation procedure automatically. However, a validation procedure can become invalid under certain conditions in which case you need to recompile it.

For example, suppose you change an element in one of your site-specific or protocol-specific PL/SQL procedures which is called by the validation procedure. You must then recompile the validation procedure to make it valid again. In addition, if any of the PL/SQL functions, procedures, or packages that are called by a validation procedure are invalid when the validation procedure is compiled, then the validation procedure is given a status of Invalid.

To ensure that a validation procedure is valid:

1. First compile the functions, procedures or packages that the validation procedure calls.
2. Compile or recompile the validation procedure.

   From the **Panel** menu, select **Compile >> Validation Procedure**.

Note: When you import a protocol, the validation procedures for the panels in that protocol are recreated and compiled. If any of these validation procedures are given a status of Invalid, you must correct the problems and recompile.

Process for creating rules and derivations

The process for creating rules and derivations in Design is as follows:

1. Optionally, create stored PL/SQL functions and procedures, or packages that contain functions or procedures, for use by rules and derivations.

Caution: Do not use COMMIT or DDL statements in custom validation routines, or invoking the Preview Global Change function will commit changes prematurely. If they are present in existing validation routines, they must be removed.

2. If you are modifying an installed panel, mark the panel for revision.
3. Create the rule or derivation.

Caution: The names of panel rules cannot be same as the names of context panel rules. Duplicate names will cause invalid validation panel procedures.

4. Enter the text of the rule or derivation.

   The text consists of PL/SQL statements that may include calls to routines for PL/SQL functions (for rules) or functions and procedures (for derivations), or for packages that contain functions and procedures.
5. Check the syntax of the rule or derivation.
6. Use Test Mode to enter values and test the results of the rule or derivation.
7. If you are modifying an installed panel, implement the revision.
8. If you are modifying an uninstalled panel, install the panel.

The following sections describe these steps. For information on marking a panel for revision, implementing revisions, and installing panels, see Chapter 10.

When to create and compile stored procedures

If your derivations and rules use PL/SQL functions, procedures, or packages, you must create and compile the functions, procedures, or packages before referring to them in a rule or derivation. Otherwise, the validation procedure cannot compile. The protocol account must have execute access to these procedures.

For information on creating and compiling PL/SQL functions, procedures, and packages, see your Oracle PL/SQL documentation and the *Reference Guide*.

How to work with installed panels

To create, modify, or delete rules or derivations for an installed panel, you must do the following:

1. Mark the panel for revision.
2. Create, modify, or delete the rule or derivation.
3. Implement the revision.

The validation procedure for the panel is created when you implement the revision.

For example, suppose you mark a derivation MY_DERIVATION for revision and then delete this derivation. If you try to create a new derivation MY_DERIVATION before implementing the revision, the Clintrial software does not allow you to save a new derivation with the same name. Only after implementing the revision that deleted the first derivation can you reuse the name for a new derivation.

How to mark a panel for revision

To mark a panel for revision:

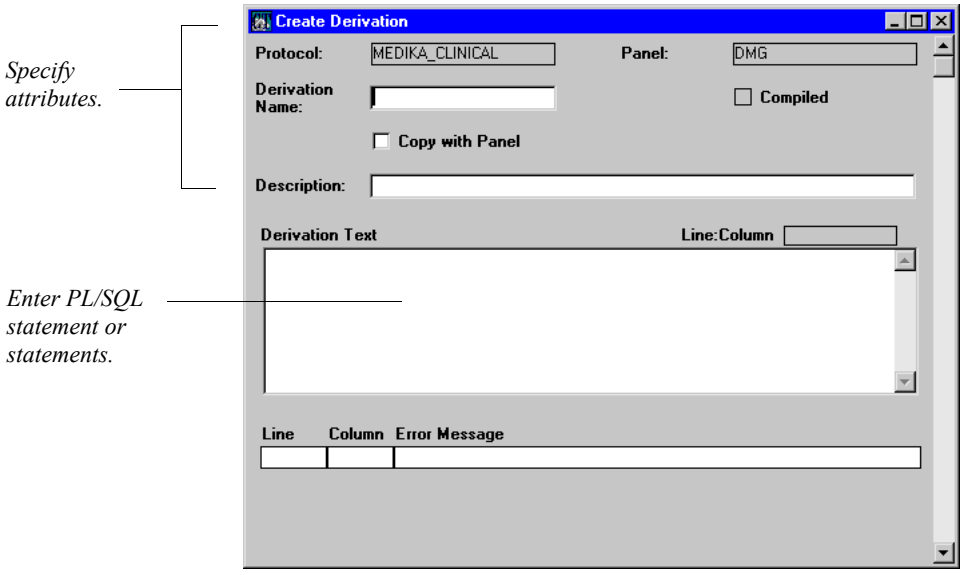
1. Open the Panel Browser.
2. Select the panel for which you want to create or modify a rule or derivation.
3. From the **Panel** menu, select **Mark for Revision**.

How to create a derivation

To create a derivation:

1. In the Panel Browser, select a panel that is uninstalled or marked for revision.
2. From the **Panel** menu, select **Derivations**. The Derivation Browser opens.

3. From the **Derivation** menu, select **Create**. The Create Derivation dialog box opens:



For more information on specifying derivation attributes, see "Creating derivations" on page 259.

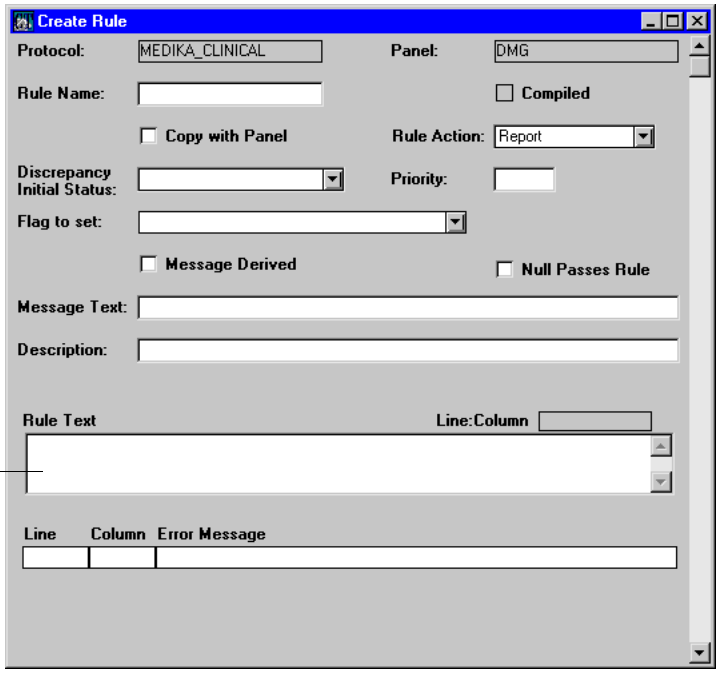
How to create a rule

To create a rule:

1. In the Panel Browser, select a panel that is uninstalled or marked for revision.
2. From the **Panel** menu, select **Rules**. The Rule Browser opens.

3. From the **Rule** menu, select **Create**.

   The Create Rule dialog box opens:



For more information on specifying rule attributes, see "Creating rules" on page 266.

How to enter PL/SQL in rules or derivations

To enter PL/SQL text in a rule or derivation, you can:

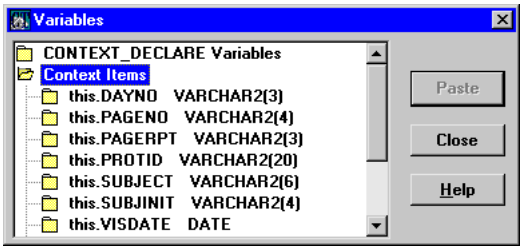
- Enter the complete PL/SQL statements (for derivations) or parts of statements (for rules) directly.
- Within the statements or parts of statements, use calls to PL/SQL functions, procedures, or packages that contain functions or procedures.

When you enter PL/SQL statements or parts of statements directly, you can select and paste variables and functions into the text of the derivation or rule you are creating or modifying.

To paste a variable:

1. From the **Syntax** menu, select **Variables**.

   The Variables dialog box opens:



   This dialog box displays a list of variables corresponding to context items, panel items, and system items, as well as variables defined in the CONTEXT.DECLARE and PANEL.DECLARE derivations.

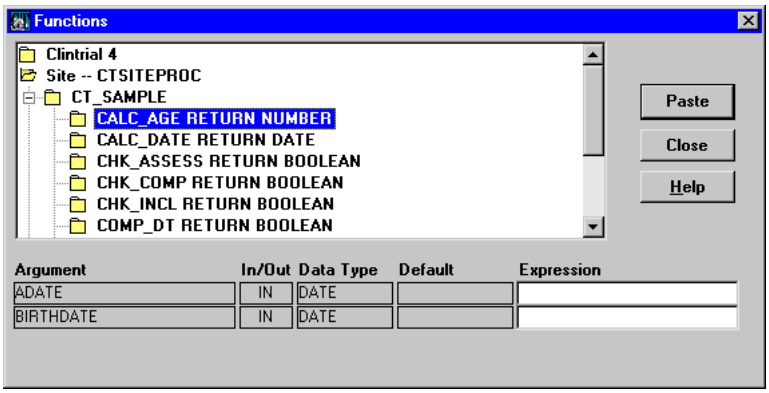
2. Select a variable from the list and then click **Paste**.

   The selected variable is pasted into the Rule Text field at the insertion point.

To paste a function:

1. From the **Syntax** menu, select **Functions**.

   The Functions dialog box opens:



   The Functions dialog box displays a list that includes Clintrial software-supplied functions and user-defined site-specific and protocol-specific functions.

2. Select a function from the list and then click **Paste**.

   The selected function is pasted into the Rule Text field at the insertion point.

The figure illustrating how a rule relates to a panel and its items shows a simple PL/SQL statement fragment that was entered directly in the text of a rule in Design.

For more information about Clintrial software-supplied system variables and system functions, see the *Reference Guide*.

How to check syntax

After you create a derivation or a rule, you can test that it processes correctly. In the window where you created the derivation or rule, run the syntax checker to test that the PL/SQL syntax is correct.

How to use Test Mode

Use Test Mode to enter values in items to determine whether the derivation or rule processes as you expected.

Note: Test Mode does not work for a derivation or a rule that refers to an item in another panel. To test a derivation or rule with references to another panel, use Enter and Manage.

Implementing panel revisions

If you are working with an installed panel, rules and derivations are assembled into a validation procedure when you implement the revision for the panel. To implement a revision:

1. Open the Panel Browser.
2. Select the panel for which you have created or modified rules and derivations.
3. From the **Panel** menu, select **Implement Revision**.

When you import existing rules and derivations

When you import a protocol, all validation procedures for the protocol's panels are recreated automatically. Therefore, when you import a protocol with existing rules and derivations that use PL/SQL functions, procedures, and packages, you should perform tasks in the following order:

1. Create and compile the PL/SQL functions, procedures, or packages in the database instance where you will import the protocol.
2. Import the protocol that uses the PL/SQL functions, procedures, and packages.

If you import the protocol before you create and compile the PL/SQL functions, procedures, and packages, you must recompile any validation procedures that refer to the PL/SQL functions, procedures, or packages.

For information on importing protocols, see Chapter 20.

Creating derivations

How to derive a value for an item

To derive a value for an item, you must do the following:

1. Select the Derived attribute of the item.
2. Attach a derivation to the panel containing the item whose value you want to derive.

For example, to derive the value of the AGE item in the DMG panel, you do the following:

1. Select the Derived attribute of the AGE item in the DMG panel.
2. Attach a derivation to the DMG panel that calculates the value of the AGE item.

For more information on setting the Derived attribute of an item, see "Derived" on page 232.

About derivations

When creating a derivation, keep in mind that a derivation can:

- Refer to multiple items to derive the value for a derived item.
- Derive values for multiple derived items in the panel.
- Refer to items from the panel to which the derivation is attached, from the context panel, or from other panels in the protocol.
- Call stored PL/SQL functions, procedures, and packages.
- Declare and use one or more temporary variables.
- Use Clintrial software-supplied variables.

Example: AGE_DRV

The figure illustrating how a derivation relates to a panel and its items shows the AGE_DRV derivation in the DMG panel of the MEDIKA_CLINICAL protocol. This derivation includes a call to the CALC_AGE procedure in the CT_SAMPLE package:

```
this.age := ct_sample.calc_age(this.consdate,this.birthdate);
```

For information on specifying the Derivation Text attribute, see the following section. For information on other derivation attributes, see "List of derivation attributes" on page 264.

What is the derivation text?

The text of a derivation is the PL/SQL text that specifies the derivation, consisting of any valid PL/SQL statement or statements. The following table shows the items to which PL/SQL statements in a derivation (including special derivations) can refer:

Panel to which the derivation is attached:	Items to which the derivation can refer:
CONTEXT	Any items in the context panel
Any panel except CONTEXT	• Any items in the context panel • Any items in the panel to which the derivation is attached

For more information on entering the derivation text, see "How to enter PL/SQL in rules or derivations" on page 255.

What is a special derivation?

The Clintrial software provides the following six special derivations that can be attached to the context panel only:

Derivation name:	Meaning:
CONTEXT_DECLARE	Use this derivation to declare global variables that can be referred to by any derivations and rules attached to any panels in the database. This derivation can contain only variable declarations. ***Note:*** This derivation cannot declare variables that are also defined by any PANEL_DECLARE derivation.
CONTEXT_INIT	Use this derivation for any initialization that will be needed by any of the derivations and rules attached to any panels in the database.

Derivation name:	Meaning:
CONTEXT_END	Use this derivation to terminate activities that are opened by the CONTEXT_INIT derivation.

The following special derivations can be attached to any panel:

Derivation name:	Meaning:
PANEL_DECLARE	Use this derivation to declare panel-specific variables that will be referenced by other derivations or rules attached to the same panel as this PANEL_DECLARE derivation. This derivation can contain only variable declarations. ***Note*:** This derivation cannot declare variables that are also declared by the CONTEXT_DECLARE derivation.
PANEL_INIT	Use this derivation for initializations for the panel to which this PANEL_INIT derivation is attached.
PANEL_END	Use this derivation for terminations of activities that are opened by the PANEL_INIT derivation.

What is a temporary variable?

A *temporary variable* is any variable that is referred to in the derivation but that is not in the panel to which the derivation is attached, the context panel, or any other panel in the protocol accessed by a SELECT statement. Temporary variables are used only within the validation procedure and are not stored in the database.

Where to declare a temporary variable

You can declare a temporary variable in the CONTEXT_DECLARE derivation, in the PANEL_DECLARE special derivation, or in the current derivation.

The location where you declare a temporary variable determines how you can reference the variable in rules and derivations.

The following table describes how you can reference temporary variables:

When the declaration is in:	The variable can be referenced by:
CONTEXT_DECLARE	Any derivations or rules attached to any panels in the database.
PANEL_DECLARE	Any derivations or rules attached to the same panel as the PANEL_DECLARE derivation.
Current derivation	The current derivation.

Example: using a temporary variable

In this example, you do the following:

1. Declare a temporary variable in the PANEL_DECLARE derivation.
2. Create a derivation and attach it to a panel.
3. Create a rule that uses the derivation and attach the rule to a panel.

Suppose that the DMG panel contains the items PREG_START and PREG_STOP, to indicate the start and stop dates of the patient's pregnancy. There is no item for the pregnancy duration, because it is not necessary to store that information. However, during validation you want to make sure that the pregnancy duration was not more than 252 days.

In the DMG panel you:

1. Declare the temporary variable preg_dur in the PANEL_DECLARE derivation:

```
preg_dur number;
```

2. Attach the following derivation to calculate the value of preg_dur:

```
preg_dur = ROUND(THIS.PREG_STOP - THIS.PREG_START);
```

3. Attach the following rule to evaluate whether the value of the temporary variable is less than or equal to 252:

```
preg_dur <= 252
```

Note: You could create a rule that combines the derivation's calculation and the rule's evaluation. However, a separate derivation for the calculation has the following advantages:

- You can reuse the derivation. If you put the calculation in the rule, you cannot use it elsewhere.
- The derivation changes the derived column's value in the data table each time the arguments' values change. Therefore, you can view the derived value in the table, for comparisons and calculations.

Example: multiple derived items

Within the same derivation, you can derive multiple items to ensure that the derived values are calculated in the appropriate order.

Suppose that the DMG panel includes the following items:

- CONSDATE — Date on which the subject gave consent
- BIRTH_MM — Two-digit representation of the month of the subject's birth
- BIRTH_DD — Two-digit representation of the day of the month of the subject's birth
- BIRTH_YYYY — Four-digit representation of the year of the subject's birth
- AGE — Subject's age in years, as derived from the subject's date of birth (stored in a temporary variable, date_birth) and CONSDATE

Although the subject's date of birth is not an item in the panel, you need to calculate it to determine the subject's age. Therefore, you must:

1. Calculate the value of a temporary variable, named date_birth, from the values of BIRTH_MM, BIRTH_DD, and BIRTH_YYYY.
2. Derive the item AGE as the difference between the temporary variable date_birth and the item CONSDATE.

You create the following derivation, named BIRTH_AGE_D:

```
/* Derive temporary variable date_birth and item AGE */


declare date_birth date;
begin
  date_birth := ct_func.convert_date(this.BIRTH_YYYY, this.BIRTH_MM,
        this.BIRTH_DD);
  this.AGE := siteDrv.DRV_CALC_AGE(this.CONSDATE, this.date_birth);
end;
```

This derivation calls two PL/SQL functions:

- A system function, convert_date, stored in the ct_func package
- A user-defined function, DRV_CALC_AGE, stored in a site-defined PL/SQL package named SITEDRV

The sample drv_calc_age procedure is as follows:

```
function drv_calc_age(adate date, birthdate date)
   return number is
begin
   if adate > birthdate then
      return trunc ((adate - birthdate)/365.14);
   else
      return null;
   end if;
end drv_calc_age;
```

When the BIRTH_AGE_D derivation is run, the value of the temporary variable date_birth is calculated first, and then the value of the item AGE is calculated.

For more information on convert_date, and other system functions, see the *Reference Guide*.

Importance of processing order

Suppose that instead of calculating both these values in one derivation, you created the following two derivations:

- BIRTH _D — Calculates the value of the temporary variable date_birth
- AGE_D — Calculates the value of the AGE item

During validation, by default, the AGE_D derivation would be run before the BIRTH_D derivation, because rules are processed in alphabetical order unless you set the panel's Validation Priority attribute. Thus, the AGE_D derivation would not result in a value, since it would not know the value of the temporary variable date_birth, which had not yet been derived.

List of derivation attributes

The following table summarizes the attributes of a derivation:

Attribute:	**Description:**
Protocol	Name of the protocol.
Panel	Name of the panel.
Derivation Name	Name of the derivation.

Attribute:	Description:
Compiled	Whether the derivation was compiled successfully. The derivation compiles when you save it successfully, when a syntax check is successful, or when you select **Compile >> Rules/Derivations** from the **Panel** menu, or **Compile** from the **Derivation** menu.
Copy with Panel	When you perform an unconnected copy of a panel, a derivation with this attribute checked is copied as a connected copy. For more information, see Chapter 19.
Description	An optional description of the derivation.
Derivation Text	PL/SQL statement(s).
Line	The number of the line in which a syntax error occurred.
Column	The number of the column at which the syntax error starts.
Error Message	The message describing the syntax error.

List of read-only attributes

When you select the **Derivation** menu's **Show** command, the Clintrial software displays the previously described derivation attributes as read-only, and also displays the following additional read-only attributes in the Show Derivation dialog box:

Attribute:	Description:
Modification Date	Date and time of the last modification to the derivation.
Modified By	Name of the user account that modified the derivation.
Modifiable Status	Modifiable if the derivation can be modified; Not Modifiable if the derivation cannot be modified. From the **Derivation** menu, select **Set Modifiable Status**.

Attribute:	Description:
Object ID	A unique identifier used internally to store the text.

Creating rules

About rules

When specifying the PL/SQL statement fragments for a rule, remember the following:

- A rule can refer to any temporary variables calculated by derivations that are:
  - Declared in the PANEL_DECLARE derivation and attached to the panel to which the rule is attached
  - Declared in the CONTEXT_DECLARE derivation and attached to the context panel
- The rule must evaluate to one value, either TRUE or FALSE.
- The rule is not a complete PL/SQL statement; therefore do not use a semicolon (;) at the end of the rule.

When creating or modifying a rule, you can set attributes that:

- Specify the rule action, REPORT or REJECT, that determines what happens when a record fails a rule.
- Associate a flag with a rule.
- Set the initial discrepancy status for use with Resolve.
- Specify a text message or derived message that appears in the Error Log when a record fails a rule.
- Specify whether the rule must be included when the panel is copied.
- Specify whether a value of NULL passes or fails a rule.
- Specify the PL/SQL rule text.

Caution: The names of panel rules cannot be same as the names of context panel rules. Duplicate names will cause invalid validation panel procedures.

For information on specifying the Rule Text attribute, see "What is the rule text?" on page 267.

For information on text messages, see "What is a text message?" on page 268.

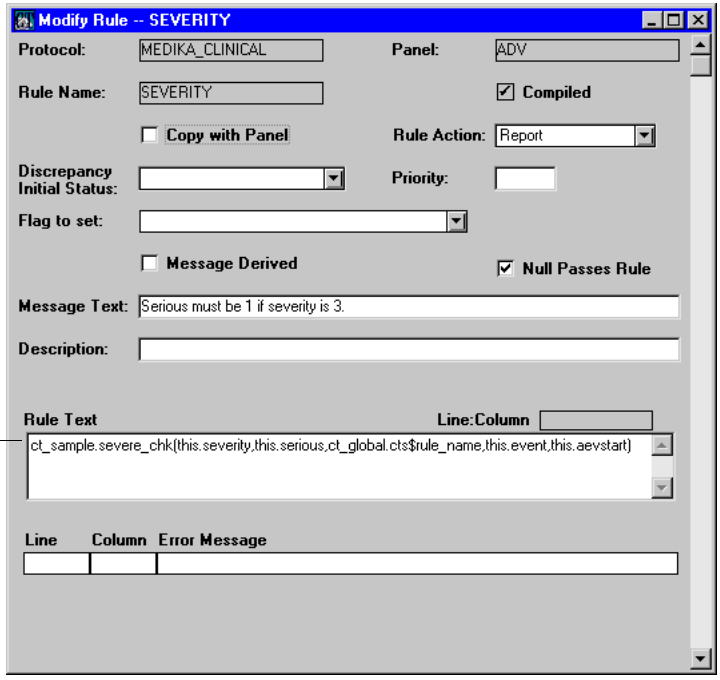
For information on derived messages, see "What is a derived message?" on page 269.

For information on rule attributes, see "List of rule attributes" on page 270.

For more information on PL/SQL syntax used in Clintrial software rules, see the Oracle PL/SQL documentation and the *Reference Guide*.

Example: SEVERITY rule

The following example shows the SEVERITY rule in the ADV panel of the MEDIKA_CLINICAL protocol. This rule includes a call to the stored function SEVERE_CHK in the CT_SAMPLE package. When you select SEVERITY in the Rule Browser, and then from the **Rule** menu, select **Modify**, the Modify Rule dialog box opens:



PL/SQL statement fragment calls function

What is the rule text?

The text of a rule is the PL/SQL text that specifies the rule, consisting of any valid part of a PL/SQL statement.

The following table shows the items to which a rule can refer:

Panel to which the rule is attached:	Items to which the rule can refer:
CONTEXT	• Any items in the context panel • Variables from CONTEXT_DECLARE and PANEL_DECLARE
Any panel except CONTEXT	• Any items in the context panel • Any items in the panel to which the rule is attached • Variables from CONTEXT_DECLARE and PANEL_DECLARE

For more information on entering the rule text, see "How to enter PL/SQL in rules or derivations" on page 255.

How a rule handles a NULL evaluation

If a rule evaluates to NULL, the Clintrial software can treat the rule either as if it evaluated to TRUE or as if it evaluated to FALSE.

If you check the Null Passes Rule attribute for a rule, then if a rule evaluates to NULL for a record, the record passes the rule. If you do not check Null Passes Rule, then the record fails the rule.

What is a text message?

For each rule, you can specify the message that will appear in the Remarks column of the Error Log if the record fails the rule. You can specify either a text message or a derived message.

A *text message* is a text string with embedded references to an item in the rule. Each embedded reference is in the format %*item-name*, with *item-name* in uppercase. When the message appears in the Error Log, the reference is replaced by a value for the item.

Example: text message

Suppose that you want a rule to check whether the value of the AGE item is greater than 17 and less than 66. You create the following rule, named AGE_CHECK:

this.AGE > 17 AND this.AGE < 66

You then create the following text message:

Age %AGE is out of range.

If the value of the AGE item for a record is 15, the record fails the rule during validation and the following message appears in the Error Log:

Age 15 is out of range.

Notice that in the message, the embedded reference %AGE is replaced by the value of the item AGE for the record.

Default text message

If you do not specify a text message or a derived message, then a default text message appears in the Error Log if a record fails the rule. The default text message is in the format:

Failed *panel-name* rule *rule-name*.

What is a derived message?

As shown in the preceding example, a text message can reflect the value of an item that is referenced by the rule. Another type of message, a *derived message*, allows you to generate a message that reflects derived values for items or temporary variables.

To create a derived message, do the following:

1. For the rule, check Message Derived.
2. Attach to the same panel a derivation that derives a temporary variable named *rule-name*$MSG.

   Note: You must initialize this variable in the derivation; however, you must not declare it because the Clintrial software declares it automatically as a result of the Message Derived attribute.

Example: derived message

Suppose that you want the message for the rule AGE_CHECK to show not only that age 15 is out of range, but also the number of years by which it is out of range. The Clintrial software needs to derive the difference between the specified value for the item AGE, and the minimum value (18) or maximum value (65) of the item age. You can create a derivation that derives that difference and specifies a message:

```
if this.age < 18 then
  age_check$msg:= 'Age' ||this.age|| 'is' ||18-this.age||
  'years less than minimum age 18';
elsif this.age > 65 then
  age_check$msg:= 'Age' ||this.age|| 'is' ||this.age-65||
  'years greater than maximum age 65';
else age_check$msg:= null;
end if;
```

Now, if the record fails the rule because 15 was entered for the item AGE, the following message appears in the Error Log:

```
Age 15 is 3 years less than minimum age 18.
```

In addition to using the derivation to derive a message, you can use the derivation to test the rule. In this case, the rule can check whether the derivation message results in a value other than null. If it results in a value other than null, then the rule evaluates to FALSE; if it results in null, then the rule evaluates to TRUE.

For example, suppose that you have attached the preceding derivation to a panel. You could then create the following rule:

```
age_check$msg IS NULL
```

If the value of the AGE item is within the age range 18 through 65, the derivation results in a null value. If the derivation results in a null value, the rule evaluates to TRUE (provided you checked Null Passes Rule).

List of rule attributes

The following table describes the attributes of a rule:

Attribute:	**Description:**
Protocol	Name of protocol.

Attribute:	Description:
Panel	Name of protocol.
Rule Name	Name of the rule.
Compiled	Whether the rule was compiled successfully. The derivation compiles when you save it successfully, when a syntax check is successful, or when you select **Compile >> Rules/Derivations** from the **Panel** menu or **Compile** from the **Rule** menu.
Copy with Panel	When you perform an unconnected copy of a panel, a rule with this attribute checked is copied as a connected copy. For more information, see Chapter 19.
Rule Action	The rule action can be either REPORT or REJECT. If the rule action is REPORT, then the following events occur if a record fails the rule during validation: • If the record is in the update table, the record status changes to indicate that the record passed validation. If the record is in the data table, its status does not change. • The entry in the Error Log shows REPORT in the Action column. If the rule action is REJECT, then the following events occur if a record fails the rule during validation: • The record status changes to indicate failure (UPDATE). • The record status does not change (DATA). • The entry in the Error Log shows REJECT in the Action column.

Attribute:	Description:
Discrepancy Initial Status Default: New	If the protocol is set up for Resolve, the Clintrial software creates discrepancy records automatically after each rule is evaluated. Each discrepancy record requires an initial status. Use this attribute to set an initial status other than the default. The values are: • Autoclosed • Confirmed As Is • Linked • New • No Action Needed • Obsolete • Ready To Send • Reissued • Released • Resolution Proposed • Resolved Internally • Resolved Manually • Sent • Source Deleted • Unresolvable For more information see *Enter, Resolve, and Retrieve.*
Priority	A priority ranking that you set for this discrepancy. For more information see *Enter, Resolve, and Retrieve.*
Flag to set	Flag to be attached to the record if the record fails the rule.
Message Derived	If selected, then a derivation (named *rule-name*$MSG) attached to the panel determines the message that appears in the Remarks column of the Error Log when a record fails the rule.
Null Passes Rule	If selected, then if the rule evaluates to NULL, it will be treated as if it evaluated to TRUE.

Attribute:	Description:
Message Text	Text of the message to appear in the Remarks column of the Error Log if the record fails the rule.
Description	An optional description of the rule.
Rule Text	Part of a PL/SQL statement that evaluates to TRUE or FALSE.
Line	Number of the line in which a syntax error occurred.
Column	Number of column at which the syntax error starts.
Error Message	A message describing the syntax error.

List of read-only attributes

When you select the **Rule** menu's **Show** command, the Clintrial software displays the previously described rule attributes as read-only, and also displays the following additional read-only attributes in the Show Rule window:

Attribute:	Description:
Modification Date	Date and time of the last modification to the rule.
Modified By	Name of user account that modified the rule.
Modifiable Status	Modifiable if the derivation can be modified; Not Modifiable if the derivation cannot be modified. From the **Rule** menu, select **Set Modifiable Status**.
Object ID	A unique identifier used internally to store the text.

Examples

This section contains examples of various derivation and rules.

Panels and items

The following examples assume the existence of the panels and items described in the following tables.

Suppose that the CONTEXT panel includes the following items:

Item:	Database format:	Meaning:
SUBJECT	VARCHAR2(6)	Subject identifier.
VISNO	VARCHAR2(1)	Visit identifier.
VISDATE	DATE	Visit date.

Suppose that the DMG panel includes the following items:

Item:	Database format:	Meaning:
SEX	NUMBER(1)	Gender of subject.
PREGRESULT	NUMBER(1)	Result of pregnancy test.
PREGDATE	DATE	Date on which the pregnancy began.

Suppose that the DRGADM panel contains the following items:

Item:	Database format:	Meaning:
CAPSPERDAY	NUMBER(1)	Number of capsules taken daily.
DOSESTART	DATE	Date on which administration of the drug began.
DOSESTOP	DATE	Date on which administration of the drug ended.

Item:	Database format:	Meaning:
NUMDAYS	NUMBER(2)	Number of days that a constant dose was maintained. This value can be derived as the difference between the value of DOSESTOP and the value of DOSESTART.
CAPS_X_DAY	NUMBER(2)	Total number of capsules taken. This value can be derived by multiplying the derived value of NUMDAYS by the value of CAPSPERDAY.

Suppose that the DRGCMP panel contains the following items:

Item:	Database format:	Meaning:
TOTCAPS	TOTCAPS(2)	Total number of capsules taken per day. This value can be derived from a variable that is calculated from the CAPS_X_DAYS item in the DRGADM panel.
CAPSREQ	NUMBER(2)	Number of capsules required per day.
COMPLIANCE	NUMBER(3)	Percent compliance. This value can be derived by dividing the value of TOTCAPS by the value of CAPSREQ and multiplying the result by 100.

Demographic information

For any records that are validated in any panel, you want to check that there is at least one record in the DMG panel with the same SUBJECT as the record being validated. You can use the find_n_records system function to find the number of records that contain the SUBJECT. You attach the following rule to the context panel:

```
ct_func.find_n_records(cts$protocol,'DMG', 'UD',
('SUBJECT= ''' || this.SUBJECT)||'''') != 0
```

If the find_n_records function finds any records that contain the SUBJECT, it returns the number of records and the rule passes.

Because this rule is attached to the context panel, it becomes part of the validation procedure for each panel in the protocol. When records in the LAB panel are validated, this rule ensures that for each record in the LAB panel, the DMG panel contains a record with the same value for the SUBJECT item.

Note: In this example, SUBJECT has a database format of VARCHAR2, and therefore must be contained within the sets of four single quotation marks and concatenation marks. If SUBJECT had a database format of INTEGER, you would not need the quotation marks and concatenation marks.

Gender and pregnancy

The following PL/SQL function named vld_preg checks the consistency between the item SEX and the items PREGRESULT and PREGDATE. If the value of SEX is M, and there is a value for PREGRESULT or PREGDATE, then the function returns the value False. This function calls two system functions, lookup_vars and get_item:

```
/* If SEX is M, there is no PREGRESULT or PREGDATE */
function vld_preg(i_protocol varchar2, test_result varchar2, test_date date, sid
varchar2)
   return boolean is
tbl_found varchar2;
one_sex varchar2(5);
begin
      tbl_found := ct_func.lookup_vars(i_protocol, 'DMG', 'UD',

    ('SUBJECT  ='|| sid), 'SEX', sex_list);
   one_sex := ct_string.get_item(sex_list, 1);
   if one_sex = 'M' and (test_result is not null or test_date is not
      null) then
      return False;
   else
      return True;
   end if;
end;
```

If you store the preceding function in a PL/SQL package named SITERULE, you can attach the following rule to the DMG panel:

```
/* Check value of SEX against value of PREGRESULT and PREGDATE */
siteRule.vld_preg(cts$protocol, this.PREGRESULT, this.PREGDATE,
this.SUBJECT)
```

Null items

You can use the IS_EMPTY system function to check whether a value for an item is present. For the DMG panel, to have the validation procedure check whether there is a value for the SEX item, you can create the following rule, which calls the stored function IS_NOTEMPTY and passes in the item name SEX:

```
ct_func.is_notempty(ct_string.make_list(this.SEX)) = TRUE
```

If a value for SEX exists, the function returns TRUE, and the rule passes.

For more information on the CT_FUNC.IS_NOTEMPTY function and the CT_STRING.MAKE_LIST function, see the *Reference Guide*.

Capsules taken and percent compliance

This example shows derivations and rules for both the DRGADM and DRGCMP panels.

For the DRGADM panel, you create the following derivation to derive the value of the NUMDAYS item, and then to derive the value of the CAPS_X_DAYS item:

```
/* Derive NUMDAYS and CAPS_X_DAYS */
this.NUMDAYS := trunc(this.DOSESTOP - this.DOSESTART);
this.CAPS_X_DAYS := this.NUMDAYS * this.CAPSPERDAY;
```

For the DRGADM panel, you also create the following rule to check whether the value of the DOSESTOP item is greater than the value of the DOSESTART item:

```
/* Check if DOSESTOP is greater than DOSESTART */
(this.DOSESTOP > this.DOSESTART)
```

For the DRGCMP panel, you create the following derivation to derive the value of the TOTCAPS and COMPLIANCE items:

```
/* Derive TOTCAPS and COMPLIANCE as follows:
Initialize variable totalCaps to 0.
If record is in UPDATE table, look in DRGADM_UPDATE table for records with the
same SUBJECT and VISNO as this DRGCMP_UPDATE record. For each such record,
add the value of CAPS_X_DAYS from the DRGADM_UPDATE record to the value of
totalCaps.
If record is not in UPDATE table, do the same, except use DRGADM_DATA table.
Derive TOTCAPS by assigning to it the value of totalCaps.
Derive COMPLIANCE by dividing TOTCAPS by CAPSREQ and multiplying by 100.
*/
declare
   totalCaps number(3) := 0;
begin
   if cts$table = 'UPDATE' then
      begin
      SELECT SUM(DR.CAPS_X_DAYS) INTO totalCaps FROM
DRGCMP_UPDATE DR
      WHERE DR.SUBJECT = this.SUBJECT AND DR.VISNO = this.VISNO;
      exception when no_data_found then totalCaps := 0;
      end;
   else
      begin
      SELECT SUM(DR.CAPS_X_DAYS) INTO totalCaps FROM DRGCMP_DATA
DR
      WHERE DR.SUBJECT = this.SUBJECT AND DR.VISNO = this.VISNO;
      exception when no_data_found then totalCaps := 0;
      end;
   end if;
   this.TOTCAPS:= totalCaps;
end;
this.COMPLIANCE := Round(this.TOTCAPS/this.CAPSREQ)*100);
```

Note: Because the derivation for the DRGCMP panel uses an item (CAPS_X_DAYS) derived for the DRGADM panel, users who are validating data should validate data in the DRGADM panel before the DRGCMP panel. This example assumes DRGCMP and DRGADM records are always merged together.

12 Codelists

Creating a codelist

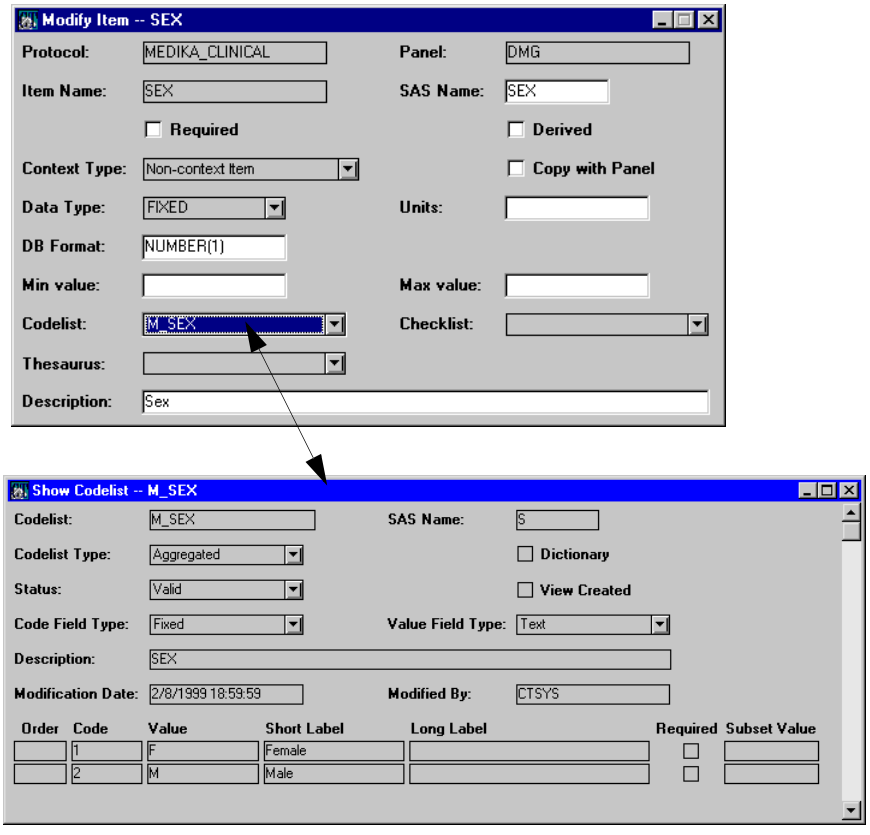
What is a codelist?

A *codelist* is a set of codes and corresponding set of values. If a codelist is attached to an item, you can only enter codes (or values) from the codelist for the item.

Codelists are available to all protocols within the database instance.

How do codelists relate to items?

The following figure shows the relationship between a codelist and an item:



Advantages of using a codelist

Some advantages of using a codelist are:

- Data validity is ensured when only codes or values from the codelist can be entered.
- The storage of codes (rather than values) in the database typically saves database space.
- Typing during data entry is typically quicker if codes are entered (rather than values).
- Internationalization is enhanced.

The figure that shows how a codelist relates to an item displays the simple M_SEX codelist. Attached to the SEX item in MEDIKA_CLINICAL's DMG panel, M_SEX allows the data to be entered as M or F, or as 1 or 2 (depending on the setting of the page section layout's Enter as Code attribute), and stored as 1 or 2. The Short Label field can contain a brief description (up to 80 characters) and the Long Label field can contain a longer description (up to 240 characters). The Order field lets you specify the sequence in which the items are displayed. The Subset Value field lets you specify values for defining subset codelists.

The following example shows how you can allow for the entry of more than one value for the same code:

Modify Codelist Values -- M_SEX2

Order	Code	Value	Short Label	Long Label	Required	Subset Value
1	1	F	Female	Female subject	☐	
2	1	Female	Female	Female subject	☐	
3	1	1	Female	Female subject	☐	
4	2	M	Male	Male subject	☐	
5	2	Male	Male	Male subject	☐	
6	2	2	Male	Male subject	☐	

In this example, the SEX data can be entered as either as F, Female or 1 to designate a female subject, and as M, Male or 2 to designate a male subject.

In this example, the values 1 and 2 correspond to the codes 1 and 2, but the Clintrial software must still encode these values when the data is stored in the clinical data table. You can require the data to be entered either as a code or as a value:

- If *code entry* is required, then a code from the codelist must be entered.

  In the page section layout, from the **Design** menu, select **Style** and select the Enter as Code attribute.

- If *value entry* is required, then a codelist value must be entered. However, the value is then encoded, and the code (rather than the value) is stored in the clinical data table.

  In the page section layout, from the **Design** menu, select **Style** and clear the Enter as Code attribute.

You can also have the short label display when a code or value is entered in a field in Enter that uses a codelist. To enable this display, in the page section layout, from the **Design** menu, select **Style** and check the Show Codelist Label attribute.

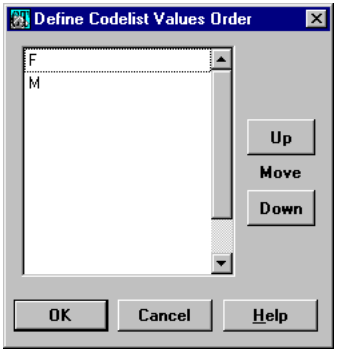
For more information on subset codelists, see "What is a subset codelist?" on page 289.

For more information on setting the Enter as Code attribute, see "Design menu commands" on page 388.

Reordering codelist values

You can also reorder the sequence in which the codelist values are displayed. (By default, codelist values are sorted lexicographically, but you can define any order.) With the **Modify Codelist Values** dialog box open, from the **Edit** menu, select **Reorder Rows**.

The Define Codelist Values Order dialog box opens, which you can use to change the order of the values:



For example, you can reverse the order in which the values for Male and Female display in the M_SEX codelist.

What is a code field type?

A *code field type* is a field type that defines whether the code from a codelist is stored as text or as a number in the coded item in the panel. The code field type must be of the same data type as the coded item in the panel.

When you create a codelist you must specify a code field type.

Only codelists with code field types of the same data type as the codelist item in the panel are available for use as codelists for the item.

Note: Only the codelist code is stored for an item with an associated codelist. However, if the Enter as Code attribute is cleared, the data-entry operator must enter the codelist's value, which can have a different data type than that of the item and the corresponding code field type of the codelist.

For example, suppose the Enter as Code attribute is cleared. The data-entry operator must enter the codelist value of F or M. However, it is the codelist code, not the codelist value, that is stored. Therefore, the item to which the codelist is attached must have the data type FIXED to match the codelist's Code Field Type attribute (in the Medika sample study).

How to use a codelist as a checklist

A *checklist* is a type of codelist for which no encoding is performed. That is, the codelist value is stored, not the codelist code. Typically, you use a checklist to view valid entries for a field. Entered values are checked against the checklist to see if they are valid.

You can override the checklist to allow the data-entry operator to enter data other than the checklist values. In the page section layout, from the **Design** menu, select **Attributes** and select the Override attribute.

What is a value field type?

A *value field type* is a field type that defines whether the value from a codelist is stored as text or as a number in the checklist item in a panel. The value field type must be of the same data type as the checklist item in the panel.

Only codelists with value field types of the same data type as the checklist item in the panel are available for use as checklists for the item.

Summary of use of codes and values

The following table summarizes the use of codes and values in codelists and checklists:

Using as:	Require entry of:	Store as:	Action:
Codelist	Code	Code	Check Enter as Code in page section layout.
Codelist	Value	Code	Clear Enter as Code in page section layout (default).
Checklist	Value	Value	Default — no other action required.
Checklist	Value or other data entry	Value or other data entry	Select Override in page section layout to allow entry of data other than the values in the checklist.

What is decoding?

If the Enter as Code attribute is cleared, when you retrieve data, the Clintrial software decodes, or translates, the codes into values. For example, if the value M is stored as the code 2, when you retrieve the data, the code 2 is translated into the value M.

You can use one codelist to encode data at data entry, and a different codelist to decode it at retrieval time, if the alternate codelist has the same code data type.

If you use duplicate codes, you may want to confirm that the labels for the duplicate codes are the same. Also, for the purpose of decoding during retrievals, you should create an alternate codelist with no duplicate codes.

Note: A codelist corresponds to a SAS format. When you define a codelist, you can give it a SAS name, which will be its format name if you use it to create a SAS format.

Types of codelists

You can create the following types of codelists:

- *Aggregated codelist* — A codelist that is stored in a single Oracle database table with other codelists.
- *Unaggregated codelist*— A codelist that is stored in its own Oracle database table.
- *View codelist* — A codelist that does not contain actual data but provides a view onto a Type 0 panel or other Oracle table. For information on view codelists, see "Steps to create a view codelist" on page 287.
- *Subset codelist* — An Oracle view onto a base codelist. By using an optional subset restriction clause, a subset codelist can make available only certain codes from the base codelist. For information on view codelists, see "Steps to create a subset codelist" on page 290.

Aggregated codelists save database space and reduce the time and resources required to initialize page sections that access codelists. Aggregation is recommended if the codelist contains less than 100 rows.

Generally, codelists containing more than 100 rows should be unaggregated, because aggregation of a large codelist may increase the search time required for information in all the aggregated codelists.

Using a codelist to initialize a sequence

You can use a codelist to initialize the values for a sequence in a page section layout.

For more information about sequences and page section layout, see "Design menu commands" on page 388.

Summary of codelist use

The following table summarizes the Clintrial software objects with attributes related to codelists, and the actions you need to take on those objects to implement a codelist:

Object:	Action:	Attribute related to codelist activity:
Codelist	Create or modify.	Set all attributes.
	Modify values.	Specify the codes, values, and labels.

Object:	Action:	Attribute related to codelist activity:
Item	Create or modify.	Set whether the item uses the codelist as a codelist or a checklist, and select a codelist with a corresponding code field type (for a codelist) or value field type (for a checklist). For more information, see Chapter 10.
Page section layout	Edit page section layout Design Attributes.	If this is a checklist item, specify whether to allow the user to override values in the checklist.
	Edit page section layout Design Styles.	If this is a codelist item, specify whether to require data entry as code and whether to display the codelist short label. Also specify what style to use (drop-down list, check box, radio buttons).
	Edit page section layout Design Sequences.	Initialize the sequence with one or more codelists. For more information, see Chapter 17.

View codelists

What is a view codelist?

A *view codelist* is a codelist that does not contain actual data but provides a view onto a Type 0 panel or other Oracle database table.

The tables store data for use by the view codelist. By emulating a codelist and storing codelist data in clinical data tables, a view codelist has the following advantages:

- Changes to the codelist can be audited (if the base is panel table).
- Importing of large standard codelists (for example, COSTART) is simplified.
- Efficient support of the cross-national coding schemes required by international companies is provided.

- The panel containing codelist records can be validated to ensure that codelist entries meet your criteria (if the base is panel table).

Steps to create a view codelist

To create a view codelist based on a Type 0 panel:

1. Create a clinical data protocol exclusively for managing view codelists.

   This is recommended but not required.
2. Create a Type 0 panel for data for use with one or more view codelists.
3. Create items for each panel.

   You must create items to map to codelist data for code, value, short label, and long label. You can use any names but the items for the codelist data must be of data type TEXT. The database format for each item must be no longer than that of the corresponding column in the codelist table.

   You can create additional items in the panel. For example, if codelist entries are shared by users of different languages, you could create additional value, short label, and long label items for each language. However, the same code would be stored for all language users.

   You can also specify status, code order, subset required and subset value items.
4. Optionally, create any rules or derivations.
5. Install the panels.
6. Create valid page section layouts. Study books are created automatically.
7. Load the codelist data interactively in Enter or by batch loading in Manage.
8. Validate and merge the data in Manage. View codelists only use records in the *panel-name*_DATA table.
9. With the Codelist Browser open, create a codelist with the codelist type of View Codelist and then do the following:

   a. Specify a base protocol, a view base panel, and items to store the codelist data.

   b. Optionally, create a SQL view restriction to select the codelist data in the view base panel based on a conditional statement.

When you save the view codelist, it is available for association with an item in any protocol in the database.

For information on the view codelist attributes, see "List of view codelist attributes" on page 295.

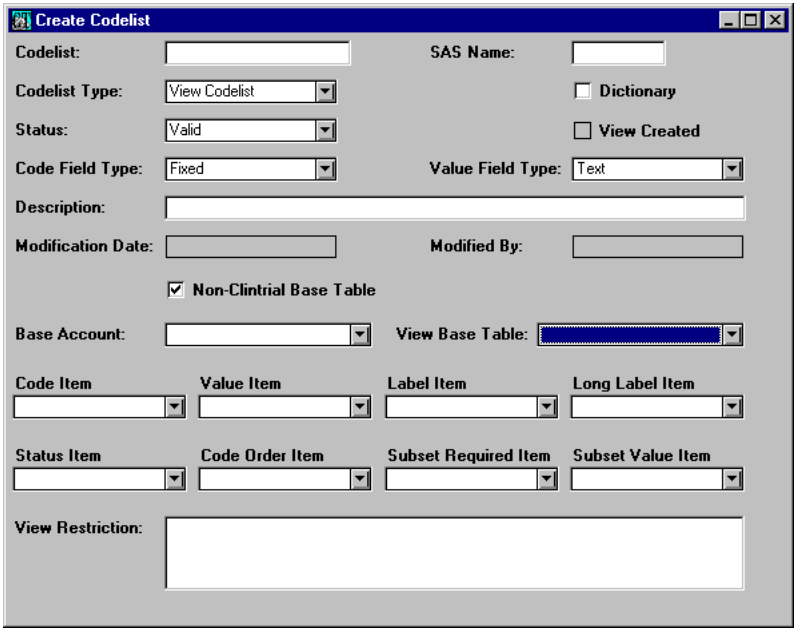
By default, a view codelist is based on a Clintrial software panel. However, you can also base a view codelist on a non-Clintrial Oracle database table by doing the following:

1. Grant the SELECT privilege on the Oracle table to the CTSCODES account. For example:

   SQL> **GRANT SELECT ON WHO_DATA TO CTSCODES WITH GRANT OPTION;**

2. In the **Create Codelist** dialog box, select the Non-Clintrial Base Table check box.

   The Base Account and View Base Table fields become modifiable:



3. From the Base Account drop-down list, select the account name for the Oracle base table that you want to specify.
4. From the View Base Table drop-down list, select the base table for the view codelist, and then specify items to store the codelist data (and, optionally, a view restriction clause). Save your changes.

What is an Oracle base table?

Throughout this guide, the term *Oracle base table* refers to an Oracle database table on which a Clintrial software view is based.

Subset codelists

What is a subset codelist?

A *subset codelist* is an Oracle view onto a base codelist. By using an optional subset restriction clause, a subset codelist can make available only certain codes from the base codelist

By making it possible to define multiple subsets of a base codelist, a subset codelist has the following advantages:

- Codelists containing the same *type* of value (for example, associated with the item SEVERITY), in one group of studies can have one subset of the base set of values, and a different subset of values in another group studies. You do not have to create a different base codelist for each set of values.
- Maintaining all the codelist values in one base codelist simplifies maintenance and standardization.
- Defining different subsets on the same base codelist provides added flexibility and specialization.

Requirements for base and subset codelists

The following requirements for base and subset codelists are as follows:

- A base codelist must have the Data Dictionary attribute checked.
- A data dictionary codelist can only be referred to in a data dictionary protocol.

Steps to create a subset codelist

To create a subset codelist, do the following:

1. Create a base codelist for the subset codelist.

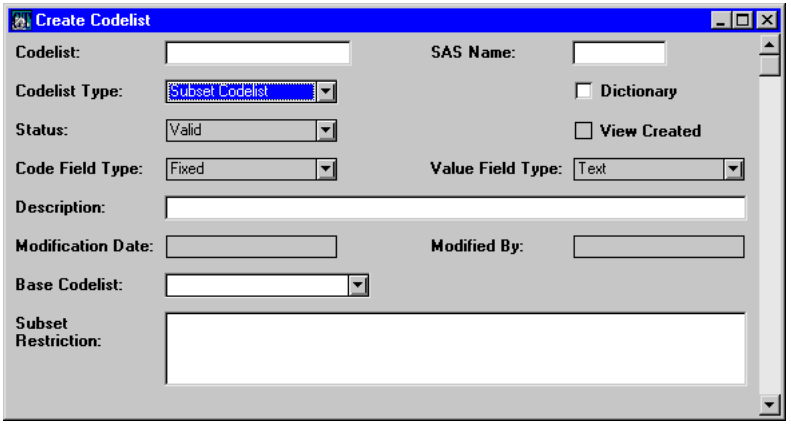
   When you specify codelist values for the base codelist, you can optionally assign a subset value to each codelist value. In base codelists containing a large number of values, assigning subset values can simplify the restriction clause used to create a subset codelist. For example, you can assign a subset value of "1" to several codelist values you want included in a particular subset codelist, and then simply specify this single value in the subset restriction clause. See "Example of subset codelist" on page 291.

2. With the Codelist Browser open, from the **Codelist** menu, select **Create**.

   The Create Codelist dialog box opens.

3. From the Codelist Type drop-down list, select **Subset Codelist**.

   The Base Codelist and Subset Restriction fields become modifiable:



4. From the Base Codelist drop-down list, select the name of a base codelist.
5. Optionally, in the Subset Restriction field, enter a subset restriction clause.

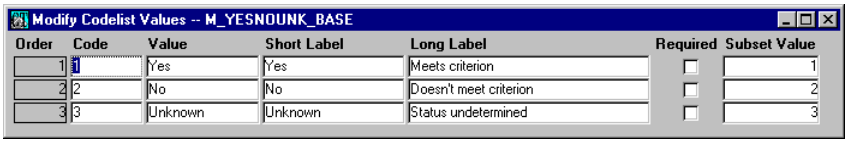
   The *subset restriction clause* is an Oracle WHERE clause that defines this subset relative to the base codelist. This clause can be entered with or without the keyword WHERE, and is not required.

   In addition to the range of codelist values specified by this restriction clause, all codelist values in the base codelist that are marked Required will be included in the subset codelist.

Example of subset codelist

For example, consider a base codelist with values of Yes, No and Unknown. (This codelist could be used with a set of exclusion criteria such as the EXCLUS page section in the Medika Sample Study.)

The set of codelist values in the base codelist might appear as follows:



From this base codelist, you could create subset codelists with the following sets of values:

- Yes/No
- Yes/No/Unknown

(In theory, you could create subset codelists with all seven possible combinations of these three values.) The following are the restriction clauses for these subset codelists.

- For the Yes/No subset codelist, either:

  VALUE='Yes' OR VALUE='No'

  SUBSET_VALUE<=2

  Note: You can reference the base codelist values directly (first example) or you can reference the subset values you have assigned to the codelist values (second example).

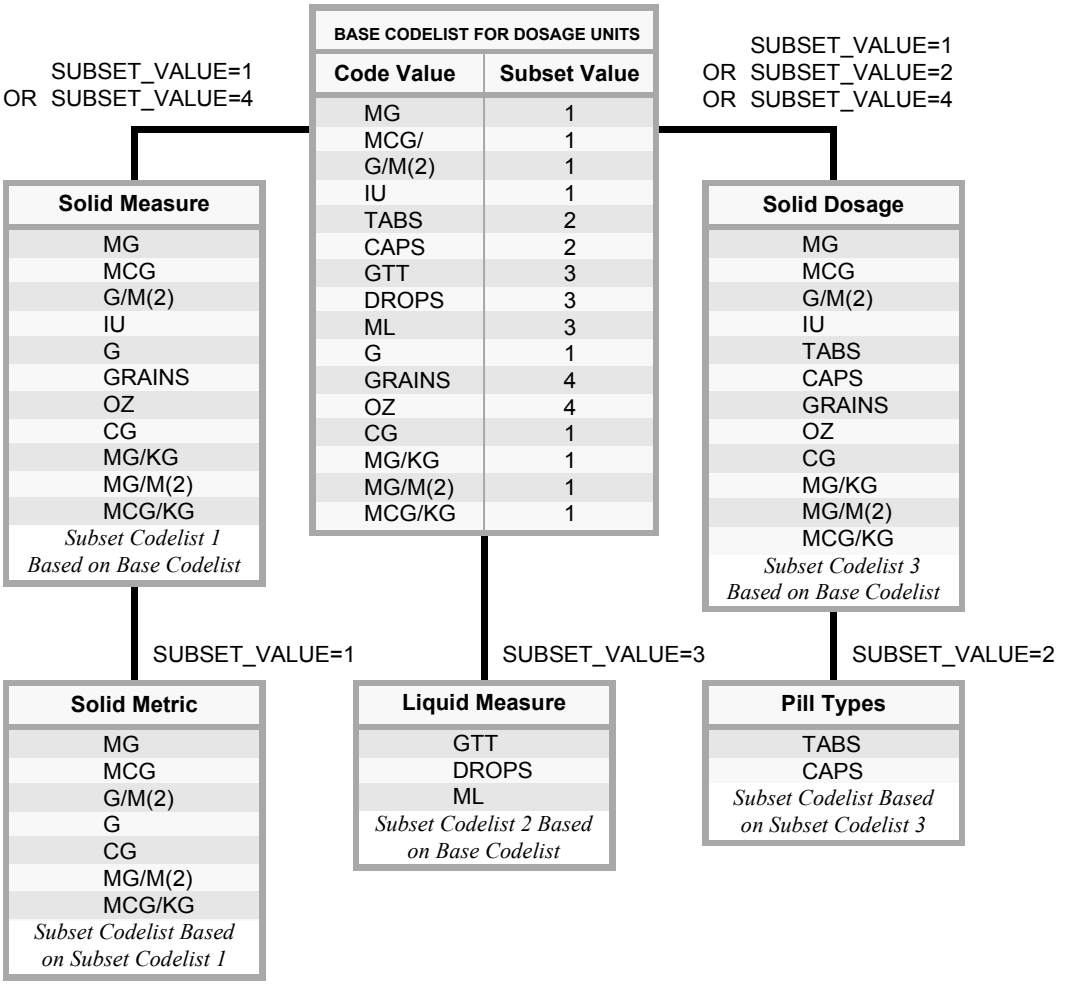
- For the Yes/No/Unknown subset codelist, either:

  VALUE='Yes' OR VALUE='No' OR VALUE='Unknown'

  SUBSET_VALUE<=3

  Note: To create a subset codelist containing all the values of the base codelist (as in this example), you can simply leave the subset restriction clause blank. The restriction clauses are provided here to illustrate the syntax.

Similarly, you can design subset codelists from base codelists with any number of values. You can also create multiple subset codelists based on a single base codelist, as in the following example:



As this illustration shows, you can create subset codelists (Solid Metric and Pill Types) based on other subset codelists.

Codelist attributes

List of codelist attributes

The following table lists the codelist attributes

Attribute:	Description:
Codelist	Name of the codelist.
SAS Name	A unique identifier used as the SAS format name. The SAS name must begin with a letter, contain only letters, numbers and the underscore character (_), and have six or fewer characters. The Clintrial software uses the first six letters of the codelist as the default SAS name. For more information on using this attribute to construct a name for an SAS format, see *Enter, Resolve, and Retrieve*.
Codelist Type	Aggregated — Stores the codelist in a database table with other aggregated codelists. If the codelist has a maximum of 100 rows, it should be aggregated. Unaggregated — Stores the codelist in a separate database table from other codelists. If the codelist has more than 100 rows, it should be unaggregated. View Codelist — A view onto a Type 0 panel or Oracle database table that defines clinical data tables to store data from an unaggregated codelist. Subset Codelist — An Oracle view onto a base codelist. A subset codelist functions as a subset of the base codelist, which can be any one of the four codelist types (Aggregated, Unaggregated, View, Subset).

Attribute:	Description:
Status	Codelist Status can have either of the following values: Valid (default) — The codelist can be used by the other Clintrial software modules. Invalid — An Invalid codelist cannot be referenced in a protocol and cannot be used in the other Clintrial software modules. You can specify an Invalid status for a codelist provided that no codelist references exist for that codelist and it is not the base codelist for a subset. A codelist with an Invalid status displays in the Codelist Browser. You can also work with an Invalid codelist using the **Codelist** menu's **Create**, **Modify**, **Delete**, and **Show** commands.
Dictionary	The Dictionary check box functions as follows: Cleared (default) — The codelist is not a data dictionary codelist. Checked — This defines the codelist as a data dictionary codelist. You typically use a data dictionary codelist as a base codelist for subset codelists. This type of codelist can be referred to only by objects in a protocol with the Dictionary attribute checked. Codelist import and codelist metadata distribution retain the value of this attribute. When modifying an existing codelist, you can specify this attribute only if no protocols containing data tables reference the codelist. You can clear this check box provided that there are no subset codelists based on this codelist.
View Created	For view and subset codelists, an indicator that the required Oracle view has been created.
Code Field Type	The code column in a codelist contains the codelist values to store in the codelist item in a panel. This attribute defines whether the code is stored as text (TEXT) or as a number (FIXED) in the codelist item. The codelist item in the panel must have the same data type as the code field type for the associated codelist. Text—Defines the data type of the code as TEXT. Fixed—Defines the data type of the code as FIXED. ***Note*:** All four columns in each codelist table are defined as TEXT.

Attribute:	Description:
Value Field Type	The value column in a codelist contains the codelist values to store in the checklist item in a panel. This attribute defines whether the value is stored as text (TEXT) or as a number (FIXED) in the checklist item. The checklist item in the panel must have the same data type as the value field type for the associated checklist. Text—Defines the data type of the value as TEXT. Fixed—Defines the data type of the value as FIXED. ***Note*:** All four columns in each codelist table are defined as TEXT.
Description	A description of the codelist.

List of read-only attributes

When you select the **Codelist** menu's **Show** command, the Clintrial software displays the previously described codelist attributes as read-only and also displays these additional read-only attributes:

Attribute:	Description:
Modification Date	Date of the most recent modification of the codelist.
Modified By	Name of the user account that created or most recently modified the codelist.
Codelist values (Order, Code, Value, Short Label, Long Label, Required, Subset Value)	Displays the columns for the codelist values.

List of view codelist attributes

The following table lists the additional view codelist attributes:

View codelist attribute:	Description:
Base Protocol	Name of the protocol or Oracle account containing the view base panel.

View codelist attribute:	Description:
Base Panel	Name of the panel that defines the clinical data tables to store the codelist data. The panel must be of the type Non-Patient Data (Type 0) or non-Clintrial Oracle table. The protocol can contain multiple view base panels, one for each codelist.
Code Item	The item in the view base panel or the column in the Oracle table that defines the column for the codelist's code data.
Value Item	The item in the view base panel that defines the column for the codelists's value data.
Label Item	The item in the view base panel that defines the column for the codelist's short label data.
Long Label Item	The item in the view base panel that defines the column for the codelist's long label data.
Status Item	The item in the view base panel that represents the codelist value's Status attribute.
Code Order Item	The item in the view base panel that represents the codelist value's Code Order attribute.
Subset Required Item	The item in the view base panel that represents the codelist value's Subset Required attribute.
Subset Value Item	The item in the view base panel that represents the codelist value's Subset Value attribute.
View Restriction	An optional SQL restriction clause that filters the codelist data in the view base panel based on a conditional statement.

List of codelist value columns

The following table lists the columns in the Modify Codelist Values dialog box for which you supply codelist values or specify the order of those values:

Column:	Description:
Order	The position of a codelist value in the list of codelist values. This position determines the order in which a Clintrial software user views codelist values, for example, in a drop-down list during data entry. To modify this order, from the **Edit** menu, select **Reorder Rows**. By default, this column has a null value.
Code	The data that is stored in a codelist item's column in the clinical data table. The code can include a maximum of 80 characters.
Value	The data that is stored in a checklist item's column in the clinical data table. (Optionally, data that is not in the checklist can be entered.) The value can include a maximum of 80 characters. You can specify multiple values for a single code, one in each row.
Short Label	An optional description of the code, with a maximum of 80 characters.
Long Label	An optional description of the code, with a maximum of 240 characters.
Required	If checked, the codelist value is required. A required value must be present in any subset codelist and is therefore included regardless of the subset specification used to define the subset codelist.
Subset Value	A number assigned to a subset value that can be used to specify that value in a subset restriction clause.

13 Flags and Notes

Overview

What are flags and notes?

Flags and *notes* are information about clinical data. In the Clintrial software, users can attach flags and notes to clinical data during such activities as data entry or verification.

Flags and notes are used primarily to label and monitor data quality problems, and to record information not accommodated by a CRF or the corresponding Clintrial software study page.

For example, suppose certain data on a CRF is illegible. When transcribing data from the CRF, the data-entry operator can flag the appropriate fields as illegible. After querying the data collection site and receiving a response, a data editor can add the clarified data, and remove the flag from those fields.

Or, suppose an investigator has made annotations in the margin of a CRF page. When transcribing data from the CRF, the data-entry operator can use a note to capture the annotations along with the clinical data.

Flags

What are flags?

A *flag* is an attachment to clinical data used to label and monitor data quality problems. For example, you might attach a flag to a number that is illegible, missing, or out of the expected range. In data-entry, you can attach a flag to an observation, a record, or an item.

There are two parts to a flag:

- A *flag category* characterizes the clinical data. Data-entry operators can attach a flag category.

  For example, suppose you want data-entry operators to be able to label situations in which data is either illegible, missing, or out of range. You could create the flag categories ILLEGIBLE, MISSING, and RANGE.

  Design supplies three flag categories: GENERAL, OVERRIDE and VERIFICATION. You can create as many additional flag categories as you require. Each flag category name must be unique.

- A *flag name* describes the action taken on the clinical data. Data-entry operators can attach a flag name within a flag category to an observation, a record, or a specific item.

  For example, suppose you want data-entry operators to be able to indicate not only that data is illegible, but also what action was, or should be, taken. Within the flag category ILLEGIBLE you could create the flag names ACCEPTED, QUERY, and TO_BE_QUERIED.

  The Clintrial software supplies four flag names, DOWNLOAD, UNSPECIFIED, AUTOFLAG, and PROPAGATED that are used in combination with the Clintrial software-supplied flag categories. You can create as many additional flag names within each flag category as required. Each flag name must be unique within its flag category.

For more information on how data-entry operators enter flag categories and flag names, see *Enter, Resolve, and Retrieve*.

List of Clintrial software-supplied flags

Design supplies six predefined flags (that is, six combinations of flag categories and flag names). The following table lists the Clintrial Design software-supplied flags:

Flag category:	Flag category description:	Flag name:	Flag name description:
GENERAL	General flags	UNSPECIFIED	Unspecified general flag
OVERRIDE	Override entry check flags	UNSPECIFIED	Unspecified override flag
OVERRIDE	Override entry check flags	AUTOFLAG	Automatic override flag
VERIFICATION	Verification flags	UNSPECIFIED	Unspecified verification flag
VERIFICATION	Verification flags	AUTOFLAG	Automatic verification flag
VERIFICATION	Verification flags	PROPAGATED	Automatic propagated flag

Note: The flags below for extended modules are available only if the extended module is installed on the server.

Classify supplies one predefined flag. The following table lists the Clintrial Classify software-supplied flag:

Flag category:	Flag category description:	Flag name:	Flag name description:
CLASSIFY	Classify flags	REQUEST	Request for more information

Lab Loader supplies two predefined flags (that is, two combinations of flag categories and flag names). The following table lists the Clintrial Lab Loader software-supplied flags:

Flag category:	Flag category description:	Flag name:	Flag name description:
LABLOADER	Lab Loader flags	DUPLICATE	Duplicate detected
LABLOADER	Lab Loader flags	TRANSFER_ FAIL	Record failed to transfer

You cannot delete from these sets of Clintrial software-supplied flags.

List of flag attributes

To create a flag, with the Flag Browser open, from the **Flag** menu, select the **Create** command. The Create Flag dialog box opens, in which you can specify flag category and flag name attributes. The following table lists the flag attributes displayed in the Create Flag dialog box:

Flag attributes:	Description:
Category	Name of the flag category.
Description	Description of the flag category.
Flag Name	The flag name.

Flag attributes:	Description:
Description	Description of the flag name.

When you select the **Flag** menu's **Show** command, the Show Flag window opens. This window displays the previously described flag category and flag name attributes as read-only, and also displays the following additional read-only attributes:

Read-only attributes:	Description:
Status	OK or Deleted — Status is always OK unless deletion is in progress. Deleted takes time, because the data is checked to see if a flag is in use.
Modifiable Status	Modifiable if the flag can be modified; Not Modifiable if the flag cannot be modified.
Database ID	Registration number of the database instance.

Notes

What are notes?

A *note* is an annotation about clinical data, made by a sponsor or the investigator. There are two types of notes: sponsor notes and investigator notes.

There are two parts to a note:

- A *note category* characterizes the source of the note — the category of person who annotates the clinical data.

  Design supplies two note categories — INVESTIGATOR and SPONSOR. You cannot create additional note categories or delete the note categories.

- A *note name* further characterizes the source of the note — the sub-category of person who annotates the clinical data.

  Design supplies one note name for each note category — UNSPECIFIED.

List of Clintrial software-supplied notes

Design contains two predefined notes (that is, two combinations of note categories and note names). The following table lists the Clintrial software-supplied notes:

Note category:	Note category description:	Note name:	Note name description:
SPONSOR	An annotation that your company makes about the processing of clinical data	UNSPECIFIED	Unspecified sponsor notes
INVESTIGATOR	An annotation that the physician makes on the CRF about clinical data	UNSPECIFIED	Unspecified investigator notes

Although you cannot create additional note categories, you can create additional note names for the SPONSOR note category. However, the INVESTIGATOR note category always has the UNSPECIFIED note name, and you cannot create additional note names for this category.

You cannot delete from this list of notes.

List of note attributes

To create additional note names for the SPONSOR note category, with the Note Browser open, from the **Note** menu, select **Create**. The Create Note dialog box opens, in which you can specify the following note attributes:

Note attribute:	Description:
Category (read-only)	SPONSOR — The Clintrial software-supplied SPONSOR note category is the only note category for which you can create a note name.
Note Name	The note name.
Description	Description of the note name.

When you select the **Note** menu's **Show** command, the Show Note window opens. This window displays previously described note category and note name attributes as read-only, and also displays the following additional read-only attributes:

Read-only attributes:	Description:
Status	OK or Deleted — Status is always OK unless deletion is in progress. The Delete status takes time to be displayed, because the data is checked to determine if a flag is in use.
Modifiable Status	Modifiable if the note can be modified, Not Modifiable if the note cannot be modified.
Tag ID	Internal, unique identifier for the flag or note.
Database ID	Registration number of the database instance.

Summary of flag and note use

How to use flags and notes

Users' access rights to clinical data in the protocol determine the activities that they can perform with flags and notes. Depending on their access privileges, they may perform the following activities. These activities include:

- Attach flags and notes to data when entering, verifying, or editing data.
- Display, modify, or remove notes when entering, verifying, or editing, data.
- Display flags and notes when displaying data.

For more information on the use of flags and notes, see *Enter, Resolve, and Retrieve*.

List of flag- or note-based restrictions

In most cases where a restriction clause can be used to access records (for example, during verification, editing, or validation), a flag or note restriction can also be used. A *flag restriction* or a *note restriction* enables the user to request only the records to which specified flags or notes are either attached or not attached.

For example, a user might want to:

- Edit only records to which the flag ILLEGIBLE/TO_BE_QUERIED is attached.
- Edit only records to which a sponsor note with the note name MONITOR is attached.
- Read only records to which any flag in the flag category MISSING is attached.

For more information on flag and note restrictions, see *Enter, Resolve, and Retrieve*, and the *Reference Guide*.

How flags work with verification

The flag VERIFICATION/AUTOFLAG is attached automatically to an item during blind verification when newly entered data conflicts with existing data in the database. During either interactive or blind verification, only flags with the category VERIFICATION can be manually attached to data. These flags are not available during data entry or editing.

For more information on verification, see the *Clintrial 4.7 Enter, Resolve, and Retrieve* manual.

How flags work with validation

A rule that is created for use during data validation can be set up so that a specified flag is attached automatically to a record for which the rule evaluates to FALSE.

For more information on rules, see Chapter 11.

For more information on validation, see *Manage, Classify, and Lab Loader.*

14 Coding Thesauruses

Overview

In order to match clinical data (verbatim text) against standard dictionary codes, you may either:

- Use Clintrial Manage, and optionally, the Clintrial Classify extended module.
- Use the Oracle Central Coding application.

The decision to use either application is made on a protocol basis. For example, existing protocols may continue to use the Classify module for coding, while new protocols may use Central Coding. Of course, existing protocols may be updated to use Central Coding if you want to use it as a standard within your organization.

What is a coding thesaurus?

A *coding thesaurus* is a dictionary thesaurus that contains standard codes for a particular type of clinical data. There are two types of thesauruses: industry-standard and user-defined.

Examples of coding thesauruses that work with the Clintrial software are:

- COSTART — The industry-standard Coding Symbols for a Thesaurus of Adverse Reaction Terminology dictionary.
- WHOART — The World Health Organization Adverse Reaction Terminology dictionary that contains standard codes for clinical events.
- WHODRL — The World Health Organization Drug Reference List dictionary that contains standard codes for drugs.
- ICD-9-CM — The International Classification of Diseases, 9th Revision, Clinical Modification dictionary that contains standard codes for diseases.
- MedDRA — The Medical Dictionary for Regulatory Activities, a mixed-case, hierarchical thesaurus that contains terminology applicable to all phases of drug development, and to the health effects of devices.

You can also create a user-defined thesaurus for your study, and add synonyms to a standard coding thesaurus.

Why use coding thesauruses?

You use coding thesauruses to:

- Provide standardization for statistical analysis.
- Summarize terms by grouping verbatim terms into standardized terms.
- Create a standard language that is comparable across all therapeutic teams.
- Reduce the time needed to reach a final format, which leads to faster regulatory approval.

Steps in setting up coding thesauruses

To set up the Clintrial software to use a coding thesaurus, you must do the following:

1. Have available the coding thesaurus data files.
2. Create a coding thesaurus protocol.

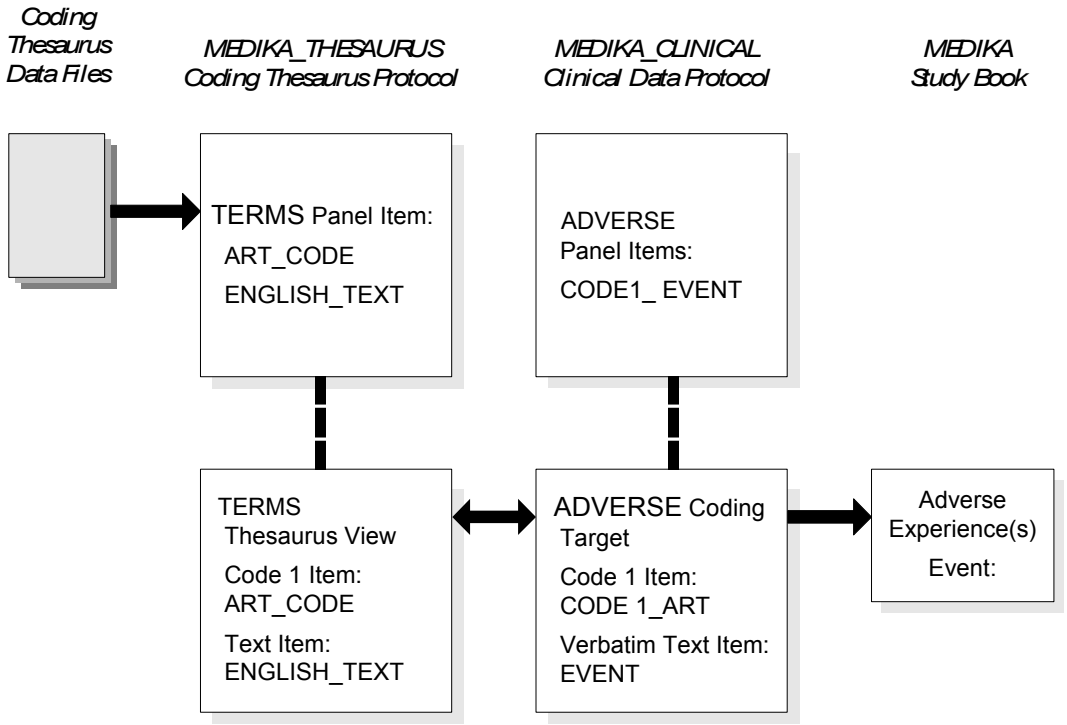
   After creating the coding thesaurus protocol, you must:

   a. Create specialized panels and items in the thesaurus protocol to receive the coding thesaurus data.

   b. Create thesaurus languages for each national language that you want the thesaurus to support.

   c. Create thesaurus views for each panel, or create views of non-Clintrial panels.
3. Load the data from the coding thesaurus data files into the appropriate panels of the coding thesaurus protocol.
4. Create the panel item in your clinical data protocol that the data-entry operator uses to enter the data to be coded, as well as additional items to contain the associated coding thesaurus data that handles the coding.
5. In the clinical data protocol, associate the verbatim text item and a set of specially created items (the coding target) with the corresponding items in the thesaurus protocol (identified by thesaurus views).

The following figure shows how a coding thesaurus relates to the sample protocols — the ART_THESAURUS coding thesaurus protocol and the MEDIKA_CLINICAL clinical data protocol containing the verbatim text item to be coded. Refer to this figure and its accompanying explanation while reading the remainder of this chapter.



In this example:

1. The designer created a TERMS panel and items needed for coding in the TERMS panel in ART_THESAURUS.
2. The designer created the TERMS thesaurus view, based on the TERMS panel. (This assumes the required thesaurus languages were already created.)
3. The designer loaded the coding thesaurus data to the TERMS panel in the ART_THESAURUS coding thesaurus protocol.
4. The designer created the verbatim text item, as well as other items, to serve as the coding target in the ADV panel in the MEDIKA_CLINICAL clinical data protocol.
5. The designer specified the ART_THESAURUS protocol as the value of the CODE1_ART item's Thesaurus attribute in the ADV panel.
6. On the basis of Step 5, the Clintrial software identified the CODE1_ART item as the Code 1 coding target item.

7. The designer specified other items in the ADV panel as other coding targets (for example, the verbatim text item EVENT is the Verbatim Text Item coding target).
8. When creating the page section layout, the designer deleted the Code 1 coding target item (and other coding target items except for the verbatim text item) from the page section layout because they are not necessary for the data-entry operator.
9. The data-entry operator can enter verbatim text in the Event field in the study book.

For information on thesaurus views, see "What is a thesaurus view on a Clintrial software panel?" on page 322.

For information on coding targets, see "Setting up coding targets" on page 330.

How automatic coding relates to interactive coding

At data entry, the data-entry operator enters the verbatim text for the item to be coded. Manage provides two methods of determining the correct code for an entered item that has been set up as a coded item:

- Automatic coding

  If a single match is found on the verbatim text using the specified thesaurus algorithm, the Clintrial software selects a code for the verbatim text item to be coded. The Clintrial software supplies a default thesaurus algorithm, or you can create your own customized thesaurus algorithms.
- Interactive coding

  The data manager selects a code from a scrolling list of similar matching terms, each of which has an associated code.

The data manager performs automatic coding first. Then the results are reviewed and corrected with interactive coding.

For information on interactive coding and automatic coding, see *Manage, Classify, and Lab Loader*.

For information on thesaurus algorithms, see "Thesaurus algorithms" on page 337.

Setting up a coding thesaurus protocol

How to create a coding thesaurus protocol

To create a coding thesaurus protocol, create a protocol with the Type attribute value of Coding Thesaurus.

Note: Oracle's standard drug dictionary metadata supports only **WHODD** format **B2**. A custom dictionary may be created to support any format.

For information on protocols, see Chapter 9.

List of panels in a coding thesaurus protocol

A coding thesaurus protocol that uses the Clintrial software default thesaurus algorithm must have the following three panels:

- A *terms panel* contains the terms and corresponding codes that are loaded from the coding thesaurus to the thesaurus protocol.
- A *synonyms panel* contains synonyms for the terms in the terms panel. Some coding thesauruses, such as COSTART, already contain some synonyms. Typically, additional synonyms are added to increase the success rate of automatic coding. Use Enter to add synonyms.
- A *stopwords panel* contains the words that should be discarded from the verbatim text as not adding meaningful information when running the automatic thesaurus algorithm.

The Clintrial software-supplied thesaurus algorithm uses views on a terms panel, synonyms panel, and a stopwords panel. Therefore, you must create these three panels if you plan to use the Clintrial software-supplied algorithm. However, if you create a custom thesaurus algorithm, you can create whatever panels are necessary to do your coding. Although you would always need at least one terms panel, you could create a thesaurus algorithm that does not require a synonyms or stopwords panel, as does the Clintrial software-supplied algorithm.

For example, the DRUG_THESAURUS sample coding thesauruses protocol contains three terms panels: DRUG TERMS (for an industry-standard drug dictionary), DRUG_TERMS_MEDIKA (for drugs under development for the Medika Corporation), and ALL_DRUG_TERMS
(a combination of the industry and Medika drugs).

For more information on thesaurus algorithms, see "Thesaurus algorithms" on page 337.

Example: the CT_MEDDRA thesaurus protocol

What is the CT_MEDDRA protocol?

The CT_MEDDRA protocol is a Clintrial software-supplied thesaurus protocol that provides Clintrial software support for a single-language coding using the MedDRA dictionary. You can enhance CT_MEDDRA to meet your coding needs, for example, coding in multiple languages.

Note: The CT_MEDDRA thesaurus protocol does not contain MedDRA dictionary data. You must purchase the MedDRA dictionary and batch load or interactively enter the MedDRA data into the appropriate panels. However, the Clintrial software includes sample control files that can be used to batch load the MedDRA dictionary data.

Setting up the CT_MEDDRA protocol

To set up the CT_MEDDRA thesaurus protocol:

1. Import the CT_MEDDRA protocol.

   The metadata import file is:

   ct_meddra_m.dmp

   It is strongly recommended that you name the imported protocol CT_MEDDRA. The CT_MEDDRA protocol is used by name in the following:

   - Coding targets for the MEDIKA_CLINICAL sample clinical data protocol.
   - The MedDRA-related functions used by the MedDRA-related Retrieve queries and available for use in SELECT statements in derivations or in the SQL Tools.
2. Create the clinical data tables, either at metadata import or by installing the panels after import is complete.
3. Optionally, log in to the CTPROC account in SQL*Plus and execute the following:

   SQL> EXECUTE ct_meddra_util.create_meddra_indexes(‘CT_MEDDRA’)

   SQL>EXECUTE ct_meddra_util.create_meddra_functions(‘CT_MEDDRA’)
4. Optionally, modify the CT_MEDDRA protocol to meet your particular needs.

When doing so, do not:

5. Delete the Clintrial software-supplied panels in CT_MEDDRA.
6. Modify or delete the items within these panels.

   You can add additional items. However, if you add items, you must update the Clintrial software-supplied page sections, and you must modify the Clintrial software-supplied sample control files.

To complete the setup of the CT_MEDDRA protocol, you can use other Clintrial software modules optionally to do the following:

- Manage — Batch load MedDRA dictionary data.
- Enter — Insert, update, and delete data from the CT_MEDDRA protocol.

For a description of the panels, views, and algorithms for CT_MEDDRA, and for the MedDRA-related functions, see the *Reference Guide.*

For information on the Medika Sample Study see *Getting Started*.

For information on the Clintrial software-supplied MedDRA-related Retrieve queries, see the Retrieve section of *Enter, Resolve, and Retrieve*.

For information on batch loading, see the Manage section of *Manage, Classify and Lab Loader.*

List of items in a terms panel

A terms panel must be a Type 0 panel (because it does not contain subject data) and contain the following items:

- Items (one for each language) to contain language-specific text for terms

  This item is used for the match against the clinical data verbatim text.
- An item to contain the code that corresponds to the term in the coding thesaurus
- Additional items for each part of a multipart code, if you plan to use multipart codes

You can optionally add other items to the terms panel.

How to create code items

You must create an item to contain the code that corresponds to the term in the coding thesaurus.

Note: If the length of a code item exceeds 240 bytes, the codes are truncated to 240 bytes during the coding in Manage. Design will allow a definition of more than 240 bytes, but you should limit the length of the item to the Manage limit.

For example, if you always want “headache” to code to “12345”, set up the record in the terms panel so that the term field contains “headache” as the code field contains “12345”.

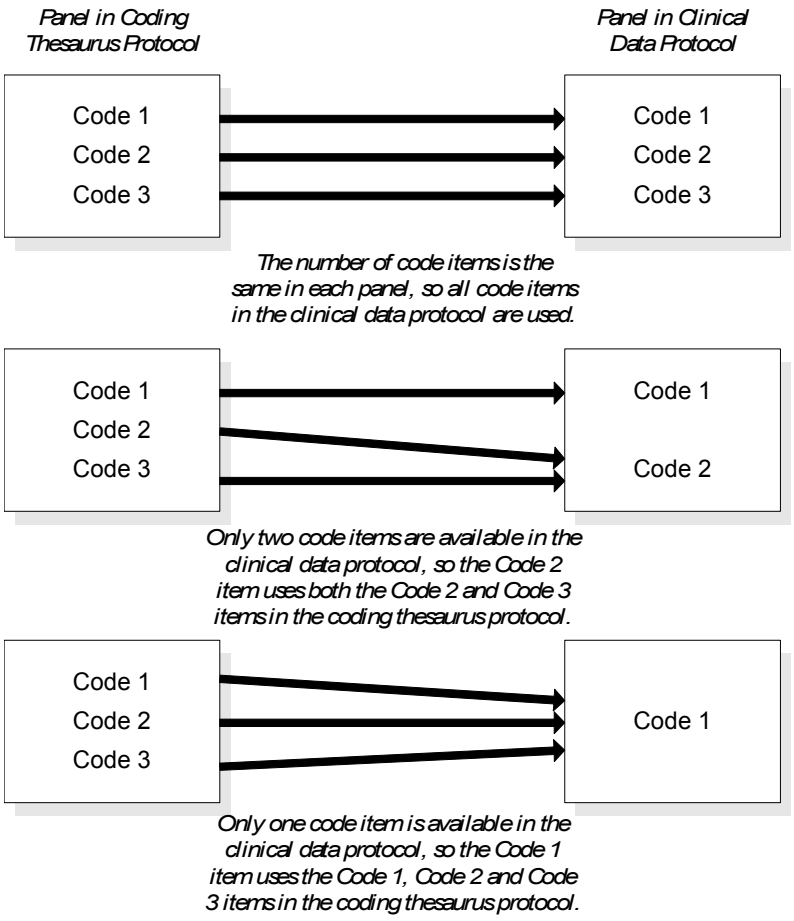
If the coding thesaurus contains multipart codes and you want to store the code in separate items, you must create an additional item for each part of the code (up to a maximum of three code items). For example, the WHOART dictionary term “ECZEMA” might have the three-part code 0012.001, with the value 0012 stored in the Code 1 item, 001 in the Code 2 item, and DERMATOLOGY in the Code 3 item.

Even if the panel in the coding thesaurus protocol stores multipart codes, you can code data in a clinical data panel to a single code. The Clintrial software concatenates a multipart code from a coding thesaurus protocol’s panel to a clinical data panel by detecting how many code items you define in the clinical data panel that contains the verbatim text item, and then expressing the code accordingly within a single code item.

Note: The Clintrial software concatenates a multipart code from a panel in a coding thesaurus protocol to a clinical data panel in one of the following ways:

- If there are the same number of code items in the coding thesaurus panel and the clinical data panel, then the Clintrial software considers the codes to have a one-to-one relationship.
- If there are three code items in the coding thesaurus panel and two in the clinical data panel, then the Clintrial software codes the Code 1 items in a one-to-one relationship, but concatenates data from the Code 2 and Code 3 items for use in the Code 2 item in the clinical data panel.
- If there are three code items in the coding thesaurus panel and three in the clinical data panel, then the Clintrial software codes the data from the three code items in a one-to-one relationship with the Code 1, Code 2, and Code 3 items in the clinical data panel.

The following figure shows how multiple code items in a coding thesaurus panel relate to code items in a clinical data protocol:



How to create language items

You must create items (one for each language) to contain language-specific text for terms.

The term can be the preferred term or an included term. A *preferred term* and an *included term* are both text descriptions, each associated with a distinct code. Typically, an included term is a more specific text description of the preferred

term. For example, in the WHOART dictionary, the preferred term might be "ECZEMA" and an equivalent included term might be "ECZEMA ALLERGIC" (which might have the three-part code 0012.003.DERMATOLOGY.

You must create one language-specific text item for each language translation that you plan to load. The item name should specify the language. One language-specific text item must be set up to include the literal term ENGLISH. Items for other languages are optional.

Note: You must create a corresponding thesaurus language for each language-specific text item. The language-specific text item name must contain the name of the thesaurus language. The item can also have a prefix and a suffix, which must be the same for all language-specific text items.

You must create the language-specific text item with the literal term ENGLISH, even if you do not plan to use English terms. However, if you are not coding in English, you do not need to create a corresponding English thesaurus language.

You can create the thesaurus languages before or after creating the coding thesaurus panels (but you must create them before creating the thesaurus views).

For example, if the coding thesaurus supports English, French, German, and Spanish coding, you must create four thesaurus languages and four language-specific text items.

The thesaurus languages could be:

ENGLISH
FRENCH
GERMAN
SPANISH

The language-specific text items could be named:

AES_ENGLISH_TERM
AES_FRENCH_TERM
AES_GERMAN_TERM
AES_SPANISH_TERM

For information about thesaurus languages, see "What is a thesaurus language?" on page 321.

Example: the ART_THESAURUS TERMS panel

You must create additional items for each part of a multipart code, if you plan to use multipart codes.

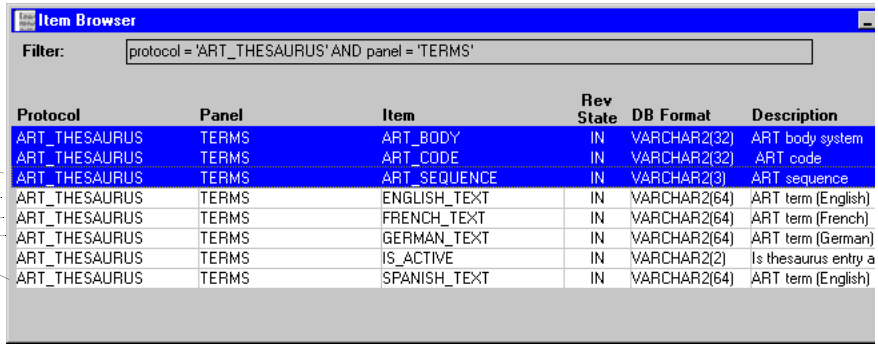
You can optionally add other items to the terms panel.

When you create and install a terms panel, the Clintrial software automatically creates a page section based on that panel, and a page template and study book, each with the same name as the panel.

The following example shows the items in the TERMS panel of the ART_THESAURUS coding thesaurus protocol:

Code 3 item
Code 1 item
Code 2 item

Language-specific test items



List of items in a synonyms panel

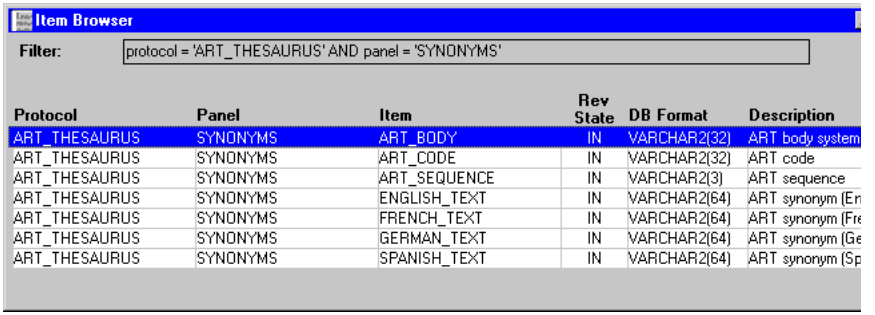
The synonyms panel must be a Type 0 panel. A synonyms panel has the same item requirements as a terms panel, including the naming conventions for multiple language-specific text items, as described in the previous section. Within any algorithm, the terms and synonyms items must match both in number and in data type.

In a synonyms panel, the text item stores the text of a synonym for a preferred term or an included term from the coding dictionary, rather than the preferred term or included term itself. Each synonym is associated with a code for the preferred term or the included term.

Some coding dictionaries provide synonyms, others do not. In either case, you can add synonyms manually in Enter or by batch loading in Manage.

When you create and install a synonyms panel, the Clintrial software automatically creates a page section based on that panel, and a page template and study book, each with the same name as the panel.

The following example shows the items in the SYNONYMS panel of the ART_THESAURUS coding thesaurus protocol:



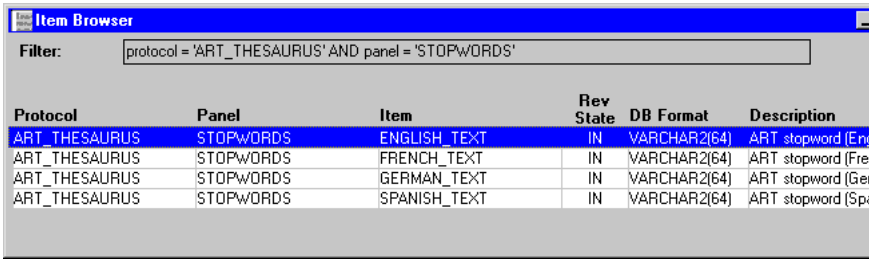
List of items in a stopwords panel

A stopwords panel must be a Type 0 panel. A stopwords panel must contain language-specific text items (one for each language) to store the stopwords text. However, the items in a stopwords panel are not required to be the same in number and data type as those in the terms and synonyms panels.

The default thesaurus algorithm that the Clintrial software uses during automatic coding includes a step in which words that do not add any meaning are dropped from the verbatim text that is entered. Typical English stopwords are articles and prepositions, such as “the,” “of,” and “in” and nonmedical words typically found in clinical event descriptions, such as “patient” and “complained.”

When you create and install a stopwords panel, the Clintrial software automatically creates a page section based on that panel, and a page template and study book, each with the same name as the panel.

The following example shows the items in the STOPWORDS panel of the ART_THESAURUS coding thesaurus protocol:



What is a thesaurus language?

A *thesaurus language* is a language name that determines which language-specific text items in the terms, synonyms, and stopwords panels are used for coding. The Clintrial software uses the language name to select a corresponding language-specific text item that contains the language name.

You must create thesaurus languages before creating thesaurus views.

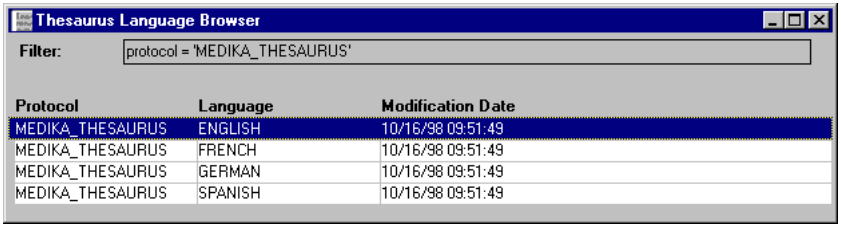
If the coding thesaurus supports multiple languages, you must create a thesaurus language for each language that is supported. Even if the coding thesaurus supports only one language, you must create a thesaurus language for that language.

You must create a thesaurus language for each corresponding language-specific text item you plan to use. The language-specific text item names must contain the name of the thesaurus language. If you create a language-specific text item but not a corresponding thesaurus language, the language-specific text item is not used.

Although you must create a language-specific text item whose name contains the literal term ENGLISH, you do not need to create a corresponding ENGLISH thesaurus language. If you do not create a corresponding ENGLISH thesaurus language, the language-specific text item containing the literal term ENGLISH cannot be used.

When you create or modify a thesaurus language, you can select the punctuation characters that you want removed from the verbatim text during automatic coding that uses that language.

To create a thesaurus language, from the **Protocol** menu, select **Thesaurus Languages**. The following example shows the thesaurus languages in the MEDIKA_THESAURUS coding thesaurus protocol:



For more information on the language-specific text item, see "How to create language items" on page 317.

What is a thesaurus view on a Clintrial software panel?

A *thesaurus view* is an Oracle view onto a panel in the coding thesaurus protocol or on a non-Clintrial table. The thesaurus view identifies items that correspond to the *coding target* items in the panel of the clinical data protocol that contains the verbatim text item to be coded.

To create thesaurus views, from the **Protocol** menu, select **Thesaurus Views**. The Clintrial software-supplied default thesaurus algorithm uses views with the following names:

- The TERMS view — A thesaurus view that identifies the items in the terms base panel that correspond to the coding target items in the clinical data protocol.

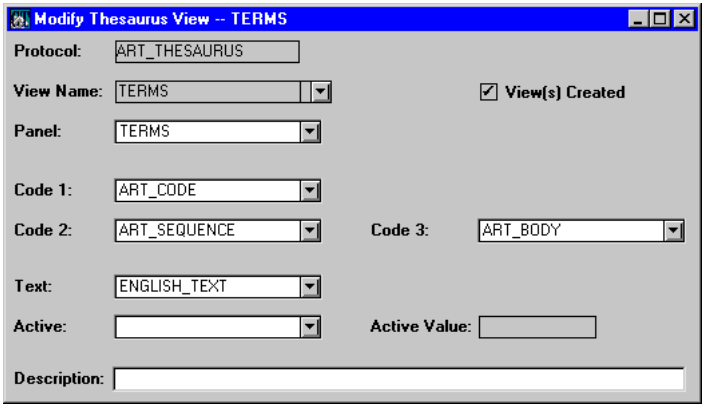
  Use the Code 1 and Text required attributes in the TERMS thesaurus view to identify the items in the coding thesaurus protocol that store the codes and any preferred or included terms. (Use Code 2 and Code 3 for multipart codes.)

  Design creates a TERMS view for each language, based on the language-specific text items that you created. Recall that when you created the language-specific text items, you incorporated the thesaurus language name in the item name. In Manage, the Clintrial software codes to the language specified in the language-specific text item you specified in the terms panel.
- The SYNONYMS view — A thesaurus view that identifies the items in the synonyms base panel that correspond to the coding target items in the clinical data protocol.

  The default Clintrial software-supplied name for the synonyms thesaurus view is SYNONYMS. The SYNONYMS view has the same requirements as the terms view, and must contain the same items with the same data types. Design creates a SYNONYMS view for each language. The TERMS and SYNONYMS views must match in number of items and data types, because they are used in the Clintrial software-supplied thesaurus algorithm.
- The STOPWORDS view —A thesaurus view that identifies the item in the stopwords base panel that stores words to be filtered from the verbatim text string.

  Use the Text attribute in the STOPWORDS thesaurus view to select the base panel item that stores the stopwords to be dropped by the thesaurus algorithm. If you are using more than one language, select the English thesaurus language item for the Text attribute. Design creates a STOPWORDS view for each language.

The following example shows the TERMS thesaurus view, set up for three-part coding in the ART_THESAURUS coding thesaurus protocol:



If you create a custom thesaurus algorithm, you can create views on any panels and you can assign them names other than those required for use with the default thesaurus algorithm.

For information on creating coding target items in the clinical data protocol, see "Setting up coded panels in a clinical data protocol" on page 327 and "Setting up coding targets" on page 330.

For information on the relationship of the items identified by the thesaurus view to the items identified as the coding targets, see the figure that shows how a coding thesaurus relates to the sample protocols.

How to create a view on a non-Clintrial table

You can also create a thesaurus view on a non-Clintrial table, rather than on a Clintrial software panel. To create a thesaurus view on a non-Clintrial table, do the following:

1. Create an Oracle account in which to create the table.

   The Oracle account must be an account outside of the Clintrial software; it cannot be a protocol account or a Clintrial system account.

2. In the Oracle account, create a table with columns that you will map to the thesaurus view.

   If the thesaurus protocol has more than one language defined, you must create columns with the language names, as you would for views on panels. That is, the table in the thesaurus view must have:

   – A verbatim text column that includes the word ENGLISH

– Additional columns named for each language you defined

Note: If the thesaurus protocol has only one language defined, you do not need to have a verbatim text column that contains the word ENGLISH, nor do you need to follow any particular naming convention for the defined language. The naming restriction remains for a panel-based thesaurus view.

3. Grant select privileges (WITH GRANT OPTION) on that table to the thesaurus protocol account before you create the thesaurus view, as follows:

SQL> **GRANT SELECT ON *account.table_name* TO *thesaurus_protocol_name* WITH GRANT OPTION;**

How to use additional views

The default thesaurus algorithm supplied by the Clintrial software uses both the panels which you must create for terms, synonyms, and stopwords, and the thesaurus views on those panels. If you want to create customized thesaurus algorithms, you can also create additional thesaurus views that are based on the panels that store the data to be coded.

For information on the default thesaurus algorithm, and on creating a customized thesaurus algorithm, see "Thesaurus algorithms" on page 337.

For information on panels for coding thesaurus protocols, see "List of panels in a coding thesaurus protocol" on page 313.

How to recreate thesaurus views

In most cases, the Clintrial software automatically updates (recreates) thesaurus views when you create, modify, or delete a thesaurus language.

You may occasionally need to recreate thesaurus views in the following cases:

- The thesaurus view is based on a non-Clintrial database table.
- The view is a copy or a distributed view.
- The view is outdated.

To recreate a view, in the Thesaurus View Browser, select the view you want to recreate, and then, from the **Thesaurus View** menu, select **Compile**. The Clintrial software recreates (updates) the view.

List of thesaurus view attributes

The following table lists the thesaurus view attributes:

Attribute:	Description:
Protocol	Name of the base thesaurus protocol.
View Name	Name of the view—One of the following (required if using the default algorithm) or a name of your choosing: TERM — View panel for the terms base panel SYNONYMS — View panel for the synonyms base panel STOPWORDS — View panel for the stopwords base panel
Panel	Name of the base panel in the coding thesaurus protocol, if the thesaurus view is based on a Clintrial software panel.
Account	Name of the Oracle account containing the base table, if the thesaurus view is based on a non-Clintrial table.
Table Name	Name of the Oracle table that is the base table, if the thesaurus view is based on a non-Clintrial table.
Code 1	Name of the base panel item that stores the code (or, if a multipart code, the first part of the code). This item must have a data type of TEXT.
Code 2	For a two- or three-part code, the name of the base panel item that stores the second part of the code. This item must have a data type of TEXT.
Code 3	For a three-part code, the name of the base panel item that stores the third and last part of a three-part code. This item must have a data type of TEXT.
Text	Name of the item that stores the terms for the language used by this view. This item must have a data type of TEXT.
Active	Optionally, the name of a base panel item that indicates which rows in the coding thesaurus are to be used (active). This item must have a data type of FIXED or TEXT.

Attribute:	Description:
Active Value	If the Active attribute is in use, the value that indicates that a row is active. For example, a value of 1 indicates that rows in the base panel item with a value of 1 are active. The value must be an integer.
Description	Description of the thesaurus view.

List of read-only attributes

When you select the **Thesaurus View** menu's **Show** command, the Clintrial software displays the previously described thesaurus view attributes as read-only, and also displays these additional read-only attributes:

Attribute:	Description:
View(s) Created	Checked if the view was created successfully.
Modification Date	Date of the most recent modification of the thesaurus view.
Modified By	Name of the user account that created or most recently modified the thesaurus view.
Modifiable Status	Modifiable if the thesaurus view can be modified, Not Modifiable if the view cannot be modified.

Batch loading to the thesaurus protocol

Steps in loading panel data

After you have created and installed the required panels in the coding thesaurus protocol, you can load data from the coding thesaurus data files. There are two ways to load coding thesaurus data to your coding thesaurus protocol:

- Use the batch-loading facility in Manage to create a control file.

  Or, you can create the control file and run SQL*Loader outside of the Clintrial software. However, you still must do the batch-loading post-processing in Manage (screening).
- Use Enter to enter data into the coding thesaurus protocol.

  When you create and install a panel (terms, synonyms, or stopwords), The Clintrial software creates a page section, page template, and study book for each panel. Because coding thesaurus protocols use only Type 0 panels (non-subject data), there is no enrollment panel or context panel. Open the study book and enter the non-subject data.

After entering the required data, merge all data to the data table for the panel. Only data in the data table is available for use in coding.

For more information on batch loading, see *Manage, Classify, and Lab Loader*.

For more information on panel types, see "Panel types" on page 193.

For more information on entering non-subject data, see the Clintrial 4.7 *Enter, Resolve, and Retrieve* manual.

Setting up coded panels in a clinical data protocol

List of items in a clinical data panel that is coded

If you want a coding thesaurus protocol to supply the codes for a verbatim text item in the clinical data panel, you must create that item, as well as at least two other required items, to serve as coding targets. You can also create optional items to store other information related to the use of a coding thesaurus. The following tables list the required and optional items:

Required items to store:	Database format:	Description:
Code	VARCHAR2(*n*), sufficient to store the code	Unique code from the coding thesaurus for the term. Set the Thesaurus item attribute for this item only.

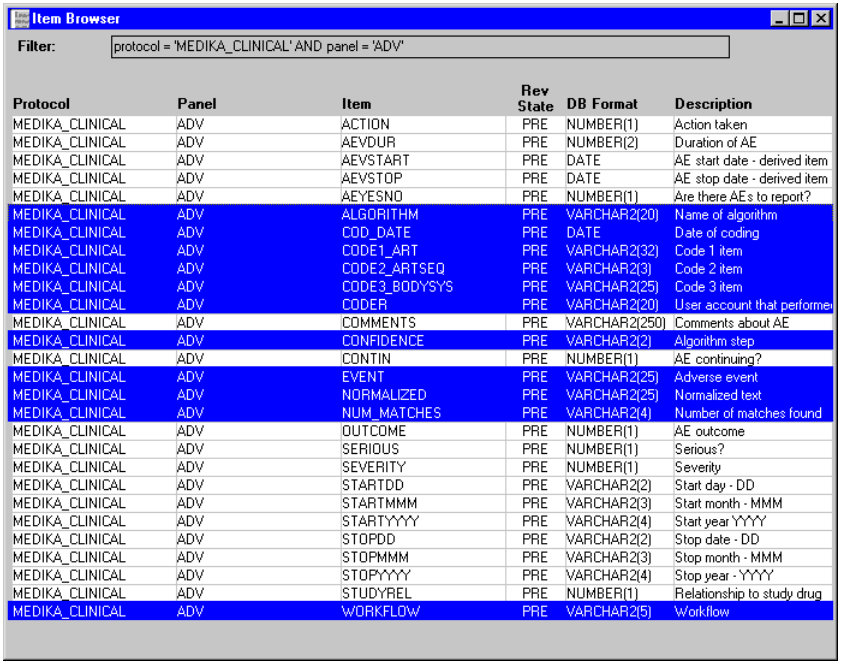
Required items to store:	Database format:	Description:
Term to be coded	VARCHAR2(*n)*	For the verbatim text input. Do not set the Thesaurus item attribute for this or any of the following items listed in these two tables.
Workflow step that coded the item	VARCHAR2(5)	Manage sets the value as either AUTO, INT, or FAIL during the coding process.

Optional items to store:	Database format:	Description:
Name of user	VARCHAR2(20)	Name of the user account that performed the interactive or automatic coding.
Date the item was coded	DATETIME	The date the item was coded.
Confidence level of coding	VARCHAR2(1) or larger	Step of the thesaurus algorithm that produced a match during automatic coding, or one more than the total number of steps in the thesaurus algorithm if no code is found.
Number of matches found by automatic coding	VARCHAR2(4)	The actual count, if Manage system parameter AUTOCODE_FULL_CNT is set to Yes. >1, if Manage system parameter AUTOCODE_FULL_CNT is set to No.
Second part of a multipart code	VARCHAR2(*n)*	If the code from the coding thesaurus is a two-part code, you can create a second code item. (Do not set the Thesaurus attribute on this item.)

Optional items to store:	Database format:	Description:
Third part of a multipart code	VARCHAR2(*n)*	If the code from the coding thesaurus is a three-part code, you can create a third code item. (Do not set the Thesaurus attribute on this item.)
Normalized text	VARCHAR2(*n*)	This optional field may be filled in during the autocode process. If so, it contains the text that remains after the verbatim text is normalized. This step converts the text to uppercase and eliminates extra white space and user-defined punctuation characters.

Note: It is recommended that you create no more than *five* coded items in any single panel. If you define more items, there is a possibility that an Oracle error will occur when you try to install the panel.

The following example shows the items in the ADV panel of the MEDIKA_CLINICAL clinical data protocol with the coding target items highlighted:



Protocol	Panel	Item	Rev State	DB Format	Description
MEDIKA_CLINICAL	ADV	ACTION	PRE	NUMBER(1)	Action taken
MEDIKA_CLINICAL	ADV	AEVDUR	PRE	NUMBER(2)	Duration of AE
MEDIKA_CLINICAL	ADV	AEVSTART	PRE	DATE	AE start date - derived item
MEDIKA_CLINICAL	ADV	AEVSTOP	PRE	DATE	AE stop date - derived item
MEDIKA_CLINICAL	ADV	AEYESNO	PRE	NUMBER(1)	Are there AEs to report?
MEDIKA_CLINICAL	ADV	ALGORITHM	PRE	VARCHAR2(20)	Name of algorithm
MEDIKA_CLINICAL	ADV	COD_DATE	PRE	DATE	Date of coding
MEDIKA_CLINICAL	ADV	CODE1_ART	PRE	VARCHAR2(32)	Code 1 item
MEDIKA_CLINICAL	ADV	CODE2_ARTSEQ	PRE	VARCHAR2(3)	Code 2 item
MEDIKA_CLINICAL	ADV	CODE3_BODYSYS	PRE	VARCHAR2(25)	Code 3 item
MEDIKA_CLINICAL	ADV	CODER	PRE	VARCHAR2(20)	User account that performe
MEDIKA_CLINICAL	ADV	COMMENTS	PRE	VARCHAR2(250)	Comments about AE
MEDIKA_CLINICAL	ADV	CONFIDENCE	PRE	VARCHAR2(2)	Algorithm step
MEDIKA_CLINICAL	ADV	CONTIN	PRE	NUMBER(1)	AE continuing?
MEDIKA_CLINICAL	ADV	EVENT	PRE	VARCHAR2(25)	Adverse event
MEDIKA_CLINICAL	ADV	NORMALIZED	PRE	VARCHAR2(25)	Normalized text
MEDIKA_CLINICAL	ADV	NUM_MATCHES	PRE	VARCHAR2(4)	Number of matches found
MEDIKA_CLINICAL	ADV	OUTCOME	PRE	NUMBER(1)	AE outcome
MEDIKA_CLINICAL	ADV	SERIOUS	PRE	NUMBER(1)	Serious?
MEDIKA_CLINICAL	ADV	SEVERITY	PRE	NUMBER(1)	Severity
MEDIKA_CLINICAL	ADV	STARTDD	PRE	VARCHAR2(2)	Start day - DD
MEDIKA_CLINICAL	ADV	STARTMMM	PRE	VARCHAR2(3)	Start month - MMM
MEDIKA_CLINICAL	ADV	STARTYYYY	PRE	VARCHAR2(4)	Start year YYYY
MEDIKA_CLINICAL	ADV	STOPDD	PRE	VARCHAR2(2)	Stop date - DD
MEDIKA_CLINICAL	ADV	STOPMMM	PRE	VARCHAR2(3)	Stop month - MMM
MEDIKA_CLINICAL	ADV	STOPYYYY	PRE	VARCHAR2(4)	Stop year - YYYY
MEDIKA_CLINICAL	ADV	STUDYREL	PRE	NUMBER(1)	Relationship to study drug
MEDIKA_CLINICAL	ADV	WORKFLOW	PRE	VARCHAR2(5)	Workflow

Setting up coding targets

What are coding targets?

A *coding target* is a Clintrial software object that identifies a set of items in a panel of a clinical data protocol; these items are used with a coding thesaurus protocol during coding.

Each coding target consists of up to eleven items. The following three items are required:

- Code 1 Item — The item that stores the code.
- Verbatim Text Item — The item that stores the verbatim text to be coded.
- Workflow Item — The item that indicates whether the panel was coded automatically or coded interactively.

The following eight items are optional:

- Normalized Text Item — The item that stores the normalized text discarded if the Comprehensive Normalization attribute is selected.
- Code 2 Item — For a multipart code, the item that stores the second part of the code.
- Code 3 Item — For a multipart code, the item that stores the third and last part of a three-part code.
- Date Item — The item that stores the date the item was coded.
- User Item — The item that stores the name of the user account doing the coding.
- Auto Matches — The item that stores the number of matches found by automatic coding.
- Algorithm — Name of the algorithm to be used in coding the verbatim text item in this panel.
- Confidence — The item that stores the step of the automatic thesaurus algorithm that produced a match.

For information on how to specify these items, and how they are used, see "List of coding target attributes" on page 335.

How to set up a coding target

To set up a coding target, do the following:

1. In the panel that will contain the item to code, create the three required coding target items, and any optional items you need.

   When you specify a coding thesaurus protocol as the Thesaurus attribute for an item and save the item, Design creates the coding target with that item as the Code 1 Item attribute for the coding target. You must then specify other coding target items.

2. To specify the other coding target items, do the following:

   a. In the Panel Browser, select the panel for which you want to specify coding target items.

   b. From the **Panel** menu, select **Coding Targets**.

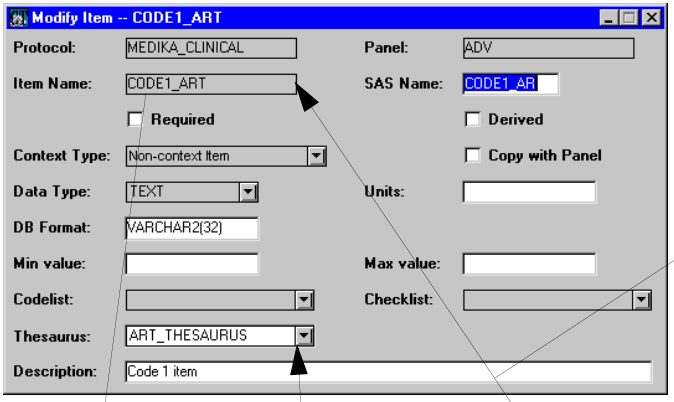
   c. In the Coding Target Browser, select the coding target for which you want to specify coding target items.

   d. From the **Coding Targets** menu, select **Modify**.

   e. Specify the Verbatim Text Item and the Workflow Item, as well as any optional coding target items that you need.

When you install the panel in the clinical data protocol, the Clintrial software maps the coding target to the coding thesaurus view, using the Code 1 Item coding target item in the clinical data panel to map to the Code 1 item specified in the thesaurus view. The following example shows the relationship, using the MEDIKA_CLINICAL and ART_THESAURUS sample protocols:

Clinical data protocol (MEDIKA_CLINICAL)

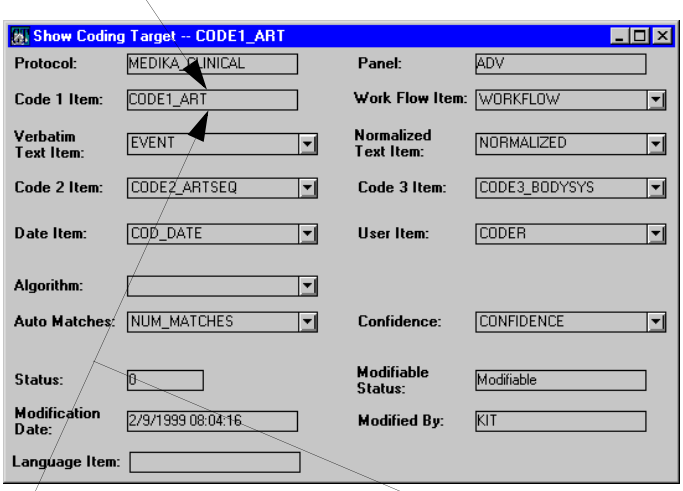


The clinical data panel's Code 1 item is automatically the value of the Code 1 Item coding target item in the clinical data panel's coding target.

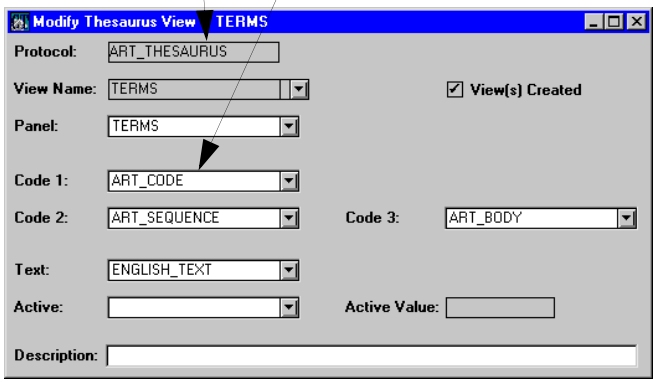
The item (CODE1_ART) that has a value for the Thesaurus attribute is the Code 1 item in the clinical data panel (ADV).

The Code 1 item's Thesaurus attribute specifies which coding thesaurus protocol to use (ART_THESAURUS).

Clinical data protocol (MEDIKA_CLINICAL)



Coding thesaurus protocol (ART_THESAURUS)



The item that is the value of the clinical data panel's Code 1 Item coding target (CODE1_ART) corresponds to the Code 1 item (ART_CODE) in the view of the terms panel in the coding thesaurus protocol.

You can modify the coding target items (that is, select different items), except for the coding target item for the Code 1 Item attribute.

The Clintrial software uses a default thesaurus algorithm for coding the verbatim text item. To use a customized thesaurus algorithm, do the following:

1. Create the panels needed to hold your coding thesaurus information.
2. Create customized thesaurus views.
3. Create a customized thesaurus algorithm.
4. Specify the customized thesaurus algorithm as the value of the coding target Algorithm attribute for the panel containing the verbatim text item.

Note: An item can serve only once in a panel as one of the values of the coding target, with the exceptions of the Verbatim Text Item, the Date Item, and the User Item, each of which can be used as a value by multiple coding targets. Items used as other values for a coding target cannot be used by another coding target in the same panel, or as a different value within the same coding target.

For more information on thesaurus views, see "What is a thesaurus view on a Clintrial software panel?" on page 322.

For more information on creating items to be coded, see "Setting up coded panels in a clinical data protocol" on page 327.

For more information on customized thesaurus algorithms, see "How to customize coding algorithms" on page 340.

List of coding target attributes

The following tables list the required and optional coding target attributes:

Required attribute:	Description:
Protocol (read-only)	Name of the protocol containing the panel with the item to be coded.
Panel (read-only)	Name of the panel with the item to be coded.
Code 1 Item (read-only)	Name of the item that stores the code (or, the first part of the code if it is a multipart code). The value is the name of the item in the clinical data panel that has the Thesaurus attribute set, and maps to the item in the coding thesaurus protocol that stores the codes.
Work Flow Item	Name of the item that indicates whether the panel was autocoded or coded interactively. Values displayed in the item after coding: • AUTO — Autocoded • INT — Interactively coded • FAIL — Failed to code during an autocode attempt (Possible only if Manage system parameter AUTOCODE_SET_FAIL is set to Yes.)
Verbatim Text Item	Name of the item that stores the verbatim text to be coded.

Optional attribute:	Description:
Code 2 Item	For a multipart code, the name of the item that stores the second part of the code.
Code 3 Item	For a three-part multipart code, the name of the panel item that stores the third and last part of a three-part code.
Date Item	Name of the item that stores the date the item was coded.

Optional attribute:	Description:
User Item	Name of the item that stores the name of the User doing the coding.
Algorithm	Name of the algorithm to be used in coding the verbatim text item in this panel. If you do not specify a value, then the Clintrial software-supplied algorithm is used, and no value appears here. The default is the Clintrial software-supplied algorithm, which is used if you do not specify an algorithm. Modify the coding targets to specify a customized thesaurus algorithm if you do not want to use the default thesaurus algorithm.
Auto Matches	Name of the item that stores the number of matches found by automatic coding. This is useful if the Manage system parameter AUTOCODE_FULL-_CNT is set to Yes, to capture an exact count.
Confidence	Name of the item that stores the step of the automatic thesaurus algorithm that produced a match.
Normalized Text Item	Name of the item that stores the verbatim text at the point that the thesaurus algorithm ends (either by successfully coding, or by completing all steps without successfully coding). This item is optional, whether the Comprehensive Normalization attribute is set.

List of read-only attributes

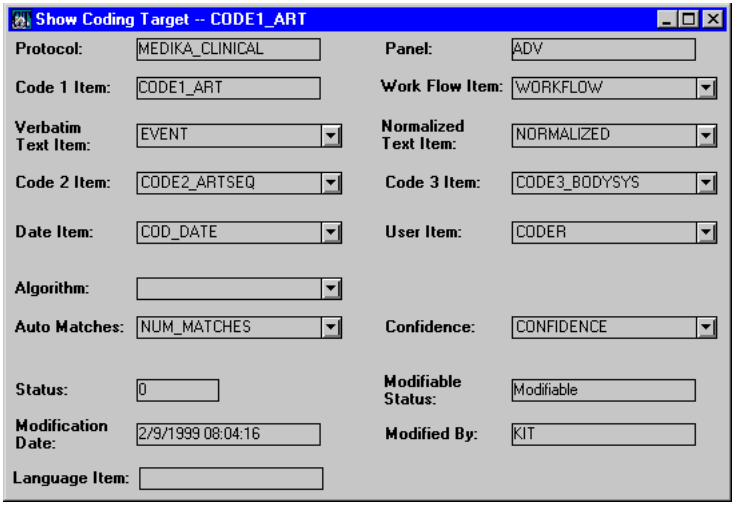
When you select the **Coding Target** menu's **Show** command, the Clintrial software displays the previously described coding target attributes as read-only, and also displays these additional read-only attributes:

Attribute:	Description:
Status	For internal Clintrial software use.
Modifiable Status	Modifiable if the coding target can be modified, Not Modifiable if the coding target cannot be modified.

Attribute:	Description:
Modification Date	Date of the most recent modification of the coding target.
Modified By	Name of the user account that created the coding target.
Language Item	For future use.

To create coding targets, set your protocol to the clinical data protocol, and from the **Panel** menu, select **Coding Targets**.

The following example shows the coding targets in the ADV panel in the MEDIKA_CLINICAL clinical data protocol:



Thesaurus algorithms

What is a thesaurus algorithm?

Automatic coding is a method of coding in which the Clintrial software uses an algorithm to search the coding thesaurus and assign an appropriate code to verbatim text.

A *thesaurus algorithm* is a sequence of steps that determines the most appropriate code match for the verbatim text (that is, text entered by the user), such as a disease or a drug name. The Clintrial software supplies a default thesaurus algorithm. You can create a customized thesaurus algorithm if the default algorithm does not meet your requirements.

For information about automatic coding, see *Manage, Classify, and Lab Loader*.

Steps in a Clintrial software-supplied coding algorithm

The Clintrial software supplies a default thesaurus algorithm for use in automatic coding in Manage. Before automatic coding begins, the Clintrial software first normalizes the entered verbatim text as follows:

1. Changes all characters to uppercase.
2. Discards any punctuation identified in the thesaurus language being used to code the item storing the verbatim text.
3. Discards any extra spaces or tab characters between words, beyond a single space.

Then, the thesaurus algorithm runs, consisting of the following steps:

Step Number:	**Step Description:**
1	The Clintrial software looks in the TERMS view for a term that is an exact match to the verbatim text: • If exactly one match is found, the Clintrial software returns the code, records the number of matches as "1", and records the confidence level as "1". The algorithm ends. • If multiple matches are found, the Clintrial software returns no code and records the number of matches as ">1" or the actual number of matches, depending on the setting of the system parameter AUTO_FULL_CNT. The algorithm ends. If no match is found, the Clintrial software goes to Step 2.

2	The Clintrial software looks in the SYNONYMS view for a synonym that is an exact match to the verbatim text: • If exactly one match is found, the Clintrial software returns the code, records the number of matches as “1”, and returns the confidence level as “2”. The algorithm ends. • If multiple matches are found, the Clintrial software returns no code and records the number of matches as “>1” or the actual number of matches, depending on the setting of the system parameter AUTO_FULL_CNT. The algorithm ends. • If no match is found, the Clintrial software goes to Step 3.
3	The Clintrial software removes from verbatim text any word that is an exact match to a stopword defined in the STOPWORDS panel, creating a filtered text string. The Clintrial software also breaks up the verbatim text into an array of words. The Clintrial software goes to Step 4.
4	The Clintrial software looks in the TERMS view for a term that is an exact match to the filtered text string: • If exactly one match is found, the Clintrial software returns the code, records the number of matches as “1”, and returns the confidence level as “4”. The algorithm ends. • If multiple matches are found, the Clintrial software returns no code and records the number of matches as “>1” or the actual number of matches, depending on the setting of the system parameter AUTO_FULL_CNT. The algorithm ends. • If no match is found, the Clintrial software goes to Step 5.
5	The Clintrial software looks in the SYNONYMS view for a synonym that is an exact match to the filtered text string: • If exactly one match is found, the Clintrial software returns the code, records the number of matches as “1”, and returns the confidence levels as “5”. The algorithm ends. • If multiple matches are found, the Clintrial software returns no code and records the number of matches as “>1” or the actual number of matches, depending on the setting of the system parameter AUTO_FULL_CNT. The algorithm ends. • If no match is found, the Clintrial software goes to Step 6.

6	The Clintrial software looks in the TERMS view for a term that contains each word of the array in any order: • If exactly one match is found, the Clintrial software returns the code, records the number of matches as "1", and returns the confidence level as "6". The algorithm ends. • If multiple matches are found, the Clintrial software returns no code and records the number of matches as ">1" or the actual number of matches, depending on the setting of the system parameter AUTO_FULL_CNT. The algorithm ends. • If no matches are found, the Clintrial software goes to Step 7.
7	The Clintrial software looks in the SYNONYMS view for a synonym that contains every word in the array in any order: • If exactly one match is found, the Clintrial software returns the code, records the number of matches as "1", and returns the confidence level as "7". • If multiple matches are found, the Clintrial software returns no code and records the number of matches as ">1" or the actual number of matches, depending on the setting of the system parameter AUTO_FULL_CNT. • If no matches are found, the Clintrial software returns no code, records the number of matches as "0", and records the confidence level as "8".

How to customize coding algorithms

The Clintrial software allows you to create a customized thesaurus algorithm. You can:

- Normalize or forgo normalization.
- Add steps.
- Omit steps.
- Reorder steps.
- Specify views other than TERMS, SYNONYMS, and STOPWORDS.

To create a custom algorithm, do the following:

1. From the Protocol Browser, select the protocol for which you want to create a custom algorithm.
2. From the **Protocol** menu, select **Thesaurus Algorithms**.
3. In the Thesaurus Algorithm Browser, from the **Thesaurus Algorithm** menu, select **Create** and specify the thesaurus algorithm attributes.

For information on the custom thesaurus algorithm attributes, see "List of customized algorithm attributes" on page 342.

How to customize normalization

When you create a customized thesaurus algorithm, the Comprehensive Normalize attribute is selected by default. When this attribute is selected, the customized algorithm normalizes the verbatim text, as described in "Steps in a Clintrial software-supplied coding algorithm" on page 338.

If you clear the Comprehensive Normalization attribute, characters are not changed to uppercase and punctuation is not discarded. The result is that automatic coding codes with case sensitivity, and is sensitive to differences in punctuation.

Caution: Even if you clear the Comprehensive Normalize attribute, Design discards any extra spaces, beyond a single space.

Note: It is recommended that you create your custom algorithms with the Comprehensive Normalize box checked. However, you may want to code without comprehensive normalization in the following circumstances:

- If you are using a dictionary that contains case-sensitive terms (that is, terms in mixed uppercase and lowercase)
- For use with Classify for more complex customizations of matching terms

For more information on Comprehensive Normalization, see the Design Help.

Example: Customized algorithm

For example, suppose there are two protocols, CARDIAC_DRUG and RENAL_DRUG. You want both protocols to use the COSTART coding thesaurus, but you want each protocol to code by different steps and use different synonyms data, appropriate to cardiac or renal adverse events.

You then do the following:

1. In the COSTART protocol, in addition to the SYNONYMS panel, create two additional synonyms panels, CARDIAC_SYN and RENAL_SYN, to define tables to store synonyms data appropriate to cardiac or renal adverse events.
2. In the COSTART protocol, in addition to the SYNONYMS thesaurus view, create two additional synonyms thesaurus views, CARDIAC_SYN and RENAL_SYN, based on the corresponding panels of the same name.

3. In the COSTART protocol, create two customized thesaurus algorithms, CARDIAC_ALG and RENAL_ALG. Create steps as needed, and base each step on an appropriate thesaurus view, including the customized synonyms thesaurus views.
4. In the CARDIAC_DRUG and RENAL_DRUG clinical data protocols, create the appropriate items in the ADV panels, including a CODE1 item in each that specifies the COSTART coding thesaurus protocol.
5. Save the ADV panels.
6. In the CARDIAC_DRUG and RENAL_DRUG clinical data protocols, modify the coding targets. Specify the appropriate items as coding targets, and select the CARDIAC_ALG and RENAL_ALG thesaurus algorithms, respectively, as the algorithms to be used for coding.

You specify a customized thesaurus algorithm as the coding target Algorithm attribute. If you do not specify this attribute, the Clintrial software uses the default algorithm.

List of customized algorithm attributes

The following table lists the attributes you specify to create or modify a customized thesaurus algorithm:

Attribute:	**Description:**
Protocol	Name of the coding thesaurus protocol that contains the thesaurus algorithm.
Algorithm	Name of the algorithm you are creating or modifying.
Comprehensive Normalization	Select to have normalization occur before beginning to apply the customized thesaurus algorithm; clear to leave the verbatim text unnormalized before applying the algorithm.
Preferred Terms View	Name of preferred terms view for algorithm. (The default Clintrial software-supplied algorithm uses the TERMS view.)
Description	Description of the customized thesaurus algorithm.

Attribute:	Description:
Order	The number of the step. You can add or delete steps, or, in effect, change the order of the steps by changing the Step Type and Step View attributes. This number appears in the item that you identify as the Confidence coding target in the clinical data protocol containing the panel with the verbatim text item.
Step Type	Type of activity performed by this step. The values are: Exact —Matches the verbatim text string exactly. Contains — Matches with terms containing the verbatim text, in any order. Filter — Removes stopwords from the verbatim text.
Step View	The thesaurus view used by this step.

Caution: The Filter step uses the view on a stopwords table with a single item for each language. If you inadvertently try to create a Filter step on a terms or synonyms view, Oracle errors result that may not be discovered until the first automatic coding runs.

List of read-only attributes

When you select the **Thesaurus Algorithm** menu's **Show** command, Design displays the previously described customized thesaurus algorithm attributes as read-only, and also displays these additional read-only attributes:

Attribute:	Description:
Modification Date	Date of the most recent modification of the thesaurus algorithm.

Attribute:	Description:
Modified By	Name of the user account that created the thesaurus algorithm.
Modifiable Status	Modifiable if the thesaurus algorithm can be modified. Not Modifiable if the thesaurus algorithm cannot be modified.

15 Using Central Coding with Clintrial

Overview

Coding is the process of assigning a standard code from a coding dictionary to a value that has been entered for an item as verbatim text. For example, a drug could be assigned a standard code from the WHO-DD dictionary. If the verbatim text entry for the drug a subject took to treat a headache during the study was entered as "Tylenol", then from the WHO-DD dictionary, it could be coded as "002144.01.008", with the Trade Name "Tylenol aches and strains medication" and the Preferred Name "Parafon".

Coding provides the following benefits to clinical data management:

- It standardizes the use of terminology describing information, such as adverse events data, that is collected during clinical studies.
- It reduces the time and effort that is required to manually look up and select terms in a coding dictionary.

The decision as to which coding utility will be used is made in the Admin and Design modules, either as a system-wide parameter for all protocols, or on an individual protocol basis using a protocol parameter.

The protocol designer determines which items will be coded. This is done in the Clintrial Design module. Typical items to be coded are clinical events, drugs, or diseases.

What is a coding dictionary?

A *coding dictionary* is a dictionary thesaurus that contains standard codes for a particular type of clinical data. There are two types of dictionaries: industry-standard and user-defined.

Contact Oracle for a list of supported dictionary versions.

What is verbatim text?

The text entered by the user is referred to as verbatim text. The *verbatim text* is the entered text for which you want to assign a standard code from a coding dictionary.

Why use coding dictionaries?

You use coding dictionaries to:

- Provide standardization for statistical analysis.
- Summarize terms by grouping verbatim terms into standardized terms.
- Create a standard language that is comparable across all therapeutic teams.
- Reduce the time needed to reach a final format, which leads to faster regulatory approval.

Which coding utility should I use?

In order to match clinical data (verbatim text) against standard dictionary codes, you may either:

- Use Clintrial Manage, and optionally, the Clintrial Classify extended module.
- Use the Oracle Central Coding application.

The decision to use either approach is made on a protocol basis. For example, existing protocols may continue to use the Manage and the Classify module for coding, while new protocols may use Central Coding. Of course, existing protocols may be updated to use Central Coding if you want to use it as a standard within your organization.

The Central Coding software is a web-based application that integrates with both Oracle's InForm and Clintrial software to provide centralized coding for studies within an organization.

Because the Central Coding software works independently from the Clintrial and InForm software, you can have parallel work paths for the clinical study teams. As a result, coding can happen earlier in the study cycle, providing valuable data visibility for study managers who review and assess study safety concerns.

When you use the Central Coding software for coding, you are not limited to three-part coding as you are with the Clintrial software.

Note: The instructions contained in this chapter for setting up the Clintrial application for use with Central Coding assume that all Central Coding tasks have been performed to support Clintrial. For information on Central Coding setup, see the *Central Coding Users Guide*.

Types of dictionaries

Raw dictionary data files from a regulatory authority or third-party vendor are loaded into Central Coding by the customer. The Clintrial software provides a means to use industry-standard dictionaries, such as:

- World Health Organization Drug Dictionary (WHO-DD)
- World Health Organization Drug Dictionary C Format (WHO-DD C) (which may be used to autocode a Preferred Term or Trade Name)
- Coding Symbols for a Thesaurus of Adverse Reaction Terminology (COSTART)
- Medical Dictionary for Drug Regulatory Activities (MedDRA and MedDRAJ)
- Data File for Ethical Drugs (Coding Table) from Iyaku-Joho-Kenkyujo, Inc. (JDrug)

The Central Coding software also provides a means to use user-defined dictionaries.

How automatic coding relates to interactive coding

At data entry, the data-entry operator enters the verbatim text for the item to be coded. Central Coding provides two methods of determining the correct code for an entered item that has been set up as a coded item:

- Automatic coding

  If a single match is found on the verbatim text using the specified coding algorithm, the Central Coding software selects a code for the verbatim text item to be coded. The Central Coding software supplies a default coding algorithm, or you can create your own customized coding algorithms.

- Interactive coding

  In Central Coding, the data manager selects a code from a scrolling list of similar matching terms, each of which has an associated code.

The Clintrial data manager sets up which panel items will be coded by specific Central Coding dictionaries in the Design module, and actually sends them to Central Coding using the Manage module. The requests may be batch loaded or sent immediately to be coded in Central Coding. Depending on how the Central Coding application is setup, the items will be designated for Automatic Coding first if possible. The uncoded verbatims are then reviewed and coded via interactive coding, also in Central Coding.

For more information on automatic and interactive coding, see the *Central Coding Users Guide*.

Performing coding in Central Coding

Automatic coding is the process by which the Oracle Central Coding software automatically assigns codes to verbatim text. All actual coding is performed in the Central Coding application.

When you run automatic coding, the Central Coding software tries to match the verbatim text to terms (or synonyms) in the coding dictionary:

- If one match is found, then the Central Coding software assigns the associated code.
- If there are no matching terms (or synonyms), then no code is assigned.
- If there are multiple matching terms (or synonyms), then no code is assigned because the Central Coding software does not know which code is the most appropriate.

For records for which no matches or multiple matches are found, you may want to perform interactive coding in the Central Coding software, or you may want to review the data, create an appropriate synonym, and then run automatic coding again.

For more information on automatic coding as performed by the Central Coding application, see the *Central Coding User Guide*.

Setting up a dictionary in Clintrial for use with Central Coding

Steps in setting up Clintrial to work with Central Coding

To set up the Clintrial software to use a Central Coding dictionary, you must do the following:

1. Ensure network or Internet access to the Central Coding Server. (See the Clintrial *Getting Started* guide for more information.)
2. Configure the CC_HOST parameter with the HTTP address of the Central Coding interface. This may be set in Admin as a System or Protocol parameter.
3. Set the parameter USE_CENTRAL_CODING to Yes for the target protocol. This may be set in Admin as a System or protocol parameter.
4. Create a dictionary definition. (See "Clintrial dictionary overview" on page 350 in the Clintrial *Manage, Classify, and Lab Loader* manual.)

5. Define labels for the dictionary definition. (See "Setting up Label Types and Label Names in Clintrial Central Coding Dictionaries" on page 353 in the Clintrial *Manage, Classify, and Lab Loader* manual).
6. Select a dictionary definition for an item in a panel.
7. Create a Central Coding encoding target. (See "Creating a Central Coding Encoding Target" on page 355 in the Clintrial *Manage, Classify, and Lab Loader* manual.)
8. Map the target items with the defined labels. (See "Mapping a defined label with a Coding target" on page 356 in the Clintrial *Manage, Classify, and Lab Loader* manual.)
9. Use the Clintrial Manage module to send coding requests to Central Coding. (In this chapter.)

Clintrial dictionary overview

This is a table of the terms used in this section:

Term:	Definition:
Dictionary Definition	A configuration in Clintrial for a coding dictionary used in Central Coding.
Encoding Target	A set of information required to send and receive Central Coding requests. This information includes the location of the verbatim text and the locations of the target items in which to store the returned coding information.
Label Name	Label name for a Central Coding data value, as used by Central Coding.
Label Type	Label type includes Target, Associated and Verbatim.
Target Label	A label to use in receiving coding information from Central Coding. Each Target label gets mapped to a Clintrial item to be updated with the coding results. At least one Target Label must be specified before an Encoding Target can be used for coding.

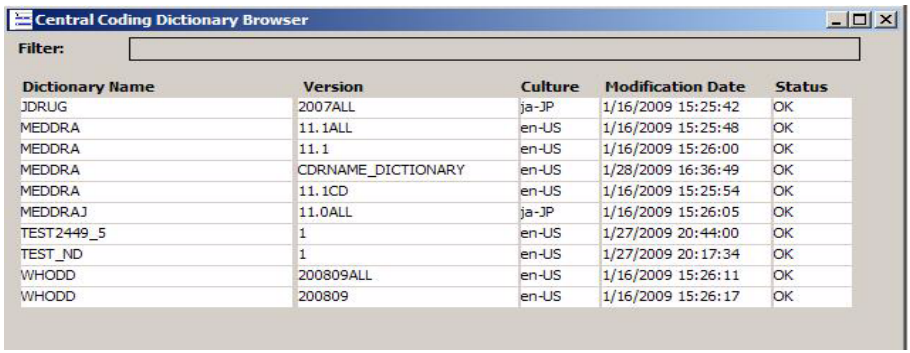
Term:	Definition:
Associated Label	A label to use in sending additional information to Central Coding. Each Associated label gets mapped to a Clintrial item. The Label Name and the mapped item value are sent to Central Coding when making requests.
Verbatim Label	A Label Type in Clintrial for coding with Central Coding. The Verbatim Label Names are used to classify the verbatim text, for example “AE” or “DISEASE”.
Verbatim Text	The text string that was entered by the user and which requires coding.

Unlike with Clintrial Coding, the actual dictionary information is not stored in Clintrial, but is stored on the Central Coding Server. However, Clintrial needs to understand how the information from the dictionary is to be used when received.

Dictionaries are defined here in Clintrial Design, and label names that are needed for a particular encoding target will be included in the dictionaries. The dictionaries you create are system-wide (like flags and notes).

You can create one or more dictionaries as needed to meet your requirements. The **Central Coding Dictionaries Browser** (accessed from the **Objects** menu) lists existing dictionaries. The functions available in this browser are similar to other browser windows: **Copy**, **Create**, **Modify**, **Show** and **Delete**. The privilege needed to define and maintain these dictionaries is DESIGN_GLOBAL.

Note: You may only delete dictionaries you have defined yourself.



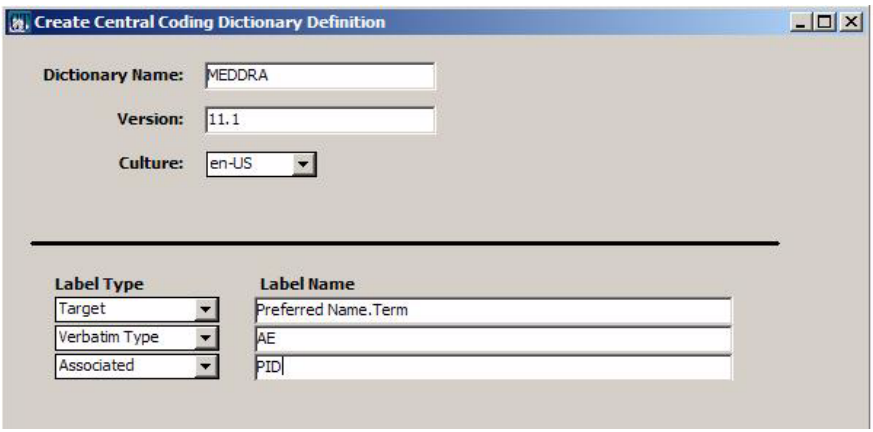
Creating a new dictionary in Clintrial

You create a new dictionary definition in Clintrial but the actual dictionary will be on the Central Coding server. For each dictionary, you must define which Label Names to request from Central Coding (Target Items), and which Label Names to send to Central Coding (Associated Items). You must also specify which Verbatim Types are allowed to be chosen for this dictionary. Labels for Target Items, and Verbatim Types are required; labels for Associated Items are optional. The **Add**, **Insert** and **Delete** options are provided in order to support a flexible number of items.

If you are creating a new dictionary, the following fields will be enterable: **Dictionary Name**, **Version** and **Culture**. The **Dictionary Name** is used only by Clintrial, and must be unique. The fields **Version** and **Culture** already exist in every Central Coding dictionary, and must match the Central Coding dictionary definition exactly. These fields can not be changed once saved in Clintrial, and will be grayed out on modification.

The **Culture** field is a drop-down list, with values populated with rows from a new Clintrial defined Codelist **CTS_CULTURE**. This codelist is installed with one row for **en-US** (American English) and one row for ja-JP (Japanese). You may add rows as necessary.

Selecting a dictionary definition for an item in a panel.



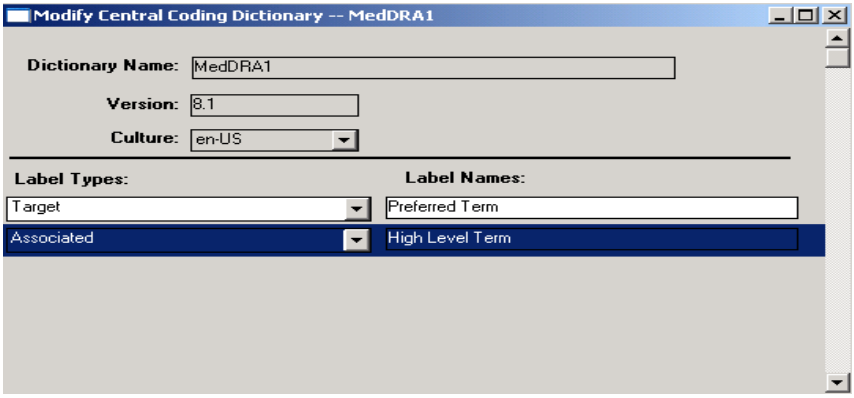
The uniqueness of a Dictionary Definition is the combination of the Dictionary Name, Version and Culture properties of the definition. Once it is in modification mode, those fields would be grayed out.

The following information is automatically sent to Central Coding when a coding request is submitted and does not need to be added as associated items:

Clintrial	Central Coding
Protocol	Study
Subject Item	Patient ID
BlockItem.<Block Repeat Item>	Visit
CodedItem.CT_RECID	Client Item Identifier
Verbatim Item	Verbatim

Setting up Label Types and Label Names in Clintrial Central Coding Dictionaries

The Label Types offered in the drop-down list are: Target, Verbatim and Associated. At least one Target Type and one Verbatim Type are required to function correctly. Associated Labels are optional. Existing rows are sorted by Label Type. As many targets as desired may be added.



The labels for the Label Names for use in Clintrial consist of the following sections separated by a period (.). The actual valid entries will vary depending on the Central Coding dictionary used.

1. The Name of the Level.
2. May be **Code**, **Term**, **AddInfo**, depending on the column in which the information is given.
3. For **AddInfo**, the name of the additional item.
4. The items **CoderName** and **Dictionary** only have a one-part name.

For example, from the **WHO-DD** dictionary, to return:

- The Preferred Name Code, use: **Preferred Name.Code**
- The Medicinal Product name and Manufacturing Date, use: **Medicinal Product.AddInfo.MA Date**

Make sure to preserve spaces and case when entering the target labels.

The Label Types and Label Names are repeated; you can add as many as you need.

The Label Names must EXACTLY match those used by the dictionary as defined at the Central Coding Server. You may refer to the Central Coding dictionary web page to ensure an exact match.

For more information on the Central Coding dictionary web page and how to connect to it, see the *Central Coding Users Guide*.

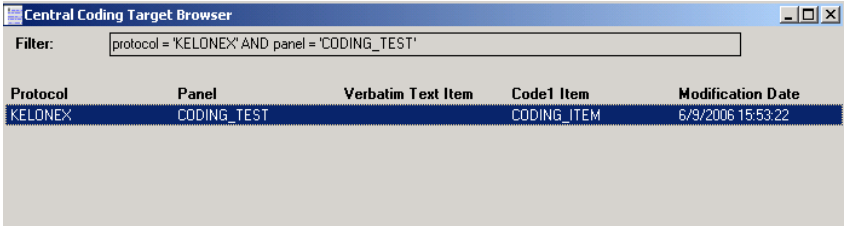
Restrictions apply once a dictionary definition is saved and used by a protocol. For example, the **Delete** action for the dictionary is not allowed if it is in use. Also if the dictionary is in use, any **Target** rows can not be deleted in the **Modify** dialog.

If the dictionary is in use and you modify it by adding or changing the Label Names, the associated Encoding Target will be invalid. The targets in use will be updated with the changes, however you must update the Encoding Target before it can be used by identifying any items for new labels and re-installing the panel.

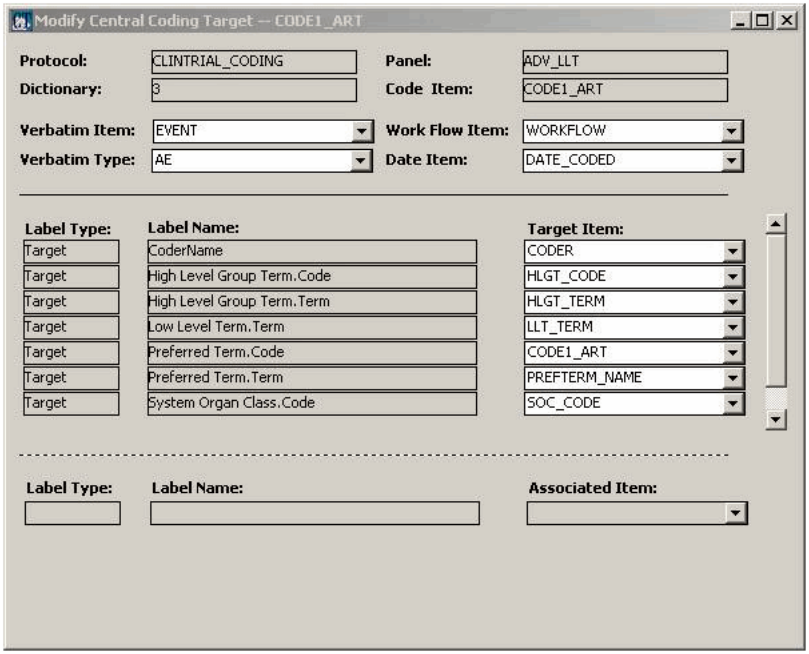
Creating a Central Coding Encoding Target

If USE_CENTRAL_CODING is set to Yes, the Coding Targets menu item will be disabled and the Central Coding Targets... menu item will be enabled.

The Central Coding target browser is very similar to the classic Clintrial encoding target browser. For example, the title **Coding Target Browser** is displayed when selecting the **Coding Targets** menu. The title **Central Coding Target Browser** is displayed when selecting the Central Coding Targets. The **Modify** and **Show** options are available from the browser.



After selecting a row to edit, the Modification window opens.



Verbatim Item, **Verbatim Type** and **Workflow Item** are required. The **Date Item** is optional.

The **Verbatim Type** drop-down shows those types defined for the dictionary used. There is a list of the names of **Target** and **Associated Items**, as defined in the dictionary. Each of these has a drop-down next to it, populated with all appropriate items within THIS panel. Items must be of the appropriate type. All chosen items must be of data type TEXT, except the **Date Item** which must be of type DATE. Associated Items may also use items from the CONTEXT panel.

Items can be used as a Target Item only once. If an item has already been selected as a Target Item, it will not appear in the list of items for selection for subsequent Target Items. Items may be used more than once as Associated Items, or as both a Target and Associated Item, so the drop-down list for Associated Items contains all items of type TEXT. The primary Target Item, for which the dictionary attribute has been defined, must be mapped to one of the target Label Names. All the Target Items and Associated Items, if defined, must be filled in.

Note: Central Coding cannot return Date Information. This field is filled in by the stored procedure in Clintrial.

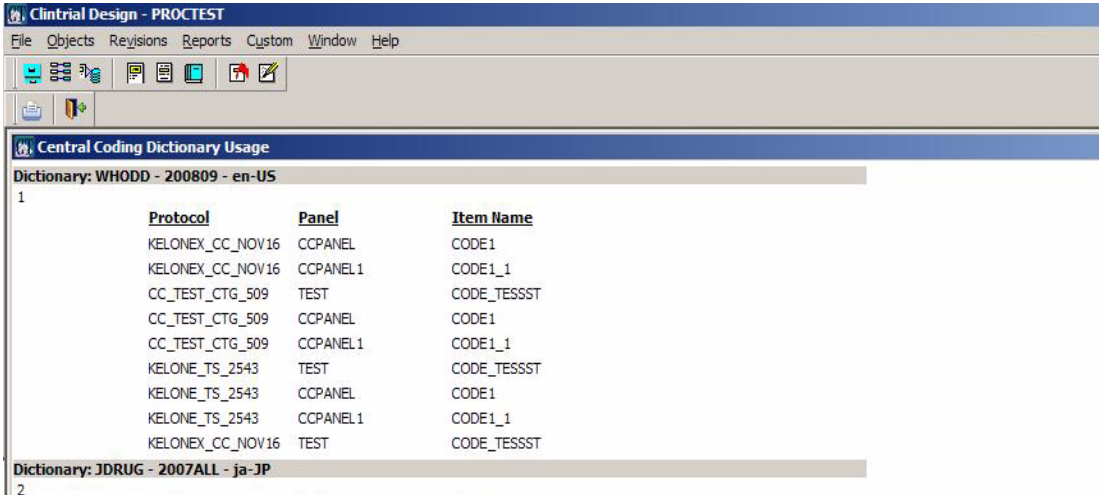
The Label Name/Target Item rows are repeating. The number of rows and the list of names in Label Name are what have been defined in the selected dictionary. The drop-down list under Target Item consists of the items defined in the specified Clintrial panel. You choose all the Target Items. If the item (Code Item) assigned to the dictionary is not selected, a warning message is displayed.

You select the panels and items to be sent to Central Coding in the manage module. See **Chapter 5: Using Manage & Central Coding to code data** in the Clintrial *Manage, Classify, and Lab Loader* manual for the details.

Dictionary Usage By Dictionary report

A report is provided to assist you in identifying which Encoding Targets use particular dictionaries. There is a new option in Design Dictionaries: **Dictionary Usage By Dictionary**. You are able to select one or more dictionaries to include in the report. The report is available to users with the DESIGN_GLOBAL privilege.

The report lists all the Central Coding dictionaries created in Clintrial by Name, Version and Culture, the protocols which use each, and the panels and items within the protocol for each dictionary.



Part III: Objects for Entering and Editing Clinical Data

16 Page Templates

Using page templates

What is a page template?

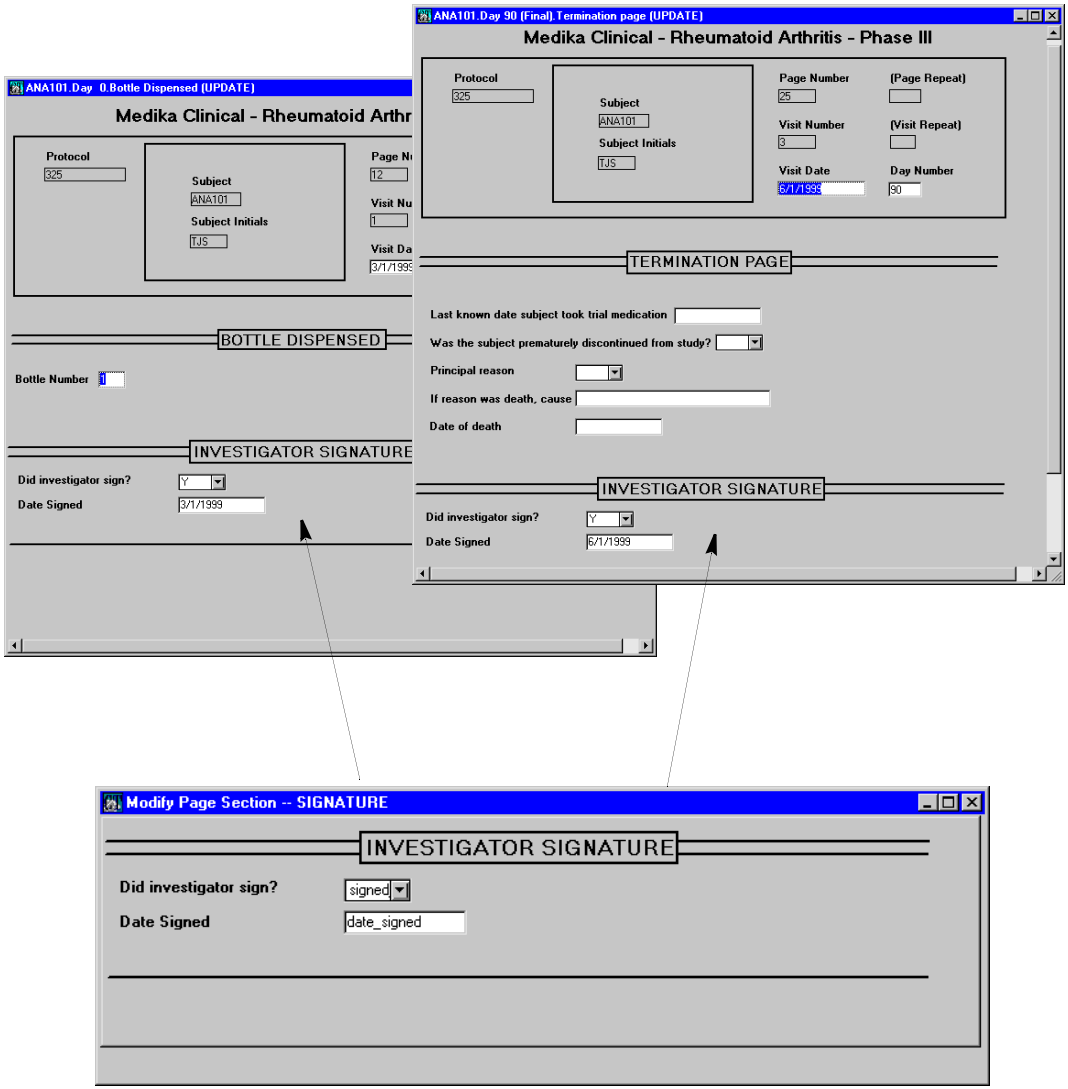
A *page template* is a grouping of page sections that defines the layout of page sections in a study page. A page template consists of one or more page sections, as well as a context page section, if the other page sections are based on panel Types 1, 2, 3, 4, or 5. One or more study pages may be based on a single page template.

How page templates relate to page sections

One or more page templates can use the same page section.

For example, in the MEDIKA_CLINICAL protocol, the BOTDISP and TERMINATION page templates both contain the page section SIGNATURE, as well as page sections unique to each page template.

The following figure shows the relationship of a single page section to two page templates:



How page templates relate to study pages

One or more study pages in a study book can use a single page template. A study page is unique in a study book, but a page template may or may not be used uniquely in a study book.

A page template has no length limit. You are not bound by the height of the screen, or by the size of the paper CRF page that corresponds to the study page using the page template. For example, the paper CRF Laboratory Examination requires two pages, but the corresponding online LABSLNG page template consists of three page sections (and the CONTEXT page section) that appear as a single study page in the first and last visit.

For information on study pages in study books, see Chapter 18.

Creating and modifying a page template

How to create a page template

To create a page template:

1. From the **Page Template** menu, select **Create**.

   The Page Template Attributes dialog box opens.
2. Enter a name and description, then click **OK**.

   The Select Page Section dialog box opens.
3. Select the page section you want to add to the page template and then click **OK**.
4. To add additional page sections, repeat Step 3 as necessary, then click **Yes** to save the page template.

Note: If you first select a page section other than CONTEXT, Design adds the CONTEXT page section to the top of the page template automatically, followed by the selected page section. The CONTEXT page section must be the first page section in the page template, and cannot be reordered.

How to modify a page template

To modify a page template:

1. With the page template displayed, use the **Page Template** menu's commands as follows:
   - To add additional page sections, select **Add Page Section**, and then specify the page section to add.
   - To change the order in which the cursor visits page sections in Enter, select **Reorder Page Sections**.
   - To modify a page section layout, click the page section to select it, and then select **Modify Page Section**.
   - To attach a data-entry processing procedure to a page section event, first click the page section to select it, then select **Modify Page Section Events**, and then specify which procedures you want to attach to which event.
   - To modify the subset key value of a page section, select **Modify Subset Key Value**.
   - To delete a page section, first click the page section to select it, select **Delete Page Section**, and then confirm that you want to delete it.
   - To modify page template attributes, select **Page Template Attributes**.
   - To attach a data entry processing procedure to a page template event, select **Page Template Events**, and then specify which procedures you want to attach to which events.
2. In addition, you can use the **Page Template** menu's commands to modify and work with page templates in the following ways:
   - To reorder page sections on a page template, click and drag the page sections where they belong.
   - To confirm that the page template contains a valid combination of page sections on the page template, select **Check Syntax**.
   - To see how the page template functions as a study book page in Enter, select **Test Mode**.
   - To change the magnification at which the page template layout is displayed, select **Change Zoom Factor**, and then select the required percentage.
   - To view the master-detail relationships among page sections on the page template, select **Show Master-Detail**.

For information on page section layout, see Chapter 17.

For information on attaching procedures to events, see the following sections.

For information on subset page sections, see "Panels with subset key items" on page 216, "Master page sections, detail page sections, and subset page sections" on page 485, and "Creating page templates with subset page sections" on page 375.

For information on the **Check Syntax** command, see "How Check Syntax works" on page 369.

For information on creating master-detail relationships, see the following:

- To create panels with master keys, detail keys, and subset keys, see "Panels with master-detail keys" on page 212 and "Panels with subset key items" on page 216.
- To create master page sections, detail page sections, and subset page sections, see "Master page sections, detail page sections, and subset page sections" on page 485.
- To include master page sections, detail pages sections, and subset page sections on page templates, see "Creating page templates with master and detail page sections" on page 370 and "Page template attributes" on page 383.

What is a data-entry processing procedure?

A *data-entry processing procedure* is a PL/SQL procedure that runs when the Enter user performs a particular action related to a study page, page section, or field. For example, a data-entry processing procedure attached to a field could convert an entered value to another value, or move the cursor to a particular field if a specific value is entered in the current field.

You can attach a data-entry processing procedure to:

- A field on a page section

  A data-entry processing procedure attached to a page section field is called a *value changed procedure*. For more information on specifying a value changed procedure in a page section, see "What is a value changed procedure?" on page 481.

- A page section within a page template

  You can specify a data-entry processing procedure in a page template that executes when certain page section events in that page occur. For more information on procedures for page section events, see the following section.

- The page template itself

  You can specify a data-entry processing procedure in a page template that executes when certain page template events in that page occur. For more

information on procedures for page template events, see the following section.

How to attach a procedure to a page section event

In a page template, you can attach a data-entry processing procedure to the following *page section events*:

- Initializing page section — Occurs when a data-entry person opens a study page
- Saving page section — Occurs when a data-entry person saves data on a study page

To attach a data-entry processing procedure to a page section event:

1. In the Page Template Browser, select the page template containing the page section event to which you want to attach the procedure, and then, from the **Page Template** menu, select **Modify**.
2. Select the page section to which you want to attach the procedure.
3. From the **Page Template** menu, select **Modify Page Section Events**.

   The Page Template Events dialog box opens, displaying a list of page section events.
4. Enter the name of the data-entry processing procedure in the Procedure Name field for the appropriate event, and then click **OK**.

How to attach a procedure to a template event

In the page template, you can attach a data entry processing procedure to the following *page template events*:

- Page opened — Occurs when a data-entry person opens a study page
- Page saved — Occurs when a data-entry person data on a study page
- Page deleted — Occurs when a data-entry person deletes data on a study page

Note: In the latter two cases, the procedure runs after changes are committed to the database.

To attach a data-entry processing procedure to a page template event:

1. In the Page Template Browser, select the page template to which you want to attach the procedure, and then, from the **Page Template** menu, select **Modify**.
2. From the **Page Template** menu, select **Page Template Events**.

   The Page Template Events dialog box opens, displaying a list of page template events.
3. Enter the name of the data-entry processing procedure in the Procedure Name field for the appropriate event, and then click **OK**.

For more information on creating data-entry processing procedures, see the *Reference Guide*.

How Check Syntax works

The **Check Syntax** command executes the following checks:

- For every set of page sections that defines a cross-panel master-detail relationship, this syntax check verifies that a valid set of master-detail relationships is defined as follows:
  - A master page section exists for every page section that is based on a detail panel of a cross-panel master-detail relationship.
  - There is only one level of cross-panel master-detail relationship.
- For every nonsubset page section that defines a within-panel master-detail relationship, this syntax check verifies that a valid set of master-detail relationships is defined as follows:
  - There are no more than two page sections (except subset page sections, which are checked separately) based on the same panel.
  - Only one of these page sections can have repeating items.
- For every set of subset page sections based on the same panel, this syntax check verifies that a valid set of subset key values is defined as follows:
  - Only one subset page section can have a null (blank) subset key value. If present, this is the first page section based on the panel, and it has no repeating items. The subset key item is not visible in this page section.
  - No more than two subset page sections can have the same subset key value. If two such page sections are present, there cannot be a page section with a null subset key value based on the same panel.
  - Only one of the page sections can have repeating items. The same page section cannot be used as both the master and the detail page section.

Creating page templates with master and detail page sections

What is a master-detail relationship?

A *master-detail relationship* is a relationship between two page sections on a study page, in which each record in one page section (the master page section) can have one or more associated records in the other page section (the detail page section).

Summary of master-detail relationship

To create a master-detail relationship among records, do the following:

1. Create a panel or panels to contain the records in a master-detail relationship. For panels in a cross-panel master-detail relationship, you must specify master key items and detail key items. A panel in a within-panel master-detail relationship does not use a master key item or detail key items.

   For information on setting up panels for a cross-panel master-detail relationship, see "How to define a master-detail relationship" on page 214.
2. Create a master page section and a detail page section.

   For a cross-panel master-detail relationship, create a master page section based on a panel with a master key item, and create a detail page section based on panel with a detail key item.

   For a within-panel master-detail relationship, master key items and detail key items are not relevant. Create two page sections based on a single panel.
3. Place the master page section and detail page section on a page template.

   The following sections describe how to complete the master-detail relationship on a page template.

Creating within-panel master-detail

Complete the within-panel master-detail relationship on the page template as follows:

1. Add the page section that you want as the master page section.
2. Add one or more page sections that you want as detail page sections.

When you add multiple page sections based on the same panel, the first of those page sections becomes the master page section, and the other page sections become detail page sections. Only one of the page sections can have repeating items.

You can also create within-panel master-detail relationships between subset page sections. For more information on subset page sections see "Creating page templates with subset page sections" on page 375.

Creating cross-panel master-detail

Complete the cross-panel master-detail relationship on the page template as follows:

1. Add the master page section.

   The master page section is the page section based on the panel that has a master key item.

2. Add one or more page sections that you want as detail page sections.

   Each page sections is a page section that is based on a panel that has a detail key item.

You can also create cross-panel master-detail relationships between subset page sections. For more information on subset page sections see "Creating page templates with subset page sections" on page 375.

Combined cross-panel and within-panel relationships

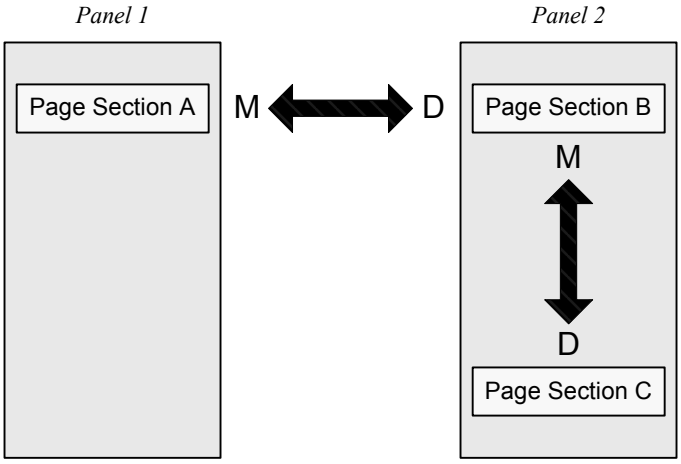
Master-detail relationships are more complex when both a within-panel relationship and a cross-panel relationship are combined. You can create the following combinations:

- Detail page section in a cross-panel master-detail relationship that is also the master page section in a within-panel master-detail relationship

  You can create a cross-panel master-detail relationship in which the detail page section in the cross-panel relationship is also the master page section in the within-panel relationship. The within-panel detail page section cannot

also be a cross-panel detail page section because a detail page section cannot have two different master page sections.

The following figure shows a within-panel master page section that is also a cross-panel detail page section:

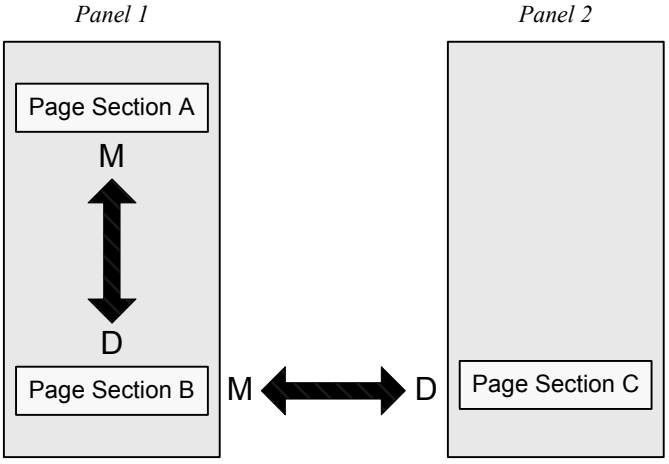


Page section A is master page section to detail page section B. (M -> D, cross-panel)

Page section B is master page section to detail page section C. (M -> D, within-panel)

- Page section (master or detail) in a within-panel master-detail relationship that is also the master page section in a cross-panel master-detail relationship

  The following figure shows a within-panel detail page section that is also a cross-panel master page section:



Page section A is master page section to detail page section B. (M -> D, within-panel)

Page section B is master page section to detail page section C. (M -> D, cross-panel)

Note: In this example, either page section A or page section B could be the cross-panel master page section to page section C.

Restrictions on Combined master-detail

Combined master-detail relationships are subject to the following constraints:

- You can only create this type of chaining if all three page sections do not access the same panel. Within-panel master-detail relationships can exist only between a master page section and one or more detail page sections. The detail page sections cannot be master page sections to other detail page sections.
- Master and detail page sections can have either repeating items or no repeating items, except that a master-detail relationship within the same panel can have at most one of the two page sections with repeating items.
- You can place a master page section by itself on a page template without any cross-panel detail page section on that page template. However, you must place the master page section on a page template before you add the detail page section.

Example of master-detail page template

The Medika sample study's HISTORY page template does not contain a combined cross-panel and within-panel master-detail relationship. However, it could be set up with such a relationship, as follows:

1. In the Medika sample protocol, a master-detail relationship could be set up between the MEDHIST and PRVMED panels, where:
   - MEDHIST is the master panel.
   - The DRUG item in the MEDHIST panel is the master key.
   - PRVMED panel is the detail panel.
   - The MEDNAME item in the PRVMED panel is the detail key.
2. Page sections could be created based on the master panel and the detail panel, where:
   - MEDHIST is a page section with repeating items based on the master panel.
   - PRVMED is a page section with repeating items based on the detail panel.
   - PRVMEDCODE is a page section with no repeating items based on the detail panel.
3. A page template called HISTORY was created and the MEDHIST, PRVMED and PRVMEDCODE page sections placed on it in that order.

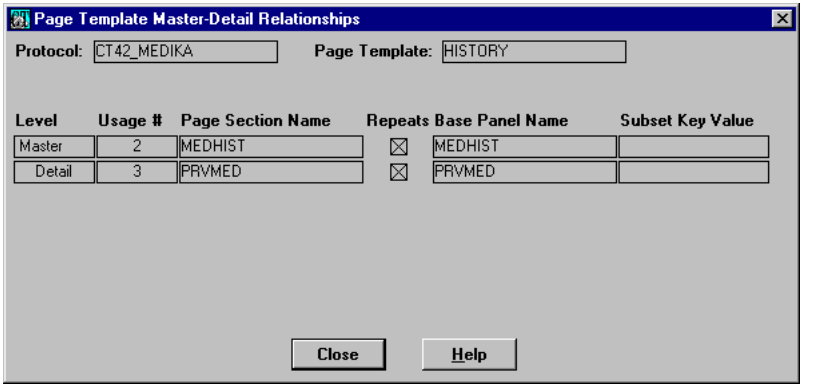
The MEDHIST and PRVMED page sections could be in a cross-panel master-detail relationship. The PRVMED and PRVMEDCODE page sections are in a within-panel master-detail relationship.

In such a scenario, when the data-entry person enters data for a medical condition under Medical History (MEDHIST page section) and then tabs out, a row appears under Previous Medications (PRVMEDS page section) that allows the user to enter information about the drug used to treat this condition. Because this detail page section is repeating, the data-entry person could also add more rows for additional drugs by selecting **Add Repeat** from the **Edit** menu. In this way, for each row under Medical History, the master-detail relationship could enable the Enter user to create a set of drug data (one row per drug) which is displayed under Previous Medications only when the master (Medical History) row is selected.

How to view master-detail relationships

To view the master-detail relationships between page sections in a page template, from the **Page Template** menu select **Show Master-Detail**. The Page Template Master-Detail Relationships window opens.

For the previous example, before the PRVMEDCODE page section is added to the page template, the page template's master-detail relationship window would appear as follows:



Creating page templates with subset page sections

What is a subset page section?

A *subset page section* is a page section based on a Type 0, Type 2, or Type 4 panel that can occur multiple times on a study page, with each different value of the subset key item representing distinct rows (subsets) of data. Each occurrence of a subset page section constitutes a different observation.

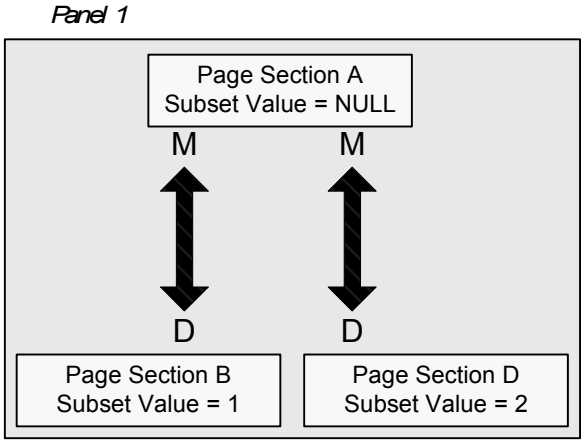
You can use subset page sections as follows:

- To establish a within-panel master-detail relationship across all subsets for that panel.

  A master-detail relationship exists between a page section that does not have a subset key value and any subset page sections for the same panel on the page template. Each detail page section displays a subset of the data, based on its unique subset key value.

  The following figure shows this relationship:

Page section A is master page section to detail page sections B and D (M -> D)



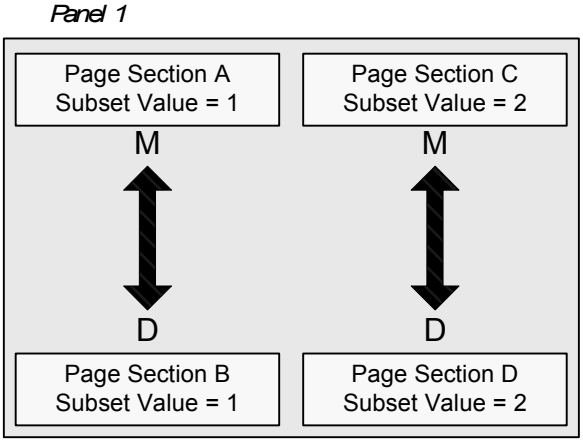
- To establish within-panel master-detail relationships within subsets, among pairs of subset page sections.

  A master-detail relationship exists between any two page sections that have the same subset key value. The first of each pair of page sections on the template is the master page section.

  The following figure shows this relationship:

Page section A is master page section to detail page section B. (M -> D)

Page section C is master page section to detail page section D. (M -> D)



- To display page sections based on the same panel, in which subsets of the data are displayed for each subset page section.

  No master-detail relationship exists. All page sections that are based on the same panel on this page template have unique subset key values. Each subset page section displays a subset of the data based on its subset key value.

Note: If you have a within-panel master-detail relationship across all subsets (master page section with no subset key value, detail page sections with unique values), you cannot have any within-panel master-detail relationships within subsets (each pair of page sections with the same subset key values).

The Medika sample study's LABLNG and LABSHT page templates use subset page sections for a within-panel master detail relationship. The page sections are based on the LAB panel.

For information on creating a panel with subset keys, see "Panels with subset key items" on page 216.

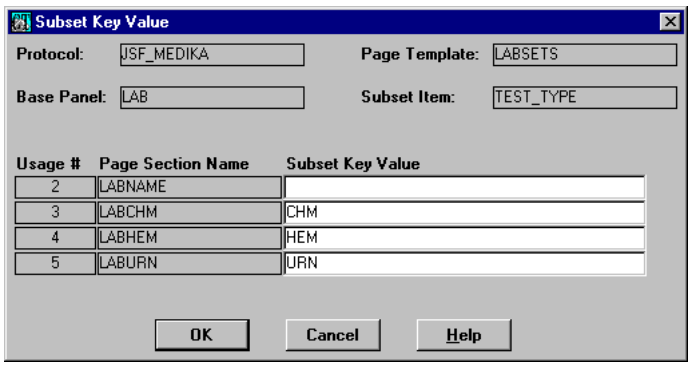
For information on creating subset page sections, see "Master page sections, detail page sections, and subset page sections" on page 485.

How to specify subset key values

To set up multiple subset page sections on a single page template, do the following:

1. Create a panel with a subset key item.
2. Create one or more page sections based on the panel with the subset key item.
3. Create a page template, and add the page sections with the same subset key item (as well as any other page sections that you want on that page template).
4. Click one of the subset page sections to select it.
5. From the **Page Template** menu, select **Modify Subset Key Value**.

   The Subset Key Value dialog box opens:



6. Enter values for the subset page sections in the Subset Key Value column as follows:
   - If you leave this field blank (null value) that subset page section becomes the master page section across all subsets, and the page sections with unique (non-null) values become the detail page sections in a within-panel master-detail relationship.

     In this example, the LABNAME page section is the master page section and the other page sections are the detail page sections.
   - If you assign two page sections the same value, you create a within-panel master-detail relationship within a subset, in which the first page section on the page template functions as the master, and the other page section functions as the detail page section for this subset.
   - If you enter a unique value for each page section, then no master-detail relationship is created.

   You cannot leave more than one value blank. Otherwise, an error message appears, telling you that you cannot have more than one master page section based on a panel with a subset key.

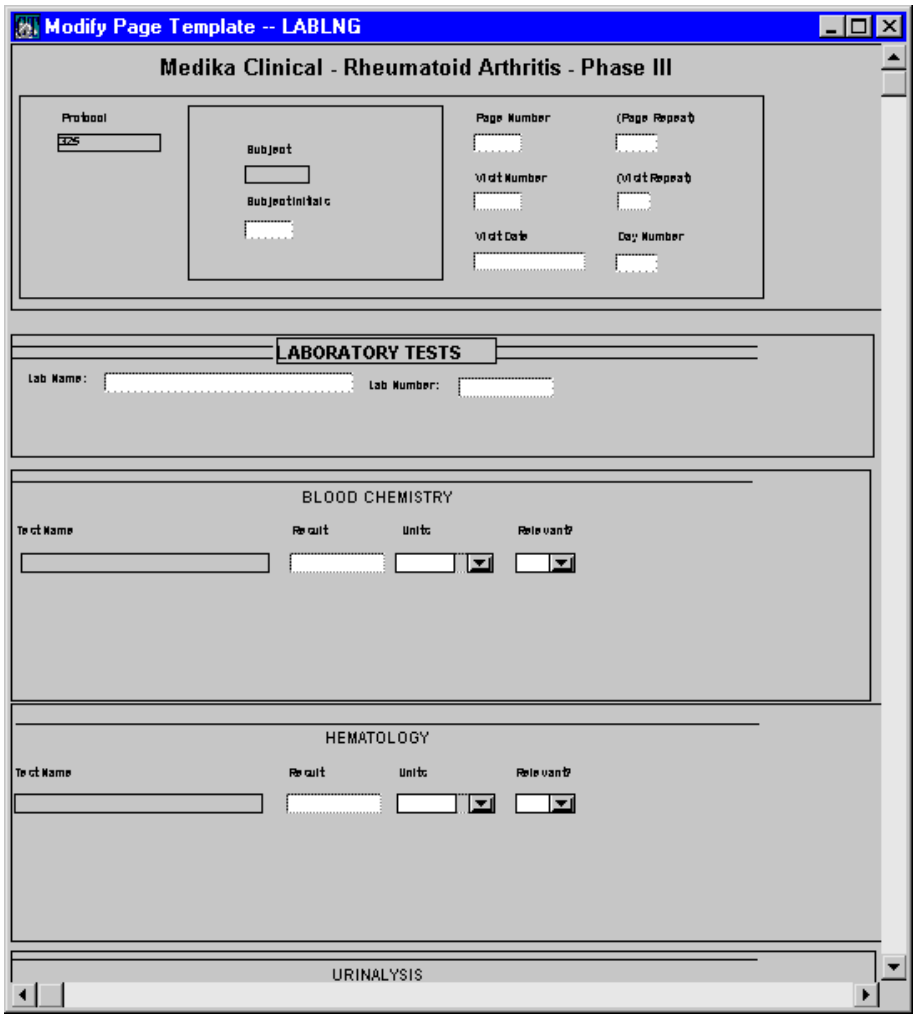
Caution: It is possible to change the values of subset key for a page template even if there is previously entered data associated with the page template. However, if subset key values are modified, the records entered prior to modification will be hidden from Enter. To avoid this problem, use Global Change to change the value of a subset item for previously entered records.

7. Click **OK** to save the subset key values, and then test and save the page template.

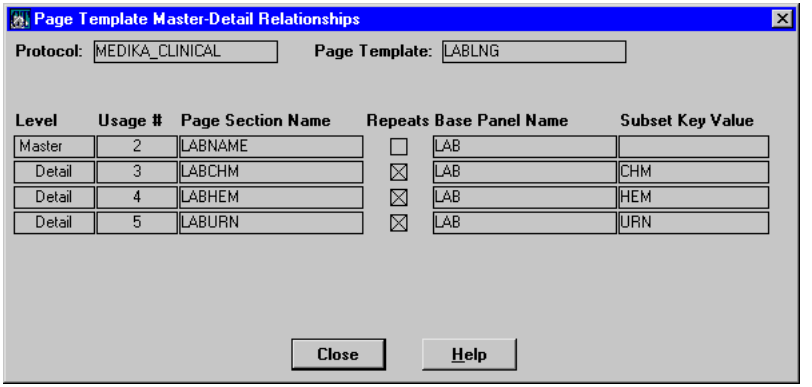
Example of subset page section

You can use subset page sections to group multiple sets of data having a similar format. For example, the Medika sample study contains three repeating page sections based on the panel LABLNG. This panel has the required item TEST_TYPE for a subset key. These three sections have the names LABCHEM (blood chemistry), LABHEM (hematology) and LABURN (urinalysis).

The following figure shows the LABLNG page template:



The following figure shows the master-detail relationships and the subset key values for the subset page sections on the LABLNG page template:



Note: This example also shows how a page section based on a panel with a subset key can be used for a within-panel master-detail relationship. Suppose you create a page section with general information (lab name, and so on.) based on the LABLNG panel. You can place this page section on a page template, assigning it a null subset key value, and then place the three test-type page sections and assign each one a unique (non-null) subset key value. In this case, the first page section automatically becomes a within-panel master page section, and the three test-type page sections become within-panel detail page sections.

Repeating items in master-detail relationships

Master:	Detail:	Description:
No repeating items	No repeating items	Each panel contains exactly one row.
No repeating Items	Repeating items	The master panel displays only a single row on the page, with corresponding rows displayed from the detail panel in a one-to-many relationship.
Repeating items	No repeating Items	Selection of a row in the master displays a corresponding row in the detail.
Repeating items	Repeating items	Selection of a row in the master displays multiple corresponding detail rows in a one-to-many relationship.

Master page section with no repeating items

If the master page section has no repeating items, when the page is displayed, one record is automatically inserted into the master page section. Any values entered in the master page section are propagated to the detail page section regardless of whether the detail page section has repeating items or not.

Master page section with repeating items

If the master page section has repeating items, the detail page section in a within-panel master-detail relationship cannot have any repeating items.

In a cross-panel master-detail relationship, the detail panel can have repeating items. The master panel displays all the applicable records. When you select one of those rows, the detail page section displays values for columns of that selected record. You can add records and delete records from the master page section, as you would for any page section with repeating items.

Repeating items in within-panel master-detail

The following table describes the possible combinations of page sections with and without repeating items for a within-panel master-detail relationship:

Master:	Detail:	Description:
No repeating items	No repeating items	Each page section shows data from different columns of a single record in the clinical data table.
No repeating items	Repeating items	Data entered into the single record displayed in the master page section is propagated to multiple records displayed in the detail page section.
Repeating items	No repeating items allowed	The master page section displays data from multiple records in the clinical data table. Selection of one row displays data from additional columns in that record.
Repeating items	Repeating items	Not allowed.

Repeating items in cross-panel master-detail

The following table describes the possible combinations of page sections with and without repeating items for a cross-panel master-detail relationship:

Master:	Detail:	Description:
No repeating items	No repeating items	Each page section shows data from a single record in its base panel's clinical data table.
No repeating items	Repeating items	The master page section displays data from a single record in the clinical data table. The clinical data table contains one record for each detail record.

Master:	Detail:	Description:
Repeating items	No repeating items	Selection of a row in the master page section displays data from a single record in the detail page section's base panel.
Repeating items	Repeating items	Selection of a row in the master page section displays multiple corresponding rows from the detail page section's base panel.

Note: When a master page section has repeating items, the master key item must have a unique value for each record.

Page template attributes

The following table lists the page template attributes:

Attribute:	Description:
Protocol	Name of the protocol in which the page template is created.
Page Template	Name of the page template.
Description	Description of the page template.

When you select the Page Template Browser's **Show** command, Clintrial displays the previously described page template attributes as read-only, and also displays the following additional read-only attributes:

Attribute:	Description:
Status	A status of Valid means that the page template can be used for a study page. A status of Invalid means that the page template cannot be used.
Modification Date	Date and time of the last modification to the page template.

Attribute:	Description:
Modifiable Status	Modifiable if the page template can be modified. Not Modifiable if the page template cannot be modified. Unmodifiable Copy if the page template was created by a connected copy from a page template where the Copies Not Modifiable attribute was selected.
Modified By	Name of the user account under which the page template was last modified.
Order	Order of the page sections in the page template, from top to bottom.
Usage #	A number that uniquely identifies each page section in a page template.
Page Section (Name)	Names of the page sections in the page template.
X	The X (horizontal) coordinate of the page section on the page template.
Y	The Y (vertical) coordinate of the page section on the page template.
Subset Value	Value assigned to the subset key item (if applicable).
Event	Type of page section or page template event that triggers the attached data-entry processing procedure.
Procedure Name	Name of the data-entry processing procedure attached to the page template or to a page section on the page template.

17 Page Section Layouts

Using page section layouts

What is a page section layout?

A *page section layout* determines the display of the page section, as viewed by the data-entry person. The page section layout can display:

- Fields for data entry that correspond to items in the page section's base panel
- Text to describe the fields or to provide other information
- Lines and rectangles to make the display clearer or more graphically interesting

How to create a page section layout

When you create a page section, Design creates a default page section layout automatically. When you save either the default or modified page section layout, Design creates a page section layout that can be used in Enter.

The following example shows the default layout of a page section for Vital Signs in the Medika sample study. The page section is based on the VITAL panel and contains the items for vital signs:

In this example, the default page section layout:

- Presents the items in a column format.

  A page section with nonrepeating items presents the items in a default column format. A page section with repeating items has a tabular default layout.

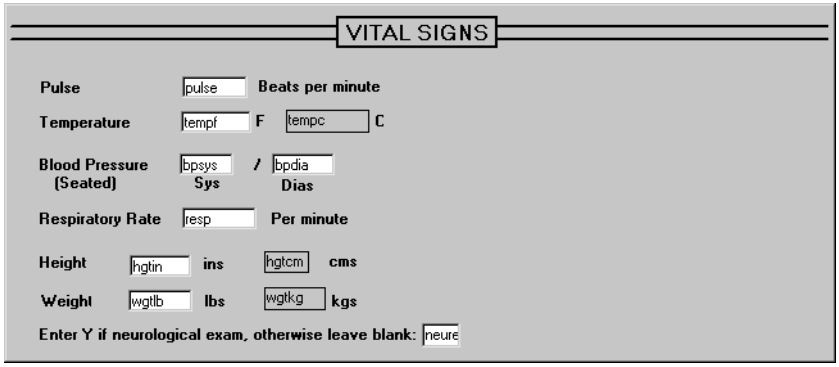
  Note: There is no requirement that a page section based on a Type 0, Type 2, or Type 4 panel must contain repeating items. However, if you create such a page section and do not specify any repeating items, the page section layout may sometimes display multiple rows of the non-repeating record. You can hide the multiple rows by sizing the page section layout to display the first row only.
- Creates fields for the items (the small boxes), except for derived items.

  The default size of a field is determined by the DB Format attribute of the item.
- Includes the name of the item in the item's field.
- Includes the text label to the left of the item's field.

  A page section with repeating items places the text above the item's field.

  The text label can be either the item name, or the item description, as set by the FRM_USE_DESCRIP system parameter.

The following example shows the page section layout after it has been edited to correspond more closely to the Vital Signs study page of the Medika Sample Study study book:



In this example, the modified page section:

- Includes the textual title Vital Signs on the page section in a larger font, and adds lines to set off the page section.
- Rearranges the placement of some of the items.
- Changes the text labels of the items to clarify their meaning.
- Adds additional descriptive text after each item.

You can also:

- Delete items from the page section layout if you do not want to display them.
- Display items previously deleted from the page section layout, or display derived items, which by default do not appear in the page section layout.

For more information on repeating and nonrepeating items with page sections based on specific panel types, see "Repeating items" on page 483.

How to modify or recreate a layout

The page section layout editor allows you to:

- Modify the design of all the elements in the page section layout (including items).
- Set controls that add elements to the page section layout which are used to make the page section more readable.

In addition to modifying the page section layout in the editor, you can do the following:

- Update the page section based on changes to the panel.

  When you make changes to a panel that cause a page section layout to become invalid, you must update the page section layout. In the Page Section Browser, from the **Page Section** menu, select **Update** to apply changes. Or, open and save the page section layout.
- Recreate the default page section layout. You may want to recreate the default page section layout for the following reasons:
  - You want to make major changes to the page section layout and the default is closer to what you want than the modified layout.
  - You want to ensure that all changes to the underlying panel are properly reflected in the page section layout.

  In the page section layout, from the **File** menu, select **Recreate**.

Design menu commands

What are the Design menu commands?

The page section layout **Design** menu commands determine:

- Attributes that determine how text and items are displayed in Enter

- Styles in which codelist data is entered and displayed in the item's field
- Sequences that display sets of values in a repeating item's field
- Alignment, spacing, and sizing of elements (text, items, lines, and rectangles) relative to each other in the page section layout
- Deletion of items from the page section layout, and addition of previously deleted items

List of Attribute commands

The page section layout Attributes determine how text and items are displayed in Enter. To set the attributes in the page section layout, click on an item or text. Then, from the **Design** menu, select **Attributes**. These attributes vary, depending whether you have selected text or an item.

The following table lists the page section layout Design Attributes:

Design Attributes:	**Description:**
For text:	
Alignment	Alignment within the text area. The text can be aligned left, right, or centered within the text area.
Font Size	Font size.
Value	The text itself.
For items:	
Alignment	Alignment within the field for the item. Select Left, Right, or Center.
Case	Select Any, UPPER, or lower.
Verify	Select to require verification; clear to not require verification. The panel Verify attribute sets the default for all items. If the panel Verify attribute is selected, you can override it for this item by clearing this attribute.

Design Attributes:	Description:
Enterable	Select to allow the data-entry operator to enter or edit a value for the item. Clear to disallow entry.
Autoskip	Select to have the cursor go automatically to the next field in Enter when the number of characters specified by the DB format for this item has been filled. Clear to require the data-entry operator to use the **Tab** key or mouse to exit the field. The ENT_AUTOSKIP_DFT system parameter and user preference must be set for autoskip to take effect. For information on user preferences, see "Setting your protocol parameters and user preferences" on page 148.
Multi-Line	Select to give the item's field a default of three lines, with word wrap enabled, and a scroll bar enabled when the field has focus in Enter. Clear to give the item's field a default of one line, with word wrap not enabled, and no scroll bar. ***Note:*** Although text items in Clintrial can have a Multi-Line attribute set, even those items without this attribute will accept carriage returns if their item heights are increased during page section design. If during data entry carriage returns are entered in such a field, the data may not be exported properly to SAS when SAS Export is performed in Retrieve. If you are planning to export data to SAS using Retrieve, you should use the Multi-Line attribute when designing page sections. N*ote:* Regardless of whether this attribute is selected or cleared, you can change the size of the item's field. But if the attribute is cleared, and you change the height of the field, any data entered will continue to be entered on a single line.
Override	Select to allow the data-entry operator to override minimum, maximum, and checklist values. Clear to disallow override.

Design Attributes:	Description:
Carry	Select to cause the entered value for a context item to carry forward from the previous study page in the block. Clear to not carry the value forward. *Note*: Even if selected, Carry does not apply to the first study page in the block.
Dup	Select to include the item as part of an automatic duplication group. Values for the group can be brought forward from the same study page for the same subject in the previous block, at the discretion of the data-entry operator. (The data-entry operator can enable automatic duplication by item whether Dup is set.) Clear to disallow inclusion as part of an automatic duplication group.
Default	You can enter a default value for the item.
Value Changed Procedure	This is a type of data-entry processing procedure associated with specific items in a page section. You can enter the name of a PL/SQL procedure or function called to supply the value for the item. If you did not create a public synonym for the procedure, function, or package, preface the name with the ORACLE account name for stored procedures and a period. ***Note*:** It is recommended that you specify a package containing the procedure or function; for example, CTSITEPROC.MY_PKG.PAT_INIT. ***Note*:** A value changed procedure cannot be associated with a derived item, except in an enrollment panel. For more information on data entry processing procedures, see the *Reference Guide*.
Help Text	You can enter a line of text that appears in the message area when the cursor is in this item's field.

The page section layout Style attributes determine how codelist data is displayed in Enter. To set the **Style** commands in the page section layout, click on an item that has an associated codelist. Then, from the **Design** menu, select **Style**.

The page section layout design styles apply only to items associated with codelists (not checklists):

Design Style:	**Description:**
Item Name	Name of the item to which the style is applies.
Item Style	Select one of the following values: • Drop Down List — Text, rather than checkbox or radio buttons. • Check Box — Only available for items for which there are two choices. • Radio Buttons — Only available for items for which there are five or fewer choices.
Enter as Code	Select to require that a codelist code is entered. Clear to require that a codelist value is entered.
Show Codelist Label	Select to display the short label for a code or value when the code or value is entered in a field in Enter. Clear to suppress the display of the short label.
Default	The initial (default) value assigned to this item. For a checkbox, this determines whether the checkbox is selected as the default display in Enter. If you specify the same value as for Value When Checked, the default display is selected. If you specify the other value, the default display is cleared. For radio buttons, determines which radio button is selected by default for display in Enter.
Value When Checked	For a checkbox, which of the two values to store when the checkbox is checked.
Check Box Label Text	The text associated with the checkbox.
Radio Button Layout	Select Vertical or Horizontal for an item displayed in Radio Button style.

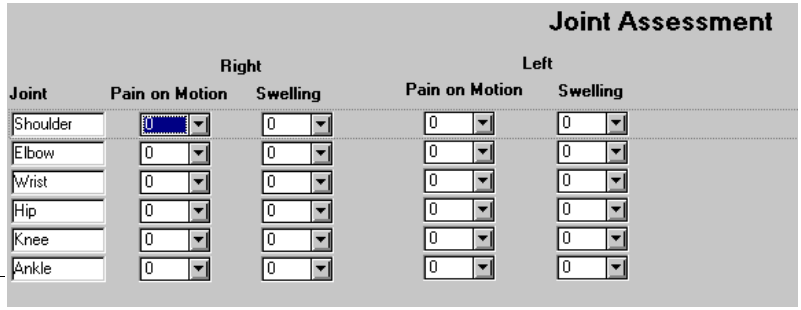
What is a sequence?

A *sequence* is a set of predefined default values for a repeating item. You associate a sequence with an item through the page section layout **Design** menu's **Sequences** command.

How repeating items relate to sequences

When you create or modify a page section, you can set up a sequence for any repeating item in the page section. For different repeating items, there can be different sequences within the same page section.

The following example shows repeating items with sequences in a page section:



In this example, the sequence supplies the values for the standard values for the repeating item "Joint." The data-entry operator enters values in the other repeating items. However, normal data-entry still applies to the item containing the sequence values. Therefore, the data-entry operator can edit or delete the sequence values, unless the page section layout Enterable Design Attribute is cleared. Also, the data-entry operator can add additional rows, as long as the Max Repeats attribute of the page section is not exceeded.

Note: An alternative to a single repeating item with a sequence is multiple items in a page section without repeats. Instead of multiple values for sequences in the Joints repeating item, each of the four types of pain could be a separate nonrepeating item, with the Joint as a text label.

In this example, the repeating items could be replaced by nonrepeating items specific to each combination of joint and condition. Instead of creating five items in the panel as JOINTS, LTPAIN, LTSWELL, RTPAIN, RTSWELL, defined as

repeating in the page section, you would create four items in the panel for each joint, such as SHOULDER_LTPAIN, SHOULDER_LTSWELL, SHOULDER_RTPAIN, SHOULDER_RTSWELL and so forth.

For information on panels and items, see Chapter 10.

What are aligned and crossed sequences?

When you create a sequence, you define it as either aligned or crossed. This determines how the sequence will combine with other sequences if there are multiple sequences in the page section.

How does an aligned sequence work?

If two sequences on a page section are *aligned*, the page section will include a repeat for each parallel set of sequence values. Each sequence must have the same number of values.

How does a crossed sequence work?

If two sequences on a page section are *crossed*, the page section includes a repeat for every combination of the sequence values.

The following example shows crossed sequences for two items:

- Time Post Drug—A repeating item with a sequence of 0, 2, 4, and 6 hours

- Blood Pressure—A repeating item with a sequence of Systolic and Diastolic

Study Monitoring - Vital Signs		
Time Post Drug	**Blood Pressure**	**Reading**
0	Systolic	_____
0	Diastolic	_____
2	Systolic	_____
2	Diastolic	_____
4	Systolic	_____
4	Diastolic	_____
6	Systolic	_____
6	Diastolic	_____

In this example, the sequence values for the repeating items Time Post Drug and Blood Pressure are crossed. Each time the blood pressure is taken, the time is noted twice, once for each of a pair of systolic and diastolic sequence values. The actual systolic and diastolic values appear as a third repeating item, Reading.

What are combined sequences?

If a page section includes one set of aligned sequences and one or more sets of crossed sequences, all sequences behave as if they are crossed. If a page section includes multiple sets of aligned sequences and one or more sets of crossed sequences, the alignments are performed first, and then the crossed sequences are performed.

The following example shows a combination of aligned and crossed sequences:

- Time—A repeating item with a crossed sequence containing the values AM and PM
- Test and Units—Two repeating items with aligned sequences

Laboratory - Clinical Chemistry			
Time	**Test**	**Result**	**Units**
AM	Glucose	_____	mg/dl
PM	Glucose	_____	mg/dl
AM	Sodium	_____	mEq/l
PM	Sodium	_____	mEq/l
AM	SGPT	_____	U/l
PM	SGPT	_____	U/l

In this example, the aligned sequences for the Test and Units items are set first (Glucose and mg/dl, Sodium and mEq/l, SGPT and U/l). Then the aligned sequences are crossed with the sequence for Time (AM and PM). As a result, each horizontal pair of values for the aligned sequences for items Test and Units appears twice to correspond to the values in the crossed sequence for the Time item. The repeating item Result contains the data.

How to initialize a sequence with codelists

When you create a sequence, you can either enter the sequence values or initialize the sequence with one or more codelists.

To initialize the sequence with a codelist, do the following:

1. In the page section layout, click on the item that you want to initialize with a sequence.
2. From the **Design** menu, select **Sequences**.

   Note: If the item is already associated with a codelist or checklist, that name appears in this codelist field and cannot be changed.
3. Select a codelist for the Codelist attribute, and click **Initialize**.
4. Optionally, click **Update** to append any new codelist values to the end of the sequence, as follows:
   - If you have deleted one or more values from the sequence after initializing it with a codelist and use **Update** with the same codelist,

Update reinitializes, adding the missing values to the end of the sequence.

– To initialize the sequence from two or more codelists, use **Initialize** to include the values from the first codelist, then select another codelist and use **Update**. The additional values are added to the end of the sequence.

Caution: If you use **Update** to initialize with multiple codelists, any codelist values that repeat are dropped from the initialization.

The buttons on the Define Item Sequence Values dialog box have the following functions:

- **Reorder** enables you to change the order of the values in the sequence.
- **Reset** clears the values from the sequence.

You can delete sequence values regardless of whether you set them by typing them or by initializing them with codelists. You can also edit these values unless the underlying item is associated with a codelist. In this case, editing is limited to selecting a different value from the drop-down list of codelist values.

List of Align commands

To align elements in the page section layout, click on two or more items, lines, rectangles, or text elements in the page section layout. Then, from the **Design** menu, select **Align**. The first element selected defines the alignment position.

The following table lists the page section layout **Align** commands:

Align:	Description:
Left	Aligns two or more selected items, lines, rectangles, or text areas to their left edges. Use to align vertical elements.
Right	Aligns two or more selected items, lines, rectangles, or text areas to their right edges. Use to align vertical elements.
Horizontal Center	Aligns two or more selected items, lines, rectangles, or text areas to their horizontal centers. Use to align vertical elements.
Top	Aligns two or more selected items, lines, rectangles, or text areas to their top edges. Use to align horizontal elements.

Align:	Description:
Bottom	Aligns two or more selected items, lines, rectangles, or text areas to their bottom edges. Use to align horizontal elements.
Vertical Center	Aligns two or more selected items, lines, rectangles, or text areas to their vertical centers. Use to align horizontal elements.

List of Size commands

To set the width and height of elements in the page section layout, click on two or more items, lines, or text elements in the page section layout. Then, from the **Design** menu, select **Size**. The first element selected defines the size used.

The following table lists the page section layout **Size** commands:

Size:	Description:
Width	Sets two or more selected items, lines, rectangles, or text areas to the same width.
Height	Sets two or more selected items, lines, rectangles, or text areas to the same height.

List of Spacing commands

To set the spacing of elements in the page section layout, click on three or more items, lines, or text elements in the page section layout. Then, from the **Design** menu, select **Spacing**. The spacing between the first two selected elements defines the spacing.

The following table lists the page section layout **Spacing** commands:

Spacing:	Description:
Horizontally Equal	Creates equal horizontal spacing among three or more selected items, lines, rectangles, or text areas. Use to space vertical elements.

Spacing:	Description:
Vertically Equal	Creates equal vertical spacing among three or more selected items, lines, rectangles or text areas. Use to space horizontal elements.

If you select elements that do not overlap each other in the dimension for which you are establishing uniform spacing, the spacing uses the outer edges to establish the relationship, as shown in the following example:

Before setting non-overlapping spacing

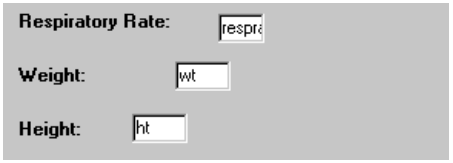
After setting non-overlapping spacing

Selecting items from top to bottom, after spacing horizontally equal, the space between the first and second items is replicated for the second and third items (and subsequent items, if more were selected). Vertical spacing works similarly.

If you select elements that overlap each other in the dimension for which you are establishing uniform spacing, the spacing uses the part of the two elements that does not overlap to establish the relationship, as shown in the following example:

Before setting overlapping spacing

After setting overlapping spacing



Selecting items from top to bottom, after spacing horizontally equal, the overlap of the first and second items is replicated for the second and third item (and subsequent items, if more were selected).Vertical spacing works similarly.

How to delete items and text

Use the page section layout **Design** menu's **Delete** command to delete items, text, lines, and rectangles from the page section layout. Deleted items and text do not appear in the study book.

Text, lines, and rectangles are not stored in the panel, so they are completely deleted when you use this command. Create new text, lines, and rectangles as needed with the page section layout **Controls** menu's commands.

If you delete an item, the item remains in the panel for the page section. The item remains available for reinsertion into the page section layout, also using the **Controls** menu.

Use the **Delete** command when you create multiple page sections based on the same panel, but want to display different text and different sets of items, depending on the page section. For example, in the MEDIKA study book, the page sections that accept data about drug administration and that are based on the DRGCMP panel display different combinations of text and items, depending on the subject visit.

Controls menu commands

List of Controls menu commands

Page section layout **Controls** menu's commands add elements to the page section layout that clarify the display in Enter. The following table lists these commands:

Command:	Description:
Text	A text field that you can place anywhere in the page section layout. You can modify the text field attributes, including font size, alignment within the text block, and the value (the text itself).
Horizontal Line	A horizontal line that you can place anywhere in the page section. You can move the line by keystrokes or with the mouse; you can modify the line length by keystrokes (as described in the Help).
Vertical Line	A vertical line that you can place anywhere in the page section. You can move the line by keystrokes or with the mouse; you can modify the line height by keystrokes (as described in the Help).
Rectangle	A rectangle that you can place anywhere in the page section. You can modify the rectangle attributes (size and position) by keystrokes and with the mouse (as described in the Help).
Item	Use this control to add an item to the page section layout. The item could be a derived item, which by default does not appear in the page section layout, or an item that you previously deleted when modifying the page section layout.

Command:	Description:
Computed Field	Use a computed field to calculate a value that is used in the page section but not entered in the database. You can specify an expression for computation and the alignment of the value within the field.
Modify Tab Order	Use this command to change the order in which the cursor visits items when tabbing through the page section in Enter. The default order is the item order of the items in the panel.
Page Section Attributes	Use this command to display the Page Section Attributes dialog box.

How to select, move, and resize layout elements

You select elements in the page section layout by using the mouse. You move and resize elements either by using the mouse or with keyboard strokes.

For information about selecting, moving, and resizing elements with the mouse or keyboard, see the Help.

Note: If a page section is not wide enough, when a flag and/or note is displayed in Enter, the red and/or yellow icon sits on top of the item for which the item's record flag or note was written. It may be necessary to widen the page section so that the flag or note indicators appear clearly on the page.

18 Study Books

Study books, blocks, and study pages

What is a study book?

In the Clintrial software, clinical data is recorded in *study books* – online representations of case report forms (CRFs). Each study book contains an ordered list of study pages, corresponding to the pages in a paper CRF. When you open a particular study book, you gain access to its study pages, which you can then work on and navigate as a set.

In addition to clinical data study books, the Clintrial software also includes enrollment study books, used to enroll study subjects, and non-subject study books, used to enter nonclinical data such as standard coding thesauruses or laboratory normal ranges.

What is a block?

A *block* is a group of related study pages in a study book. Blocks usually represent subject evaluation checkpoints, such as subject visits. For example, a block named VISIT1 can include the following study pages: Vital Signs, Concomitant Medications, and Laboratory Exams.

A block serves two purposes:

- A block replicates online the divider or tab from a CRF that typically represents a subject visit or some other time divider for sets of CRF pages.
- A block allows you to use a common page template for multiple study pages that are largely the same, except for the page number in the study book. For example, your study book might contain a study page for each visit that collects adverse reaction data.

What is a study page?

A *study page* is a data-entry window that represents a page in a CRF. A study page in which you can enter clinical data for a subject consists of a context page section (for context items), and one or more additional page sections (for clinical data). A study page is based on a page template.

For a description of page templates, see Chapter 16.

How to create a study book, block, and page

To create an empty study book you do the following:

1. With the Study Book Browser open, from the **Study Book** menu, select **Create**.

   The Create Study Book dialog box opens.
2. Specify the following study book attributes:
   - Study Book Name — The name that you want the study book to have.
   - Study Book Class — Normal, Non-patient Data, or Enrollment (drop-down list).
   - Description — A description of the study book.
3. When you have specified these attributes, click **OK**.

   The new study book is displayed in the Modify Study Book Layout dialog box, enabling you to add blocks and study pages.

To add blocks and study pages to a study book:

1. In the Study Book Browser, click on a study book, and from the **Study Book** menu, select **Modify**.
2. Add one or more blocks to the study book layout (a graphical representation of the study book).

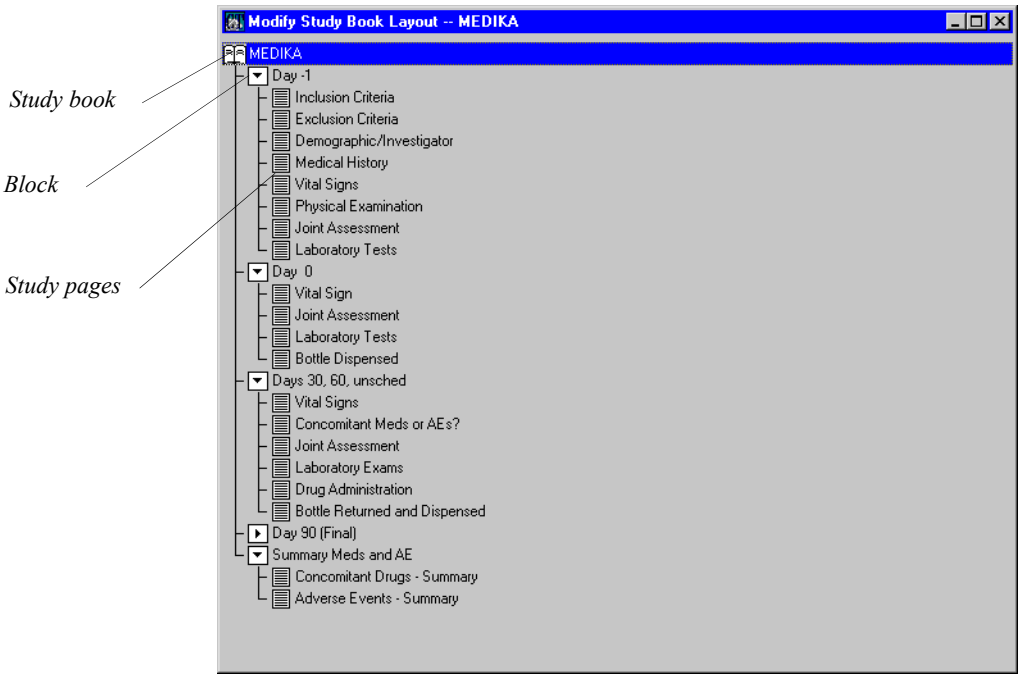
   If this is a repeating block and you plan to define the repeating block value in Design, you must specify the block key value and block repeat key value.
3. Optionally, specify any other block values.

   You can initialize values for block-related items.
4. Add one or more study pages to each block in the study book layout.

   If this is a repeating page and you plan to define the repeating study page value in Design, you must specify the page key value and page repeat key value.
5. Optionally, specify any other study page values.

   You can initialize values for page-related items.

The following figure shows the study book layout for the MEDIKA study book:



Just as a CRF consists of an ordered list of CRF pages, so the corresponding study book consists of an ordered list of study pages, within blocks.

Study books for non-patient data

Non-patient data is data that is not related to a particular subject or visit, such as standard coding thesauruses, view codelists or laboratory normal ranges. When you create a panel of the non-patient data type (Type 0 panel), the Clintrial software automatically creates the following additional objects:

- A page section based on this Type 0 panel
- A page template containing this page section
- A study book
- A block
- A study page based on this page template

Note: All of these objects have the same name as the Type 0 panel on which the page section is based.

To make an automatically-created Type 0 study page functional, you must do the following:

1. Create items on the Type 0 panel and install the panel.
2. Create and save a page section layout for the page section based on this panel.
3. Save the page template again, with this page section.

Note: All the information in this section on creating study books also applies to study books for subject enrollment data.

For more information about Type 0 panels, see "Non-Patient Data (Type 0)" on page 194.

Block values and page values

What is a block key value?

A *block key value* is the value assigned to the block key item, which must be a visit-related context item. Recall that you must specify a block key item in the context panel before you install the panel. You must specify the block key value when you create a block in the study book layout.

In a nonrepeating block, the block key value is the key for identifying data by block. In a repeating block, the block key value and the block repeat key value together are the key for identifying data by block. Typically, the block key item is the context item for identifying the visit.

The block key value (or block key value and block repeat key value) distinguishes among records you access through a page section that appears:

- On multiple study pages in a study book
- With each occurrence of the page section in a different block

For example, in the MEDIKA study book, the block key value for the first block is 0. The value 0 appears in the block key item (VISNO) for each study page in the block, uniquely identifying the block within the study book. (In the MEDIKA study book, the block with the block key value of 0 displays as Day -1).

When you create or copy a block, the block key value that you specify can include mixed case characters and Oracle reserved words. However, the block key value must be no longer than the context panel's designated block key item, and it must be compatible with the data type of the block key item.

What is a block repeat key value?

A *block repeat key value* is the value given to the optional block repeat key item, which must be a visit-related context item. You must specify the block repeat key value in Design when you create a repeating block in the study book layout editor, or in Enter when you create a repeating block.

Recall that you must specify a block repeat item in the context panel before you install the panel, if you plan to have repeating blocks.

The block repeat key values can be specified in either of the following ways:

- When you create or modify a block in a study book that has a block repeat item defined, check the Has Repeats attribute and enter a value for each planned instance of the repeating block. You can also set the Max Repeats attribute to limit to the number of repeats.

  The resulting study book in the Enter Navigator will have a predefined set of repeating blocks.

  Note: You can create a similar study book by creating identical non-repeating blocks.
- If you set the Max Repeats value to unlimited or to any number larger than the number of repeat key values that you predefine when you create or modify the block, the Enter user can add additional instances of the repeating block, up to the Max Repeats value.

  The Enter user is prompted for a block repeat key value for each repeat that is created.

What is a page key value?

A *page key value* is the value assigned to the page key item, which must be a page-related context item. You must specify a page key value when you create a study page in the study book layout. The page key value can be modified only if there is no clinical data that refers to the value.

Recall that you must specify a page key item in the context panel before you install the panel.

In a nonrepeating study page, the page key value is the key for identifying data by study page. In a repeating study page, the page key value and the page repeat key value together are the key for identifying data by study page. Typically, the page key item is the context item for naming or numbering the study page.

The page key value (or page key value and page repeat key value) distinguishes among records you access through a page section that appears:

- On multiple study pages in a study book
- With multiple occurrences of the study page in a single block

For example, suppose you had multiple non-repeating study pages based on a single page template for vital signs. If that identical page template were used once for each visit, the block key item would be sufficient to distinguish among study pages. However, suppose that vital signs are collected twice each visit, at the beginning and end of the visit. In this case, the block key item is the same for both study pages, and the page key item is required to distinguish between the two study pages in each block.

When you create a study page, the page key value that you specify can include mixed case characters and Oracle reserved words. However, the page key value must be no longer than the context panel's designated page key item, and must be compatible with the data type of the page key item.

What is a page repeat key value?

A *page repeat key value* is the value assigned to the optional page repeat key item, which must be a page-related context item. You must specify the page repeat key value in Design when you create a repeating page in the study book layout editor, or in Enter when you create a repeating page.

The page repeat key values distinguish among repeating instances of the study page. Recall that you must specify a page repeat item in the context panel before you install the panel, if you plan to have repeating study pages.

The page repeat key values can be specified in either of the following ways:

- When you create or modify a page in a study book that has a page repeat key item defined, select the Has Repeats attribute and enter a value for each planned instance of the repeating page. You can also set the Max Repeats attribute to limit to the number of repeats.

  The resulting study book in the Enter Navigator will have a predefined set of repeating pages.

  ***Note*:** You can create a similar study book by copying a page multiple times.

- If you set the Max Repeats value to unlimited or to any number larger than the number of repeat key values that you predefine when you create or modify the page, the data-entry person can add additional instances of the repeating page, up to the Max Repeats value.

  The data-entry person is prompted for a page repeat key value for each repeat that is created.

For example, in the MEDIKA study book, the Vital Signs study page is a repeating page. The first instance of the repeating page is the expected taking of the subject's vital signs at the beginning of the visit. If the vital signs are taken more than once during a visit, the page repeat key value allows for the creation of additional Vital Signs study pages in that visit.

Caution: Using the data type FLOAT to define key items in the context panel for study pages is not recommended. Using this data type may cause navigation problems between pages when using the Enter module.

How to specify nonkey block and page values

Optionally, you can create or modify initial values for visit-related or page-related context items that are not key items.

To specify a nonkey block value:

1. Select the block from the study book layout.
2. Then, from the **Study Book** menu, select **Modify Block Values**, and the specify the block value that you want.

To specify a non-key page value:

1. Select the study page from the study book layout.
2. From the **Study Book** menu, select **Modify Page Values**, and then specify the page value that you want.

For more information

For information on specifying block key items, block repeat key items, page key items, and page repeat key items, see "Special context items" on page 203.

For information on context panels, see "Context panels and context items" on page 199.

Study book, block, and study page attributes

List of study book attributes

You specify the study book attributes from the Study Book Browser. The following table lists the study book attributes:

Attribute:	Description:
Protocol Name	Name of the protocol in which the study book is being created.
Study Book Name	Name of the study book.
Study Book Class	Non-Patient — For a study book containing non-patient (Type 0) panels Normal — For a study book containing only Type 1, 2, 3, or 4 panels Subject Enrollment — For a study book containing enrollment (Type 5) panel
Description	An optional description of the study book.

When you select the **Study Book** menu's **Show** command, the Clintrial software displays the previously described study book attributes as read-only, and also displays the following additional read-only attributes:

Attribute:	Description:
Status	For future use.
Modification Date	Date and time of the last modification to the study book.
Modified By	Name of the user account that last modified the study book.
Modifiable Status	Modifiable if the study book can be modified. Not Modifiable if the study book cannot be modified. Unmodifiable Copy if the study book was created by a connected copy from a study book where the Copies Not Modifiable attribute was selected.

Attribute:	Description:
Database ID	Registration number of the database instance.
Study book layout	Graphic display of the study book layout, including the full tree of blocks and study pages.

List of block attributes

You specify the block attributes from within the study book layout. The following table lists the block attributes:

Attribute:	Description:
Protocol	Name of the protocol.
Study Book	Name of the study book containing the block.
Block Key Value	A value for this block for the item set as the Block Key Item — Uniquely identifies this block in the study book (unless this is a repeating block).
Block Title	Name of the block as it appears in the study book layout. The title appears in the graphical representation of the study book.
Has Repeats	Select to enable this block as a repeating block. Clear to disallow the use of this block as a repeating block. This attribute is available only if the context panel contains a block repeat key item.
Max Repeats	The maximum number of repeats allowed. Set to -1 for unlimited repeats.
Order	The order of the instances of the repeating block (specific to the block repeat). You can change this order by clicking the **Reorder** button.
Static	A nonenterable field on each row that you add or insert to contain the block repeat key value (specific to the block repeat). Static indicates that you have predefined the value when you create or modify the block in Design, rather than allow the data-entry person to define the value.

Attribute:	Description:
Block Repeat Key Value	A value that distinguishes this instance of the repeating block from other instances (specific to the block repeat).
Values Used	The number of block repeat key values that you have entered. The difference between Max Repeats and Values Used represents the number of additional instances of the repeating block that the data-entry person can create.
Max Repeats	The maximum allowed number of repeats of this block.

List of study page attributes

You specify the study page attributes from within a block in the study book layout. The following table lists the study page attributes:

Attribute:	Description:
Protocol	The name of the protocol.
Study Book	The name of the study book containing the study page.
Block Key Value	A value for the block that contains this study page, for the item set as the Block Key Item — Uniquely identifies this block in the study book (unless this is a repeating block).
Block Title	Name of the block that contains this study page. The title appears in the graphical representation of the study book.
Page Key Value	A value for the item set as the Page Key Item — Uniquely identifies this study page in this block (unless this is a repeating page).
Page Number	The text to appear on the printout of the study page to identify the page to the user. Typically, this is the same as in the CRF. The data type is textual so that values like “10-A” can be entered.
Page Title	Name of the study page. The title appears in the graphical representation of the study book.

Attribute:	Description:
Page Template	The page template used by this study page. A page template can be used by one or more study pages in one or more study books.
Help Context	A context number for the topic that you want to associate with the Help file for this study page, for use in Enter. If you create your own Help file, you can specify a context number for a Help topic specific to this study page. If you set the context number here, Help accessed from this study page uses the Help file you specified as the Help File Name protocol attribute, but does not use the default Help file topic or the topic set by the Help Context protocol attribute. For more information on setting Help context numbers for all study books in a protocol, see "Help Context" on page 177.
Has Repeats	Select to enable this block as a repeating page. Clear to disallow the use of this page as a repeating page. This attribute is available only if the context panel contains a page repeat item.
Max Repeats	The maximum number of repeats allowed. Set to -1 (the default value) for unlimited repeats.
Order	The order of the repeating study pages. You can modify the order by clicking the **Reorder** button (specific to the page repeat).
Static	A nonenterable field on each row that you add or insert to contain the page repeat key value (specific to the page repeat). Static indicates that you have predefined the value when you create or modify the page in Design, rather than allow the data-entry person to define the value.
Page Repeat Key Value	A value that distinguishes this instance of the repeating page from other instances (specific to the page repeat).

Attribute:	Description:
Values Used	The number of page repeat key values that you have entered. The difference between Max Repeats and Values Used represents the number of additional instances of the repeating page that a data-entry person can create.
Max Repeats	The maximum allowed number of repeats of this study page.

Part IV: Standardization in the Clintrial software

19 Revision Control

Overview

What is revision control?

Revision control is a set of features in Design that helps you enforce metadata consistency throughout the protocol hierarchies you create. By using revision control, you can guard against unwanted and unintentional inconsistencies between protocols.

You can use revision control for modifications to objects as follows:

- Establish a connection between a copied object (the destination object) and the object from which it is copied (the source object)
- Audit modifications to connected objects
- Propagate modifications to connected objects
- View connected objects and modifications to their metadata

Establishing connections between objects

What is a source object?

A *source object* is an object from which one or more other objects have been copied with connections.

What is a destination object?

A *destination object* is an object that has been copied from another object with a connection.

How to connect objects

You can copy objects within a protocol or between protocols in a database instance. When you copy an object, it is under revision control if the Establish Connection attribute is checked. If you clear this attribute, the copied object is not subject to revision control. If the protocol parameters specifying connected copy requirements for particular classes of objects are set to Yes, the copied

objects must be connected copies, and you cannot clear the Establish Connection attribute. For more information about particular parameters, see "How to specify required connections" on page 423.

If you copy an object that uses or contains other objects, those objects are also copied with connections. For example, if you copy a page template as a connected object, the following objects are also copied with connections:

- The page sections used in that page template
- The panels on which the page sections are based
- Any items, rules, and derivations contained in the panels (indirectly, through the panel connection)

Conversely, you can copy an object without establishing a connection, while specifying associated objects for which you need to establish a connection.

Note: In the connections established by revision control, the relationship of an object to a related object is determined by whether or not those objects share a container-to-contained relationship. For example, an item is a contained object in its container panel object, but a page section is not a container object in a page template that uses it.

For more information on unconnected copies, see "Copying panels as unconnected copies" on page 424.

What is a container object?

A *container object* is an object that contains other objects that exist only within the container object. For example, a panel is a container object, which contains rules, derivations, items, and coding targets

Contained objects cannot exist without a container object. There is a unique one-to-one relationship between a contained object and its container object.

Some objects use other objects in the protocol without the container-contained relationship. For example, a page template consists of one or more page sections. Therefore, a page template uses pages sections, but a page section is not a contained object in a page template. That is, the page section is defined outside of the page template, and any number of page templates can use the same page section. If you modify a connected page section, that page section is audited, but any page templates that use it are not audited.

The following table lists the objects you can copy with connections for revision control. If the object is contained in another object, the container object is also listed.

Object:	Container object:
Panel	N/A
Item	Panel
Rule	Panel
Derivation	Panel
Study book	N/A
Block	Study book
Study page	Study book, block
Page template	N/A
Page section	N/A
Thesaurus algorithm	N/A
Thesaurus language	N/A
Thesaurus view	N/A

Connected copy is read-only

When you perform a connected copy operation on a Clintrial software object, the destination object becomes, in effect, a read-only copy of the source object. When objects are connected in this way, you cannot modify the destination object except as follows:

- In a connected copy, you can modify the codelist attribute of a destination item to refer to a subset codelist of the source item's codelist.
- A view restriction on a view panel is always a connected copy from the base panel, but you can modify it.

Change propagation reflects this modification during any subsequent refreshes of the connection.

How to specify required connections

You can define whether connections are required connections between a source object and a destination object by setting the following protocol parameters:

- CONNECT_REQD_DD specifies whether Connected Copy mode is required for panels, items, thesaurus algorithms, thesaurus languages, and thesaurus views.
- CONNECT_REQD_FRM specifies whether Connected Copy mode is required for page sections, page templates, and study books.
- CONNECT_REQD_VLD specifies whether Connected Copy mode is required for rules and derivations.

These protocol parameters can have a value of No (the default) or Yes. A value of Yes means that Connected Copy mode is required when you copy objects of the corresponding type into the protocol. During a cascading copy operation, the settings of these parameters apply in combination. For example, if you set CONNECT_REQD_FRM to No and CONNECT_REQD_DD to Yes, and then copy a page section in Unconnected mode, the panel associated with that page section is copied in Connected Copy mode.

How to remove object connections

To remove a connection from an object that has already been copied in Connected Copy mode:

1. From the **Revisions** menu, select **Objects Copied Into**.
2. In the Objects Copied Into window, select the object and then, from the **Propagation** menu, select **Remove Connection**.

Note: If the corresponding CONNECT_REQD parameter is set to Yes, the Remove Connection command is disabled.

Note: After you remove an object from revision control by disconnecting it from its source object, you cannot reconnect the disconnected object.

Copying panels as unconnected copies

When you copy a panel in Unconnected mode, you can control which panel components are copied and whether they are connected copies as follows:

- Any panel component (item, rule, derivation) that has the Copy with Panel attribute selected is treated as a required component. When you copy a panel in Unconnected mode, all such components are copied in Connected Copy mode.
- When you copy a panel in Unconnected mode, a dialog box opens that lets you specify how the copying process handles optional components (those for which the Copy with Panel attribute is cleared and for which the CONNECT_REQD parameter is set to No).

Note: The Copy with Panel attribute applies only during an unconnected copy of a panel. This attribute has no effect when you copy a panel in Connected Copy mode since all components are copied connected, or when you copy individual components (for example, an item).

Note: If you copy a panel in Connected Copy mode and later remove the connection to that panel, a connection is then created for each panel component that has the Copy with Panel attribute selected.

For more information, see the following sections.

Examples of unconnected copies of panels

When you perform an unconnected copy of a panel using the Copy Panel dialog box, you can specify connected or unconnected copies of the panel's component objects. The panel itself is an unconnected copy, but each connected object becomes a distinct connected copy in the destination panel.

For example, you might specify a connected copy of all of a panel's items, but unconnected copies of all its rules and derivations. This means that none of the items in the destination panel can be deleted or modified. In the destination panel you can also:

- Add new items.
- Use existing rules and derivations without modification.
- Modify or delete existing rules and derivations.
- Add new rules and derivations.
- Propagate changes from some items.

Note: All items in the source panel with the Copy with Panel attribute selected are copied as connected copies, regardless of whether the panel copy mode is connected or unconnected.

In another example, you might copy a panel as an unconnected copy and also specify all panel components as unconnected copies. In this case, any objects with the Copy with Panel attribute become connected copies. The destination panel is identical to the source panel. In addition, the following conditions are true:

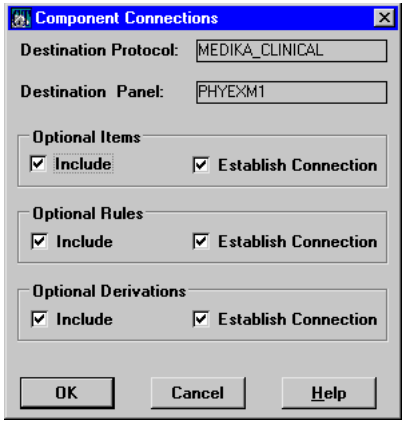
- None of the source components having the Copy with Panel attribute can be modified or deleted in the destination panel.
- Change propagation maintains any modifications to objects having the Copy with Panel attribute in the source panel.
- You can modify or delete all other (optional) components in the destination panel. (These components are not involved in change propagation.)
- You can add additional items, rules or derivations as needed.

How to copy panels as Unconnected

To copy a panel in Unconnected mode:

1. With the Panel Browser open, from the **Panel** menu, select **Copy,** select the panel you want to copy, and then clear the Establish Connection checkbox.

   The Component Connections dialog box opens:



   This dialog box contains a set of two checkboxes for each component type (item, rule, derivation).

2. Perform the following steps as needed:

- If you want *all* the optional objects of the specified type copied with the panel, select the appropriate checkboxes (Items, Rules, Derivations) in the Include Optional Objects area.

  The Include Optional Objects checkboxes are selected by default. Clearing these checkboxes means that no optional object of this type is copied. The status of this checkbox has no effect on the copying of component objects with the Copy with Panel attribute set.

- If you want the optional objects of the specified type to be copied as connected objects, select the Establish Connection checkbox.

  If you clear all of the checkboxes in the Include Optional Objects area, the Establish Connection checkbox is cleared automatically and none of the optional objects are copied as connected copies. However, this checkbox setting has no effect on the copying of component objects that have their Copy with Panel attribute selected.

3. Click **OK.**

   The panel copy operation proceeds according to the checkbox settings that you have selected.

Cascading copy operations

You can also copy a panel using a cascading copy operation, in which you specify the copy operation for an object associated with the panel. For example, if you copy a page section and the base panel for that page section does not exist in the destination protocol, the base panel is also copied. This cascading copy operation executes as follows:

- If you specify a connected copy of the page section, then its base panel is also copied in connected mode.
- If you specify an unconnected copy of the page section, then the panel is copied in connected mode if CONNECT_REQD_DD is set to Yes. Otherwise, the panel is copied in Unconnected mode, but any objects contained by the panel with the Copy with Panel attribute selected are copied as connected objects. (All other associated objects are copied in Unconnected mode.)

  Note: In the latter example, the Component Connections dialog box is not available. If you want more control over the details of an unconnected copy of a panel, you must first copy the panel, and then copy the page section or other associated object.

Auditing modifications to connected objects

What objects are audited?

When you modify connected objects (source objects only), Design can audit the modifications. The auditing of modified connected objects depends upon whether the objects have a container-to-contained relationship. When you copy a container object with a connection, the contained objects may or may not have their own connections. However, modifications to the contained objects are reflected in the auditing of the container object.

For example, each item or rule is defined in part by its singular relationship with a particular panel (although copies of the item or rule may have similar relationships with other panels). If you copy the panel with a connection, an item in the panel may or may not have its own connection. If it does not, and you then modify either the panel or the item, the panel is audited.

For information on how items in a connected panel may or may not have its own connection, see "Copying panels as unconnected copies" on page 424.

How to enter a reason for modification

If you establish revision control by connecting objects, and then modify a source object, you can record a reason for the modification for the modified object.

If the protocol parameter AUDIT_METADATA is set to Yes, when you modify a connected source object or destination object, you must record a reason for modification. Design prompts you for the reason for modification in the Audit Objects dialog box. Type a reason for modification or select a reason from the Clintrial software-supplied CTS_REVISE_REASON codelist.

Note: If you are planning to use Clintrial Resolve to Edit Source Data, the Reason for Change must be limited to 20 characters.

The CTS_REVISE_REASON codelist contains a single value, UNSPECIFIED. You can modify the codelist values to specify your own reason for change values.

If AUDIT_METADATA is set to No, you are not prompted for a reason for change.

Propagating modifications to connected objects

How to refresh destination objects

You can propagate metadata modifications made to the source object by refreshing the destination object.

To propagate the modification, do the following:

1. From the Objects Copied Into window in the protocol containing the destination object, select the object.
2. From the **Propagation** menu, select **Refresh Destination**.

If you are refreshing a destination panel, the source panel must be installed and not marked for revision. If you are refreshing an item, rule, or derivation in a destination panel, the source panel must be installed, and the destination object's containing panel must be deinstalled or marked for revision.

Viewing connected objects and their modifications

After you establish revision control connections, you can do the following:

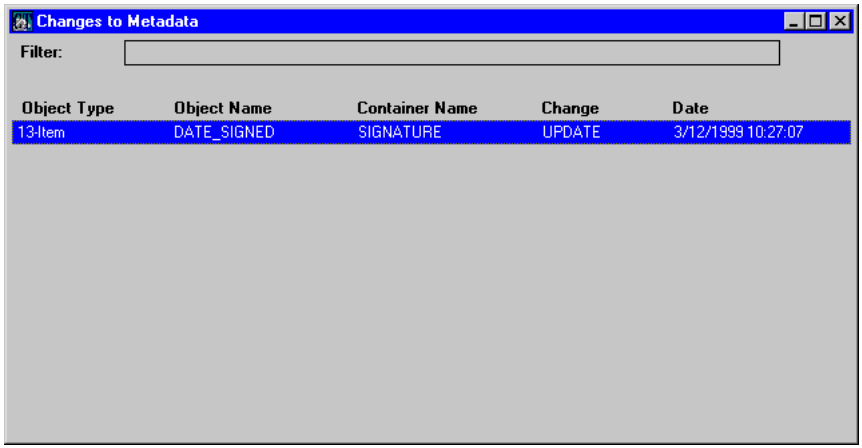
- View a list of objects in this protocol whose metadata has been modified.
- View a list of objects copied into a protocol with connections.

The following sections describe how to view the connections and the modifications to connected objects.

How to view metadata modifications

You can view a list of objects in a protocol that were modified and are being audited for modification, as well as view the reasons for modification.

To view the list of modified objects, from the **Revisions** menu, select **Changes to Metadata**. The following example shows the Changes to Metadata window:



The Changes to Metadata window shows one row for each change to an object. To show audit information for that change, select an object. Then, from the **Propagation** menu, select **Show Details**.

Note: The Changes to Metadata window displays only audited modified objects.

Consider the following information about audited and not audited objects:

- The value of the protocol parameter AUDIT_METADATA determines whether connected objects are audited when they are modified.

  If the protocol parameter AUDIT_METADATA has a value of Yes, Design prompts for a reason for modification when the object is modified. Then, the modified object is listed in the Changes to Metadata window. If the protocol parameter AUDIT_METADATA has a value of No at the time the object is modified, no reason for modification can be entered when the modification is made, and the object is not listed in the Changes to Metadata window.
- Modifications to connected panels, and to a panel's contained objects (items, rules, derivations, and coding targets), can be audited both for installed and uninstalled panels.
- Objects whose metadata is modified by means of the **Refresh Destination** command are not audited objects, and do not appear in the Changes to Metadata window.

  The **Refresh Destination** command modifies the metadata of destination objects from within the protocol that contains those objects. The modifications made to the metadata of destination objects are based on

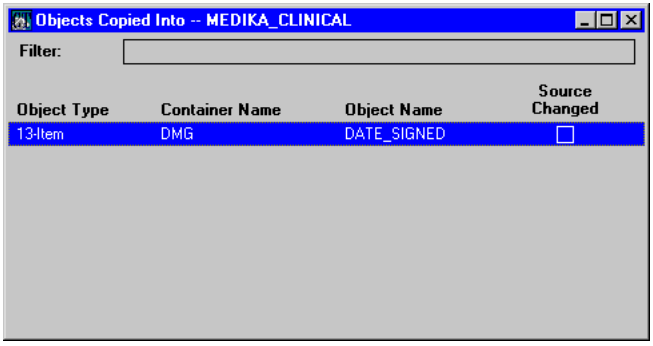
modifications to the metadata of the source objects from which they were copied. The source objects may or may not be audited at the source protocol.

For information on using the **Refresh Destination** command, see "How to refresh destination objects" on page 428.

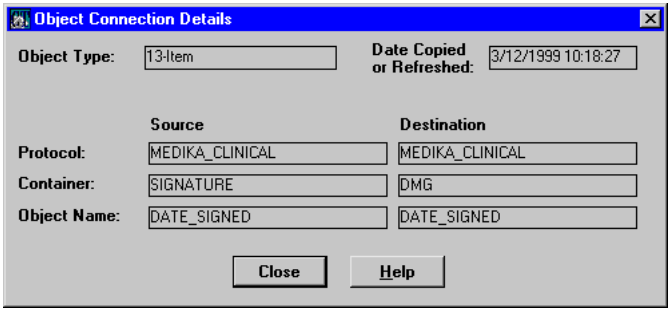
How to view objects copied into a protocol

You can view which objects in a protocol were created by copying from this protocol or another protocol with a connection. To view a list of the connected objects, from the **Revisions** menu, select **Objects Copied Into**. The Objects Copied Into window displays the objects that were created in this protocol by copying with a connection from this protocol or another protocol. If an established connection is removed, the Objects Copied Into window no longer displays the destination object.

The following example shows the Objects Copied Into window:

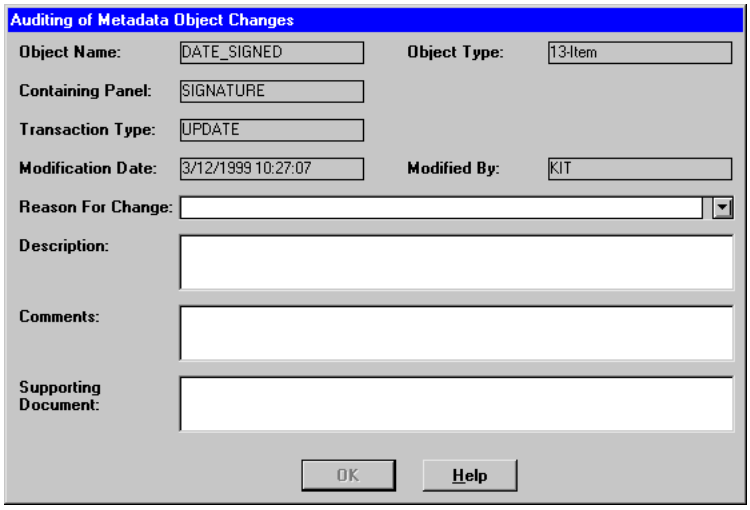


To view more information about the connections, select an object. Then, from the **Propagation** menu, select **Show Details**. The Object Connection Details dialog box opens:



To view information about the reasons for modification for a source object, select an object in the Objects Copied Into window with the Source Changed column selected. From the **Propagation** menu, select **Show Source Audits**. Design opens the Auditing of Metadata Object Changes windows, a succession of one or more windows containing the reason for change information for the source object of the object that you selected.

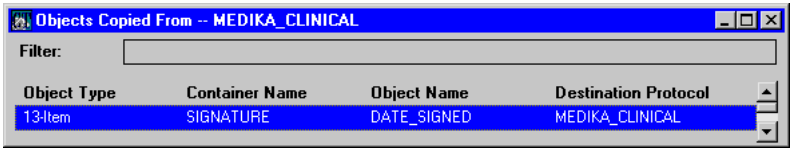
The following example shows the record in the Auditing of Metadata Object Changes window for the modified DATE_SIGNED item:



How to view objects copied from a protocol

You can view which objects in a protocol were used as source objects. To view a list of the connected objects, from the **Revisions** menu, select **Objects Copied From**. The Objects Copied From window opens, displaying the objects that were copied as source objects from this protocol, to this protocol or another protocol.

The following example shows the Objects Copied From window:



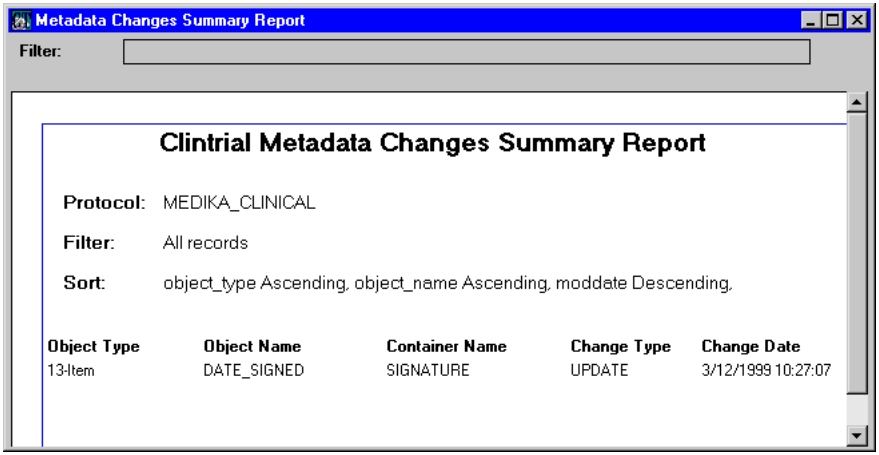
For more information about the connections, select an object. Then from the **Propagation** menu, select **Show Details**. The Object Connection Details window opens, displaying detailed information about the modified object.

How to view metadata change reports

Design provides two metadata modification reports for you to view audited modifications to metadata for connected objects:

- Metadata Changes Summary Report

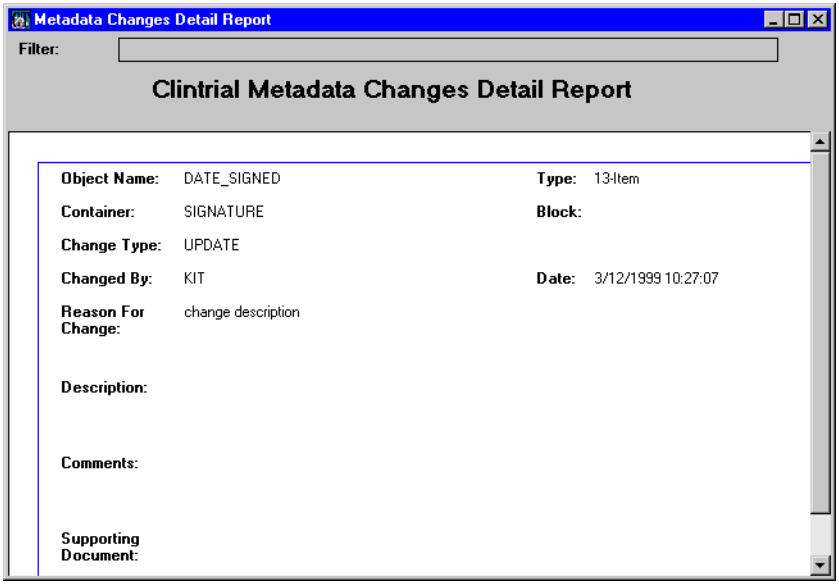
  The following example shows the Clintrial Metadata Changes Summary Report:



  To display this report, from the **Revisions** menu, select **Changes to Metadata**. Then, from the **Reports** menu, select **Revision Summary**.

- Metadata Changes Detail Report

  The following example shows the Clintrial Metadata Changes Detail Report:



To display this report, from the **Revisions** menu, select **Changes to Metadata**. Then, from the **Reports** menu, select **Revision Details**.

20 Codelist and Protocol Export and Import

Overview

The Clintrial software allows you to transfer both codelists and protocols (containing only metadata or both metadata and clinical data) between Clintrial software database instances or within a database. These export and import utilities make it possible for you to:

- Facilitate work with contract research organizations (CROs) — For example, by sharing completed protocols or by sharing the burden of building new protocols.
- Archive studies.
- Share codelists.

You can also share protocols and codelists by using the metadata distribution feature of the Multisite module. For more information, see *Multisite* and the Multisite on-line Help.

Export and import utilities

There are two methods of exporting and importing codelists and protocols supported in Clintrial.

The recommended method utilizes Oracle 10g's Data Pump Export and Import utilities (named **expdp** and **impdp** by Oracle). The new features for these utilities are listed below. These utilities support all Oracle 10.2 features except for XML schemas and XML schema-based tables. The design of these utilities results in greatly enhanced data movement performance over the Export and Import utilities used prior to Oracle 10g.

The following are the major new features that provide this increased performance, as well as enhanced ease of use:

- The ability to specify the maximum number of threads of active execution operating on behalf of the Data Pump job.
- The ability to restart Data Pump jobs.
- The ability to detach from and reattach to long-running jobs without affecting the job itself.
- Support for export and import operations over the network, in which the source of each operation is a remote instance.
- The ability, in an import job, to change the name of the source datafile to a different name in all DDL statements where the source datafile is referenced.
- Enhanced support for remapping tablespaces during an import operation.

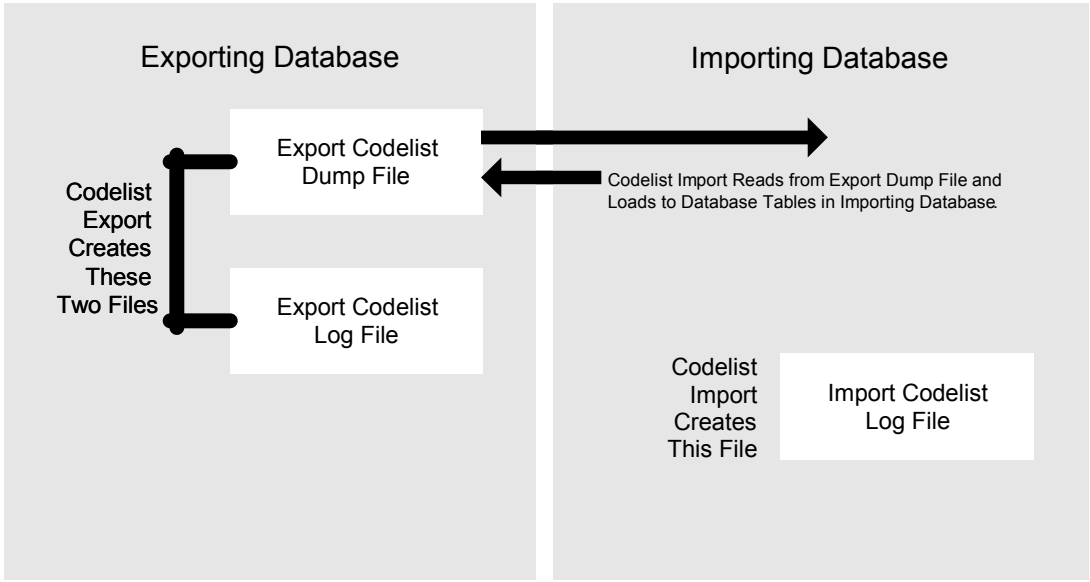
- Support for filtering the metadata that is exported and imported, based upon objects and object types.
- Support for an interactive-command mode that allows monitoring of and interaction with ongoing jobs.
- The ability to estimate how much space an export job would consume, without actually performing the export.
- The ability to specify the version of database objects to be moved. In export jobs, VERSION applies to the version of the database objects to be exported.
- Most Data Pump export and import operations occur on the Oracle database server.

Note: If you are exporting then importing a non-unicode database to an Oracle 10.2 or 11.2 installation, you are required to use the expdp and impdp utilities.

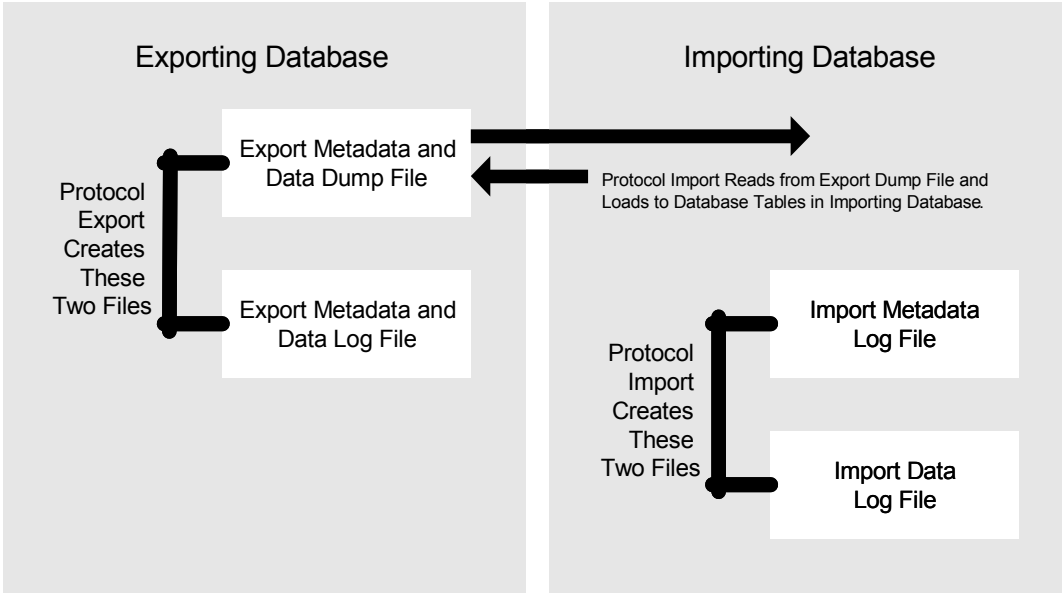
Beginning with release 4.6, the Clintrial software uses the Oracle 10g Data Pump Import/Export utility by default. Clintrial 4.7 uses this method, as well as the previous export and import Oracle utilities **exp** and **imp**. These utilities are called **Legacy Export** and **Legacy Import** in Clintrial 4.7.

For more information on the Oracle 10g Data Pump Import/Export utilities, see the Oracle document, *Oracle Database Utilities 10g Release 2 (10.2)*, Part 1, "Oracle Data Pump", Part Number B14215-01.

The following figure shows how codelists are exported from one database and imported to another using Oracle Data Pump Export and Import:



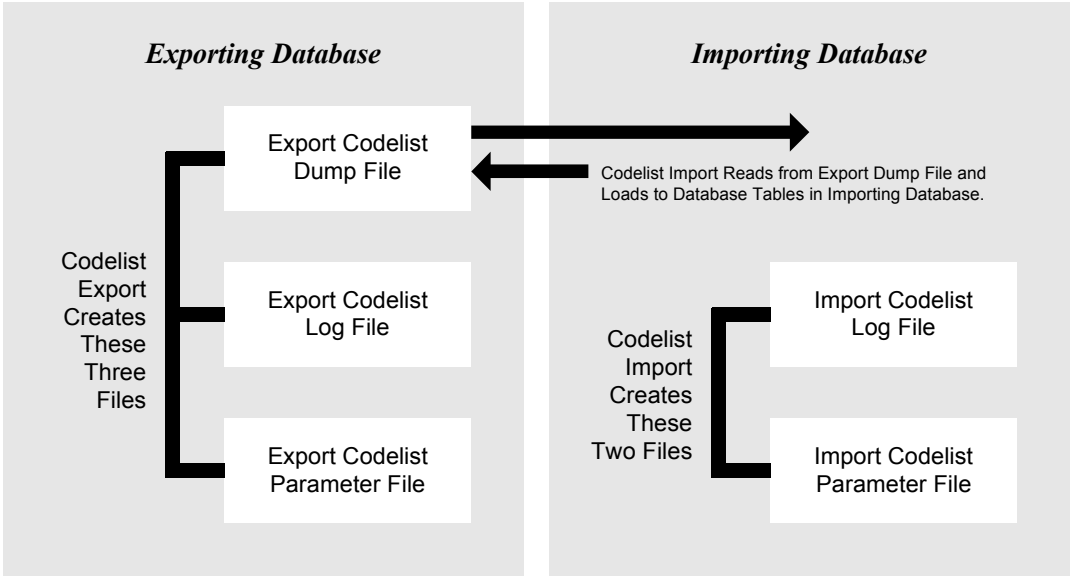
The following figure shows how protocols are exported from one database and imported to another using Oracle Data Pump Export and Import.



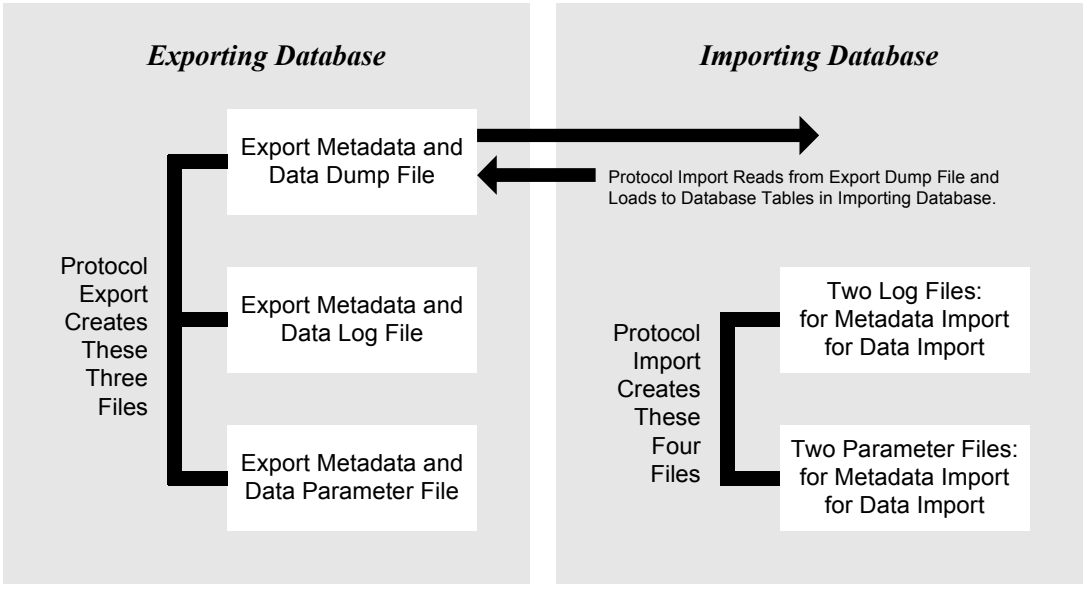
The second method supported for export and import utilizes the Oracle utilities named **exp** and **imp** by Oracle. These utilities support all Oracle 9.2 features. These utilities are called **Legacy Export** and **Legacy Import** in Clintrial 4.7.

For codelists and protocols exported from previous versions of Clintrial, you must import using the **Load Legacy Files** menu selection under the Design Import menu. Legacy exported protocols may include metadata and clinical data or only metadata. Using these utilities, protocol metadata and clinical data were exported to separate files. Using the Data Pump utilities, both metadata and (optionally) clinical data are exported to a single file.

The following figure shows how codelists are exported from one database and imported to another using the Oracle Export and Import utilities (Legacy Export and Import):



The following figure shows how protocols are exported from one database and imported to another using the Oracle Export and Import utilities (Legacy Export and Import):



Preparing to export or import codelists and protocols

Types of required Clintrial access rights

To export or import codelists, your Clintrial user account must have the Design non-protocol Global access right, with the Full access level.

To export or import protocols, your Clintrial user account must have the Design non-protocol System access right, with the Full access level.

For more information on access rights and access levels, see Chapter 4.

What is required for using the Oracle 10 Data Pump Export and Import utilities?

To import or export a file using the Oracle Data Pump Import and Export Utilities, you must have:

- The utilities installed on your server computer.
- Access to the shared directories on the Oracle data base server used for import and export dump and log files.

To export codelists and protocols, you must have been granted:

- WRITE access in Oracle to the target Dump Directory.
- WRITE access in Oracle to the target Log Directory.

To import codelists and protocols, you must have been granted:

- READ access in Oracle to the target Dump Directory.
- WRITE access in Oracle to the target Log Directory.

Note: In addition to the proper Clintrial and Oracle accesses, if you are using the Linux or Unix operating system on the Oracle data base server, you must be sure that the export/import dump and log directories are shared and you have created a symbolic link to them. You may do this using an application such as Samba on a Linux server, or by setting up NFS.

For more information on creating Oracle directories, and suggestions on using Samba, see **Chapter 2: Preparing to install or upgrade the server to Clintrial 4.7** in the Clintrial 4.7 *Getting Started* guide.

What is required for using the Oracle Legacy Export and Import utilities?

To import or export a file, you must have:

- the Oracle Import/Export Utilities installed on your client computer.
- Your path must include the directory where the executable export and import programs reside (as well as any required DLLs).

Limitations on protocol export and import

You cannot export or import view protocols.

Exporting protocols from a Clintrial 4.5, 4.6, or 4.7 database instance, and then importing them into a Clintrial 4.7 database instance by using the Oracle Import/Export utility requires many steps.

Oracle recommends that you upgrade the entire database instance.

If you perform a full database export from a non-Unicode database in a Clintrial 4.5 or Clintrial 4.6 application, and import to a Unicode database in a Clintrial 4.7 application, you must run the prv_sys.sql script to re-grant Select privileges on the SYS tables for the core modules.

If you are using the Multisite module, you must also perform these grants:

```
grant select on sys.dba_db_links to prv$wxrm;
grant select on sys.dba_tablespaces to prv$wxrm;
grant select on sys.dba_objects to prv$wxrm;
grant select on sys.dba_source to ctsrm;
```

For more information, contact the Global Support Center.

Exporting codelists using the Oracle 10 Data Pump utility

Why export a codelist?

Typically, you export codelists to:

- Accompany the export of a protocol and items in that protocol which refer to one or more codelists that do not exist at the importing database.
- Facilitate standardization of codelists across databases.

How to choose codelists to export

The first step in exporting one or more codelists is to identify the appropriate codelists. You can select either specific codelists for export, or select all codelists associated with items in a specific protocol.

To export codelists, with the Codelist Browser open, from the **Codelist** menu, select **Export**. Then, select one of the following:

- **Codelist**

  If you select Codelist, select from the list one or more codelists you want to export.
- **Protocol**

  If you select Protocol, select from the list one or more protocols. Design exports any codelists attached to items in the protocol or protocols.

What happens at export to the dump file and log file?

When you export a codelist, Design creates two files to store the exported codelist data and the log information generated by the codelist export. You can accept the default names or specify different names. The files are:

- A dump file to which the codelist data is written. The default name of the dump file is EXPORT_C.DMP.
- A log file to which the log information is written. The default name of the export log file is EXPORT_C.LOG.

Exporting codelists using the Oracle Export Legacy utility

This section describes exporting codelists using the Oracle Export Legacy utility.

Why export a codelist?

Typically, you export codelists to:

- Accompany the export of a protocol and items in that protocol which refer to one or more codelists that do not exist at the importing database.
- Facilitate standardization of codelists across databases.

How to choose codelists to export

The first step in exporting one or more codelists is to identify the appropriate codelists. You can select either specific codelists for export, or select all codelists associated with items in a specific protocol.

To export codelists, with the Codelist Browser open, from the **Codelist** menu, select **Export**. Then, select one of the following:

- **Codelist**

  If you select Codelist, select from the list one or more codelists you want to export.

- **Protocol**

  If you select Protocol, select from the list one or more protocols. Design exports any codelists attached to items in the protocol or protocols.

What happens at export to dump file and log file?

When you export a codelist, Design creates two files to store the exported codelist data and the log information generated by the codelist export. You can accept the default names or specify different names. The files are:

- A dump file to which the codelist data is written. The default name of the dump file is EXPORT_C.DMP.
- A log file to which the log information is written. The default name of the export log file is EXPORT_C.LOG.

When you export the codelists, Design also creates a parameter file that stores the Oracle parameters required in the export. The parameter file is written to the default directory (CT_TEMP_PATH), with the file name EXPORT_C.PAR. After exporting, you can manually delete the parameter file, which is recreated with each import.

Once the export process creates the parameter file and executes the Oracle Export utility, the codelist export runs independently of Design. The Oracle Export utility opens the EXPORT Status window, which displays the same information that is written to the log file.

All codelists are loaded into the CXFR_SEND account.

How to clear the export account

After you export the codelists, the CXFR_SEND account continues to store the exported codelist data. Design does not delete this data automatically. When you no longer need to export from the account, you can clear the account to delete the contents.

Importing codelists using the Oracle 10 Data Pump utility

Steps in importing a codelist

Note: It is recommended that you import the codelists that a protocol uses before you import the protocol. If you import the protocol first, and the codelists that it uses are not in this database instance, you must reconcile the codelists after creating or importing them.

To import a codelist, with the Codelist Browser open, from the **Codelist** menu, select **Import**. Then, do the following:

1. Load the file.

   Note: A default worksheet is created automatically.
2. Edit the worksheet (if needed).
3. Integrate the codelists.

   Note: You may choose to do this as a last step in loading a codelist.

Note: If you are importing any view codelists or subset codelists, ensure that the necessary base codelists (for subset codelists) or base panels and tables (for view codelists) already exist at this site. It is also possible to import base codelists in the same session.

What happens at file loading?

When codelists are exported, the Clintrial software creates an export dump file to store codelist information from the exported codelists, as well as an export log file. When you import codelists, the Clintrial software:

- Reads the exported dump file.

  Design provides the placeholder name IMPORT_C.DMP. The Clintrial software client computer must have access to the export dump file. The dump file must reside on the server in the path specified by the chosen Oracle directory. The browser can be used to select the filename, or the filename may be typed in.

- Writes to a log file the Oracle messages resulting from the import.

  Design provides the placeholder name IMPORT_C.LOG. The log file must reside on the server in the path specified by the chosen Oracle directory. The browser can be used to select the filename, otherwise the filename may be typed in.

  Note: If you give the import log file the same name as the export log file, the import log file will overwrite the export log file.

  Note: Only the filename is read from the filename field. The path is taken from the directory field. You may browse to the correct file, or write it in.

  When you load the imported protocol data, Design runs the Oracle Import utility. The data import runs independently of Design.

  All codelists are loaded into the CXFR_RECV account. If the receiving account is not empty, you are prompted to confirm deletion.

- Creates the default worksheet
- Integrates the imported codelists with the existing codelists, if specified when loading.

How a default worksheet matches codelists

After loading in the dump file, Design creates a default worksheet. The default worksheet attempts to match imported codelists with those in the current database based on:

- Previous imports

  You can specify Previous Imports and select at least one row from the list of previous imports. Design associates codelists as they existed in the previous import or imports.

- Name and data type

  If you do not use a previous import as the basis for codelist association, the Clintrial software attempts to associate the imported codelists with codelists that have the same names and data types (for the code and value items).

How to edit the worksheet

When you edit the worksheet, you can specify for each imported codelist:

- Action

  Omit — In this case, Design does not install the imported codelist. You might select this option if, at the exporting database, you selected more codelists for export than you currently want to import.

  Associate — In this case, a new codelist is not created. During protocol import Design can substitute item references to codelists at the exporting database with references to existing codelists at the importing database. Items that were exported and then imported now refer to a codelist that already existed at the importing database.

  Install — In this case, Design creates a new codelist at the importing database.

- Target Codelist

  Specify an existing codelist with which to associate the imported codelist, or name a new codelist to be created from the imported codelist.

  The worksheet also displays:

- Name Match — A codelist with the same name is found.
- Type Match — A codelist with the same name and matching data type is found.
- Status — A codelist can have a status of Valid or Invalid.

How to integrate the codelists

If you chose not to integrate the codelists at loading, you may do it as an independent step. When you integrate the codelists, Design checks to see that the selected integration options are acceptable before you can proceed. For example, if you indicate that you want to install an imported codelist as a new codelist, Design verifies that the codelist has a unique name. Similarly, if you indicate that

you want to associate an imported codelist with an existing codelist, Design verifies that the code field type and value field type for both codelists have the same data type.

You continue this process of editing and verifying the codelist import worksheet until you are satisfied that you have selected the correct integration options.

Design verifies that the attributes of two codelists are compatible. For example, Design can prevent you from integrating two codelists whose respective Code Field Type attributes have the same data type. However, Design cannot prevent you from mistakenly integrating two codelists that have compatible attributes but whose data are completely different (for example for SEX and YESNO in the sample study). If you choose to associate these two codelists, imported items will be decoded incorrectly.

Importing codelists using the Oracle Import Legacy utility

This section describes importing codelists using the Oracle Import Legacy utility.

Steps in importing a Legacy codelist

To import a codelist, with the Codelist Browser open, from the **Codelist** menu, select **Import,** then **Load Legacy Files,** and finally **Load Codelists**. This is the sequence of events:

1. Load the file.
2. Create a default worksheet.
3. Edit the worksheet (if needed).
4. Integrate the codelists.

Each of the four steps above must be executed manually.

For more information on how the default worksheet matches codelists, see "How a default worksheet matches codelists" on page 447.

For more information on editing the worksheet, See "How to edit the worksheet" on page 448.

For more information on integrating the codelists, see "How to integrate the codelists" on page 448

It is recommended that you import the codelists that a protocol uses before you import the protocol. If you import the protocol first, and the codelists that it uses are not in this database instance, you must reconcile the codelists after creating or importing them.

Note: If you are importing any view codelists or subset codelists, ensure that the necessary base codelists (for subset codelists) or base panels and tables (for view codelists) already exist at this site. It is also possible to import base codelists in the same session.

What happens at file loading?

When codelists are exported, the Clintrial software creates an export dump file to store codelist information from the exported codelists, as well as an export log file. When you import codelists, the Clintrial software client computer at the importing database:

- Reads the exported dump file.

  Design provides the placeholder name IMPORT_C.DMP. The Clintrial software client computer must have access to the export dump file. Enter the name of the export dump file for the codelists you want to import, or use the browser to select the file.

- Writes to a log file the Oracle messages resulting from the import.

  Design provides the placeholder name IMPORT_C.LOG. You can accept the placeholder name or supply another name.

  Note: If you give the import log file the same name as the export log file, the import log file will overwrite the export log file.

- Creates a parameter file, IMPORT_C.PAR, to store the Oracle parameters required by the import of the codelists. The parameter file is written to the default directory. You can manually delete the parameter file, which is recreated with each import.

  When you load the imported protocol data, Design runs the Oracle Import utility. The Oracle Import utility opens the IMPORT Status window displaying the same information that is written to the log file. The data import now runs independently of Design.

All codelists are loaded into the CXFR_RECV account. If the receiving account is not empty, you are prompted to confirm deletion.

How to clear the import account

After you import the codelists, the CXFER_RECV account continues to store the imported data. You can clear the worksheet or you can clear the entire account.

Exporting protocols using the Oracle 10 Data Pump utility

This section describes exporting protocols.

Steps in exporting a protocol

To export a protocol, in the Protocol Browser, click on a protocol, and then from the **Protocol** menu, select **Export**. Then, you specify:

1. The directory and filename for the dump file
2. The directory and filename for the log file
3. Whether the clinical data should be included in the dump file

Then, execute the export.

How to specify export dump and log files

At protocol export, Design creates the following files:

- A dump file, to which the metadata and the clinical data is exported.

  The default name of the dump file is:

  – *first-6-characters-of-protocol-name*_M.DMP
- A log file to which the metadata and clinical data export log information is written.

  This file is informational and is not used by the protocol import. The default name of the export log file is:

  – *first-6-characters-of-protocol-name*_M.LOG

You can accept the default name for the .DMP file name or the .LOG file name, or specify different names.

What happens at the export?

When you export a protocol, Design clears the sending account and copies all necessary metadata to the PXFR_SEND account. This account stores all metadata from the exported protocol. It is created at installation.

When the copy is complete, Design runs the Oracle Export utility. The export runs independently of Design. Oracle Export creates a dump file and a log file in Oracle directories on the database server which were created at installation. You must have been granted WRITE access in Oracle to both the Dump Directory and to the Log Directory.

List of protocol export reports

Design provides information on codelists and flags and notes that are referred to by the metadata. When you select the Protocol Export window's **Reports** command, you can view or print the following reports:

- Codelist References — Codelists that items use as a Codename.
- Flag/Note References — Flags and notes referred to either in the TAGS or TAGS_AUDIT tables, or flags and notes that rules use.

The export reports use the copied metadata from the latest export. You can run a report any time after a metadata export.

Exporting protocols using the Oracle Export Legacy utility

This section describes exporting protocols using the Oracle Export Legacy utility.

Steps in exporting a protocol using Clintrial Legacy Export

To export a protocol using Clintrial Legacy export, in the Protocol Browser, click on a protocol, and then from the **Protocol** menu, select **Legacy Export**. Then, you specify:

1. That the export contain either metadata or data (that is, clinical data and flags and notes).

   If you plan to export data, you need to export twice—once specifying metadata and once specifying data.

2. The temporary tablespace to store the exported tables.
3. The dump file name.
4. The log file name.

Then, execute the export.

How to specify temporary tablespace

You specify temporary tablespace to store all the copied clinical data tables for the exported protocol. The tablespace must have enough space to store the entire protocol's clinical data tables and tag data.

How to specify export dump and log files

At protocol export, Design creates the following files:

- Two dump files, one to which the metadata is exported and one to which the clinical data is exported.

  These files must be accessible to the client computer that performs the protocol import. The default names of the dump files are:

  – *first-6-characters-of-protocol-name*_M.DMP — For metadata
  – f*irst-6-characters-of-protocol-name*_D.DMP — For clinical data
- A log file to which the metadata or clinical data export log information is written.

  These files are informational and are not used by the protocol import. The default names of the export log files are*:*

  – *first-6-characters-of-protocol-name*_M.LOG — For metadata information
  – *first-6-characters-of-protocol-name*_D.LOG — For clinical data information
- Two parameter files that store Oracle parameters.

  These files are informational and are not used by the protocol import.

  The names of the parameter files are:

  – *first-6-characters-of-protocol-name*_M.PAR — For parameters related to metadata information
  – *first-6-characters-of-protocol-name*_D.PAR — For parameters related to clinical data information

You can accept the default names for the .DMP file names or the .LOG file names, or specify different names. You cannot change the names of the .PAR files.

How storage parameters are set

The space parameters used in creation of the copied tables default to those in the CTS_DATA storage parameter group. If you export data, you can change the default using Admin or during export.

For more information on setting default space parameters, see Chapter 5.

What happens at the export?

When you export a protocol using Clintrial Legacy export, Design clears the sending account and copies all necessary data or metadata to the following accounts:

- The PXFR_SEND account stores all metadata from the exported protocol. This account is created at installation.
- The *protocol*_DATATRANS account stores all the tag data (flags and notes) and clinical data from the exported protocol. This account is created when you export the data.

When the copy is complete, Design creates the parameter file and runs the Oracle Export utility. The export now runs independently of Design. Oracle Export creates the dump file at the client computer.

The Oracle Export utility displays the EXPORT Status window which displays the same information that is written to the log file.

List of protocol export reports

Design provides information on codelists and flags and notes that are referred to by the metadata. When you select the Protocol Export window's **Reports** command, you can view or print the following reports:

- Codelist References — Codelists that items use as a Codename.
- Flag/Note References — Flags and notes referred to either in the TAGS or TAGS_AUDIT tables, or flags and notes that rules use.

The export reports use the copied metadata from the latest export. You can run a report any time after a metadata export.

How to clear the export account

After you export the protocol, the *protocol*_DATATRANS account continues to store the exported clinical data. Design does not delete this data automatically. When you no longer need to import from the account, you can clear the account to delete the contents.

Importing protocols using the Oracle 10 Data Pump utility

This section describes importing protocols.

Steps in importing a protocol

To import a protocol, with the Protocol Browser open, from the **Protocol** menu, select **Import** then **Load**. Then, do the following:

1. Load the protocol.

   Note: The local protocol is automatically created in the importing database after loading, the metadata and clinical data are loaded (unless deselected during the loading process), and the local protocol is automatically released for general use if no reconciliation issues exist for the codelists, flags and notes.
2. Reconcile codelists, flags, and notes.
3. Resume the import

If codelists, flags or notes needed to be reconciled, the import process must be resumed in order to complete the import, and release the protocol for general use.

Note: If you are importing clinical data, it must be imported at the same time, and from the same dumpfile as the metadata.

Note: It is recommended that you import the codelists that a protocol uses before you import the protocol. If you import the protocol first, and codelists that it uses are not in this database instance, you must reconcile the codelists after creating or importing them.

Note: If you are importing a protocol which uses a Central Coding dictionary:

- If the dictionary *does not* already exist in the importing database, it will be automatically created. The initial status will be CREATED. You should re-save every auto-created dictionary to make them available for

future use. After you review and save the dictionary, the status will be changed to OK.

- If the Central Coding dictionary *does* exist, the coding targets should be reviewed to ensure that they are still valid, since the target items may not be consistent across databases.
- If the Central Coding targets are invalid before export, any panel which uses the targets cannot be installed. Before exporting a protocol, make sure that all the Central Coding targets are valid. If the panel was not installed, while importing that protocol, ORA-01400 will be displayed.

What happens at protocol loading?

When the protocol is exported, Design creates an export dump file to store the metadata and clinical data from the exported protocol, and an export log file. When you import a protocol, the Clintrial server application:

1. Loads the protocol metadata

- Reads the dump file to which the metadata was exported.

  Design provides the placeholder name IMPORT_M.DMP. The Clintrial client computer must have access to the exported dump file. Select the name of the directory in which the dump file resides. Enter the name of the exported dump file for the protocol you want to import, or use the browser to select the dump file.
- Writes two log files with the Oracle messages resulting from the import. One will be named <filename>M.log, and will contain Oracle messages concerning the import of the metadata, and the other will be named <filename>D.log, and will contain Oracle messages about the import of the clinical data. So, for example, if you accepted the placeholder name of IMPORT_M.LOG, the two log files would be named IMPORT_MM.LOG and IMPORT_MD.LOG.
- Design copies the protocol metadata to the Oracle account PXFR_RECV, which was created at installation.
- When you load the imported protocol metadata, Design runs the Oracle Import utility. The metadata import now runs independently of Design.

2. Creates the local protocol with the name specified.

   Provide the following information:

   - Local Protocol — Name of new protocol (Required)
   - Data Space — Default tablespace for clinical data tables (Required)
   - Index Space — Default tablespace for indexes (Required)
   - Parent Protocol — Name of local parent protocol

- Codelist Import — Import number to use in associating codelists

For more information on protocol attributes, including protocol naming, tablespaces for clinical data tables and indexes, and parent protocols, see Chapter 9.

3. Checks whether the codelists, flags and notes have a valid association in the database to which you are importing.

For more information on codelists, see Chapter 12.

If there are missing codelists, flags or notes, you must manually reconcile these objects and resume the Import.

4. Installs the metadata.
5. If selected, reads the same dump file to which the protocol was exported and loads the clinical data.
6. Releases the protocol for general use.

The final step in protocol import is to release the protocol for use. You can release a protocol whether clinical data has been loaded. However, after you release the protocol, you can no longer import clinical data into it. The release process:

1. Copies the metadata from the PXFR_RECV account into the CTSDD dictionary account.
2. Reassigns all internal identifiers as necessary.
3. Removes the metadata from the PXFR_RECV account.
4. If the release is successful, the protocol will no longer appear in the Protocol Import Browser, but will appear in the Protocol Browser with a status of NORMAL.

How to reconcile the metadata

If the local protocol has a status of PT_ERROR, you must associate the codelists, flags, or notes referred to in the import .DMP file from which the local protocol was created with the appropriate corresponding objects in the database instance to which the protocol was imported.

The objects to reconcile are:

- Codelists

  For each codelist referenced, select an existing codelist whose code field type and value field type matches that of a codelist in the importing database instance.

- Flags and notes

  For each flag or note referenced at the exporting database instance, do one of the following:

  – Select an existing flag or note in the importing database instance.

  – Create new flags and notes by selecting **<New>** from the drop-down list. The flag or note is created with the same category and name.

  You can filter the items to show only those in error. Each browser includes a summary line at the bottom indicating how many records are still not reconciled.

Note: Flag and note definitions (as defined in CTS.TAGDEFS and CTS.CATDEFS) are not included in protocol export. However, the flag and note data attached to clinical data (the contents of *protocol-account.*TAGS) are included in protocol export. Therefore, during metadata installation, you must reconcile flags and notes by selecting new flag definitions or note definitions for the imported tag data. Select from the existing flags and notes on the importing database, or create new flags and notes.

How to delete a protocol

You can delete a partially imported protocol. If you delete the protocol, Design deletes all metadata and clinical data that have been imported. To delete a protocol, with the Protocol Import Browser open, from the **Import** menu, select **Delete**.

Importing protocols using the Oracle Export Legacy utility

This section describes importing protocols using the Oracle Export Legacy utility.

Steps in importing a Legacy protocol

To import a protocol, with the Protocol Browser open, from the **Protocol** menu, select **Import, Load Legacy Files**, then **Load Metadata**. Then, do the following:

1. Load the metadata.
2. Create a protocol in the importing database (the local protocol).
3. Reconcile codelists, flags, and notes.

   For information on how to reconcile the metadata, see "How to reconcile the metadata" on page 457.
4. Install the metadata.
5. Load the clinical data (optional).
6. Release the protocol for general use.

Note: It is recommended that you import the metadata and the data at approximately the same time, in order to minimize the possibility of changes to the metadata that would compromise the loading of the data.

Note: It is recommended that you import the codelists that a protocol uses before you import the protocol. If you import the protocol first, and codelists that it uses are not in this database instance, you must reconcile the codelists after creating or importing them.

What happens at metadata loading?

When the protocol is exported, Design creates an export dump file to store the metadata and data from the exported protocol, and an export log file. When you import a protocol, the Clintrial software client computer at the importing database:

- Reads the dump file to which the metadata was exported. (The clinical data is loaded in a separate step.)

  Design provides the placeholder name IMPORT_M.DMP. The Clintrial client computer must have access to the exported dump file. Enter the name of the exported dump file for the protocol you want to import, or use the browser to select the dump file.
- Writes to a log file the Oracle messages resulting from the import.

  Design provides the placeholder name IMPORT_M.LOG. You can accept the placeholder name or enter another name.

  Note: If you give the import log file the same name as the export log file, the import log file will overwrite the export log file.

- Creates a parameter file, IMPORT_M.PAR, to store the Oracle parameters required by the import of the metadata.

  The parameter file is written to the default directory (CT_TEMP_PATH). You can manually delete the parameter file, which is recreated with each import.

When you load the imported protocol metadata, Design runs the Oracle Import utility. The Oracle Import utility opens the IMPORT Status window which displays the same information that is written to the log file. The metadata import now runs independently of Design.

Because the Protocol Import Browser does not know when the import is complete, you must manually refresh the window containing the list of protocols.

Design copies the protocol metadata to the Oracle account PXFR_RECV, which was created at installation.

How to install the metadata

If the local protocol has a status of PT_CHKD, you can install the panels for the protocol through the Import Protocol window. You can choose to create only data dictionary metadata, or you can create both data dictionary metadata and clinical data tables. If you choose not to create clinical data tables, you can create them later, when you load the clinical data.

When you select **Install Metadata** from the **Import** menu, a message is displayed that asks you, "Do you want to defer creation of tables until loading data?" If you click **Yes**, the clinical data tables are created when you load the clinical data. If you click **No**, the clinical data tables are created as part of installing the metadata. It is recommended that you do not defer creation of the clinical data tables. If you do so, then, when you load the data, the Clintrial software creates the clinical data tables in the same tablespace that the tables were in during export. (If this tablespace does not exist, then the Clintrial software uses the protocol's default tablespace.) The end result can be that the protocol's tablespace may be filled with unnecessary tables.

Note: You cannot create tables in a data dictionary protocol during import, nor load data.

Caution: You can import from earlier versions of the Clintrial software. However, if you import from versions prior to 4.2, you must elect to defer creation of the clinical data tables until you do the data loading step.

If the panels install correctly, the local protocol has one of the following statuses:

- PT_DD_INSTALLED — Only data dictionary metadata are created.
- PT_INSTALLED — Data dictionary metadata and empty clinical data tables are created.

What happens at data loading?

If the new local protocol is a clinical data protocol, thesaurus protocol, or Lab Loader protocol, you can now load the clinical data from the exported protocol's clinical data .DMP file. In the Protocol Import Browser, select a protocol with the status PT_DD_INSTALLED or PT_INSTALLED and load the data.

When the protocol was exported, the Clintrial software created an export dump file to store the clinical data from the exported protocol, as well as an export log file. When you load clinical data at protocol import, the Clintrial software client computer at the importing database:

- Reads the dump file to which the clinical data was exported. (The metadata was loaded in a separate step.)

  Clintrial Design provides the default name *first-6-characters-of-protocol-name*_D.DMP, on the assumption that this is the name of the export dump file from which to load the data. The Clintrial software client computer must have access to the export dump file. Accept the default, type in the name of the export dump file for the protocol you want to import, or use the browser to select the file.
- Writes to a log file the Oracle messages resulting from the import.

  Design provides the placeholder name *first-6-characters-of-protocol-name*_D.LOG. You can accept the placeholder name or supply another name. Be aware that if you give the import log file the same name as an existing file (including the export log file or the metadata import log file), the data import log file will overwrite the file of the same name.
- Creates a parameter file, *first-6-characters-of-protocol-name*_D.PAR, to store the Oracle parameters required by the import of the data.

  The parameter file is written to the default directory. You can manually delete the parameter file, which is recreated with each import.

When you load the imported protocol data, Design runs the Oracle Import utility. The Oracle Import utility opens the IMPORT Status window displaying the same information that is written to the log file. The data import now runs independently of Design.

All imported clinical data is loaded into the protocol account.

The protocol status is not updated after the loading of the imported data.

21 Design and Multisite

Overview

This chapter describes how Design works differently in a Multisite environment, and contains the following sections:

- Working with distributed protocols
- Working with distributed codelists
- Working with flags and notes in a replication environment

Working with distributed protocols

This section describes the following tasks related to working with distributed protocols:

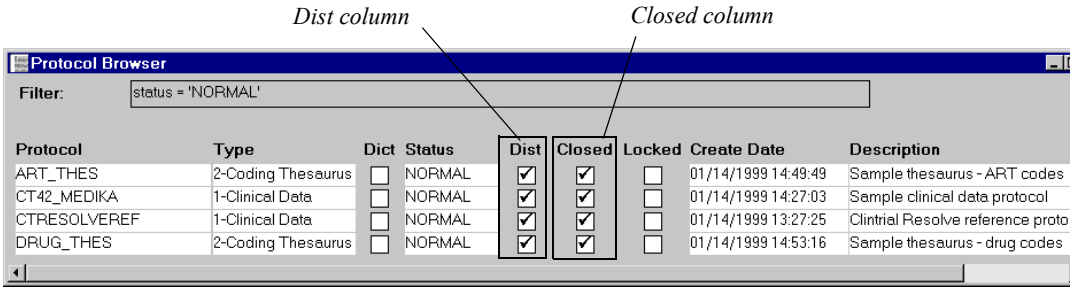
- The Protocol Browser
- Distributing searchlists
- Accepting view protocols
- Modifying protocols
- Deleting protocols
- Installing database tables
- Compiling thesaurus views

The Protocol Browser

The Protocol Browser contains two columns related to distribution:

- The Dist column, which, when checked, indicates that the protocol is in distribution.
- The Closed column, which, when checked, indicates that the protocol is at a Distribution Subordinate site, or that the protocol is at the Distribution Master site and closed for revision.

The following figure shows the Protocol Browser:



Distributing searchlists and parent protocols

When you distribute a protocol, the protocol's searchlist is also distributed. The value of the Parent Protocol attribute is also distributed.

When you accept the protocol at the Destination site, Multisite checks whether the protocols in the searchlist, as well as the Parent Protocol, are present. If protocols in the searchlist are not present at the Destination site, these protocols are deleted from the distributed protocol's searchlist.

If the Parent Protocol is not present at the Destination site, this attribute is cleared for the protocol at that site.

Accepting view protocols

If the base protocol does not exist at a Destination site for a view protocol, an error occurs when you try to accept the view protocol.

Modifying protocols

You can only modify a protocol in distribution, and modify or delete the objects contained in the protocol, at the Distribution Master site. Before you can modify the protocol at the Distribution Master site, you must open the protocol for revision. For information about opening protocols for revision, see *Multisite*.

Caution: If you or another user is working with a distributed protocol in Design at the Distribution Master site, when you first distribute the protocol or when you close the protocol for revision, the protocol and the objects it contains can still be modified. The protocol is not unmodifiable until you close all windows and set the protocol.

Deleting protocols

To delete a protocol from a Distribution Subordinate site, you must first detach the site from its Source site. Detaching a site removes the site from distribution for the protocol and enables you to modify or delete the protocol at that site.

You cannot delete a protocol from the Distribution Master as long as the Distribution Master site has Destination sites for the protocol. To delete a protocol from the Distribution Master site, you must first detach all the Destination sites for the protocol.

For information about detaching Destination sites, see *Multisite*.

Installing database tables

After you accept a protocol for the first time at each Destination site, you must install database tables for each panel contained in the protocol, even if the database tables were installed at the Source site.

Distributing revisions of protocol

When you distribute a revision of a protocol, you do not have to install panels that existed in the previous revision. These panels remain installed at Destination sites when you accept a revision of a protocol.

Caution: If you add a new panel to the protocol at the Distribution Master site, then distribute the revision, you must install that panel at each Destination site.

Deinstalling panel tables

After you distribute a protocol, you can only deinstall tables for panels if the protocol is not in replication.

Deinstalling panel metadata

After you distribute a protocol, you cannot deinstall metadata for panels contained in the protocol, even if you open the protocol for revision at the Distribution Master site.

Caution: Because you cannot deinstall panel metadata, you cannot delete items in a protocol that is in distribution.

If you must deinstall panels in a distributed protocol:

1. Detach all Destination sites. For information about detaching Destination sites, see *Multisite*.
2. Deinstall the panels at the site that was the Distribution Master site and make the necessary panel revisions.
3. Delete the protocol from all the sites that were Distribution Subordinate sites.
4. Distribute the protocol as if you are doing so for the first time.

Compiling validation procedures

When you install database tables at a Destination site after accepting a protocol for the first time, validation procedures are compiled automatically.

Caution: Functions called by rules and derivations are not distributed with protocols. Because validation procedures depend on functions being present at the current site, validation procedures that are valid at a Source site may be invalid at the Destination site. You may have to send functions from the Source site to the Destination site for validation procedures to be valid. For information about working with functions, see *Multisite*.

Compiling thesaurus views

For a distributed coding thesaurus protocol, you typically need to take an additional step to create the tables on which the thesaurus views are based at each Distribution Subordinate site.

For thesaurus views based on Clintrial software panels, you must create tables for those panels after you accept the initial distribution of the coding thesaurus protocol. Thesaurus views become valid when you create the tables.

You compile thesaurus views after you accept the initial distribution of the thesaurus protocol, and after you accept each revision of the thesaurus protocol.

Caution: Base tables used by thesaurus views are not distributed with protocols. Because thesaurus views may depend on base tables being present at the current site, thesaurus views that are valid at a Source site may be invalid at the Destination site. You must create tables at the Destination site if they do not exist, then compile the thesaurus views that use the tables. You can also copy base tables from the Source site to the Destination site to ensure that the same data is used for coding at both sites. For information about working with base tables, see *Multisite*.

Working with distributed codelists

This section describes the following tasks related to working with distributed codelists:

- The Codelist Browser
- Modifying codelists
- Compiling view codelists and subset codelists
- Deleting codelists
- Deleting codelist values

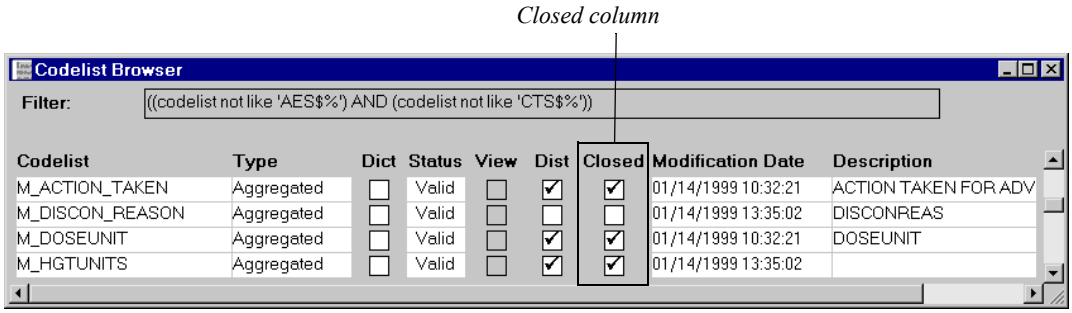
The Codelist Browser

The Codelist Browser contains a Closed column, which, when checked, indicates that the codelist is in distribution. When a codelist is marked Closed, you cannot edit or delete the codelist.

The Closed column is checked for:

- A distributed codelist at a Distribution Subordinate site
- A distributed codelist at the Distribution Master site, except when the codelist is open for revision

The following figure shows the Codelist Browser:



Compiling view codelists and subset codelists

You must compile view codelists and subset codelists at each Distribution Subordinate site. You compile view codelists and subset codelists after you accept the initial distribution of the codelist, and after you accept each revision of the codelist.

If a view codelist is based on a panel, you must import or distribute to the site the protocol containing the panel before compiling the view codelist.

Caution: Base tables used by view codelists are not distributed with codelists. Because view codelists may depend on base tables being present at the current site, view codelists that are valid at a Source site may be invalid at the Destination site. You must create tables at the Destination site if they do not exist, then compile the view codelists that use the tables. You can also copy base tables from the Source site to the Destination site to ensure that the same data is used for view codelists at both sites. For information about working with base tables, see *Multisite*.

Modifying codelists

You can only modify a codelist in distribution at the Distribution Master site. Before you can modify the codelist at the Distribution Master site, you must open the codelist for revision. For information about opening codelists for revision, see *Multisite*.

Caution: If you or another user is working with a distributed codelist in Design at the Distribution Master site, when you first distribute the codelist or when you close the codelist for revision, the codelist can still be modified. However, once you refresh the Codelist Browser, you can no longer modify the codelist.

Deleting codelists

To delete a codelist from a Distribution Subordinate site, you must first detach the site from its Source site. Detaching a site removes the site from distribution for the codelist and enables you to modify or delete the codelist at that site.

You cannot delete a codelist from the Distribution Master site as long as the Distribution Master site has Destination sites for the codelist. To delete a codelist from the Distribution Master site, you must first detach all the Destination sites for the codelist.

For information about detaching Destination sites, see *Multisite*.

Deleting codelist values

For a codelist in distribution, you can only delete codelist values at the Distribution Master site. Before you can delete codelist values at the Distribution Master site, you must open the codelist for revision. For information about opening codelists for revision, see *Multisite*.

Note: You cannot delete codelist values for a codelist that is referenced by an item.

When you accept a revision of a codelist for which you deleted codelist values at a Destination site, Design checks whether the codelist is referenced by any items in the database. If the codelist is not referenced by any items, the codelist values are deleted at the Destination site.

If the codelist is used by one or more items at the Destination site:

- The status of the codelist values that were deleted at the Distribution Master site is Invalid.
- The codelist values remain in the database because they may be used by existing clinical data.
- You can no longer use the codelist values when entering or editing clinical data.

Working with flags and notes in a replication environment

This section describes the following tasks related to working with flag definitions and note classes in Design, when the current site is part of a replication environment for the CTS account:

- Ownership of Clintrial software-supplied flags and notes
- Creating flags and notes
- Modifying flags and notes
- Deleting flags and notes

Ownership of Clintrial software-supplied flags and notes

When you set up a replication environment for the CTS account, the Replication Master site assumes ownership of the Clintrial software-supplied flags and notes.

Creating flags and notes

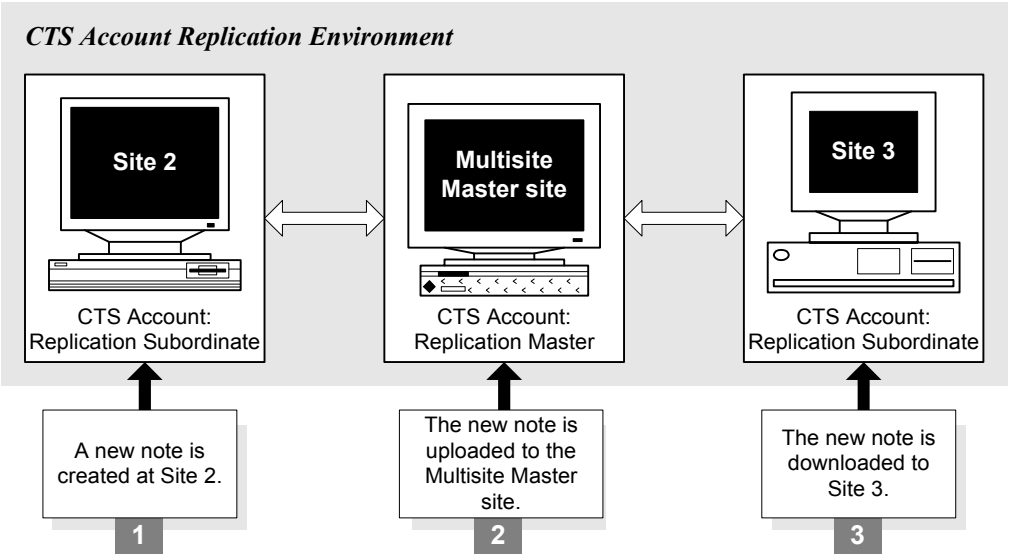
If you create a flag or note at the Replication Master site for the CTS account, the object downloads to all the Replication Subordinate sites in the replication environment for the CTS account.

If you create a flag or note at a Replication Subordinate site for the CTS account, the object is uploaded to the Replication Master site. After uploading to the Replication Master site, the new flag or note is downloaded to other Replication Subordinate sites in the replication environment for the CTS account.

Note: The Multisite Master site must be the Replication Master site for the CTS account.

Caution: If a Replication Subordinate site is invited into the replication environment for the CTS account as “Read Only”, the site cannot create flags or notes.

The following figure shows the uploading and downloading of a new note in a replication environment for the CTS account:



Modifying flags and notes

You can only modify a flag or note in a replication environment for the CTS account at the site that owns the object. You cannot modify a flag or note at any other site in the replication environment unless you first transfer ownership of the object.

When you modify a flag or note, the modifications are uploaded and downloaded in the replication environment just as new flags and notes are replicated.

Deleting flags and notes

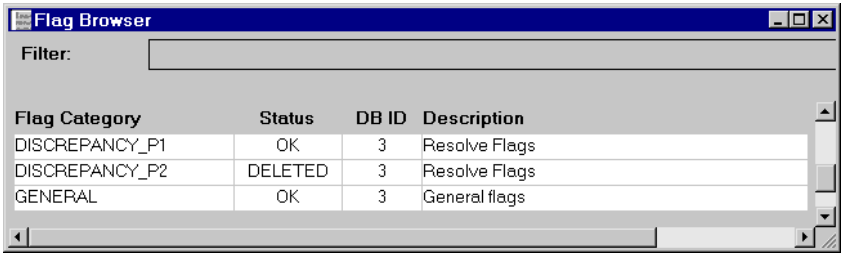
You can only delete a flag or note when the CTS account is in a replication environment at the site that owns the object. You cannot delete a flag or note at any other site in the replication environment unless you first transfer ownership of the object.

When you delete a flag or note:

- The status of the flag or note is Deleted at all sites in the replication environment for the CTS account.

- The flag or note remains in the CTS account because it may be used by existing clinical data.
- You can no longer use the flag or note when entering or editing clinical data.

The following figure shows the Flag Browser, with a deleted flag:



22 Page Sections

Overview

Part III describes the Clintrial software objects that define the data-entry windows which the data-entry operator uses in a study book in Enter. Design provides objects for entering and editing clinical data that fall into the following categories:

- Objects that define the layout of items and sections on a study page:

  Page sections — For more information, see this chapter.

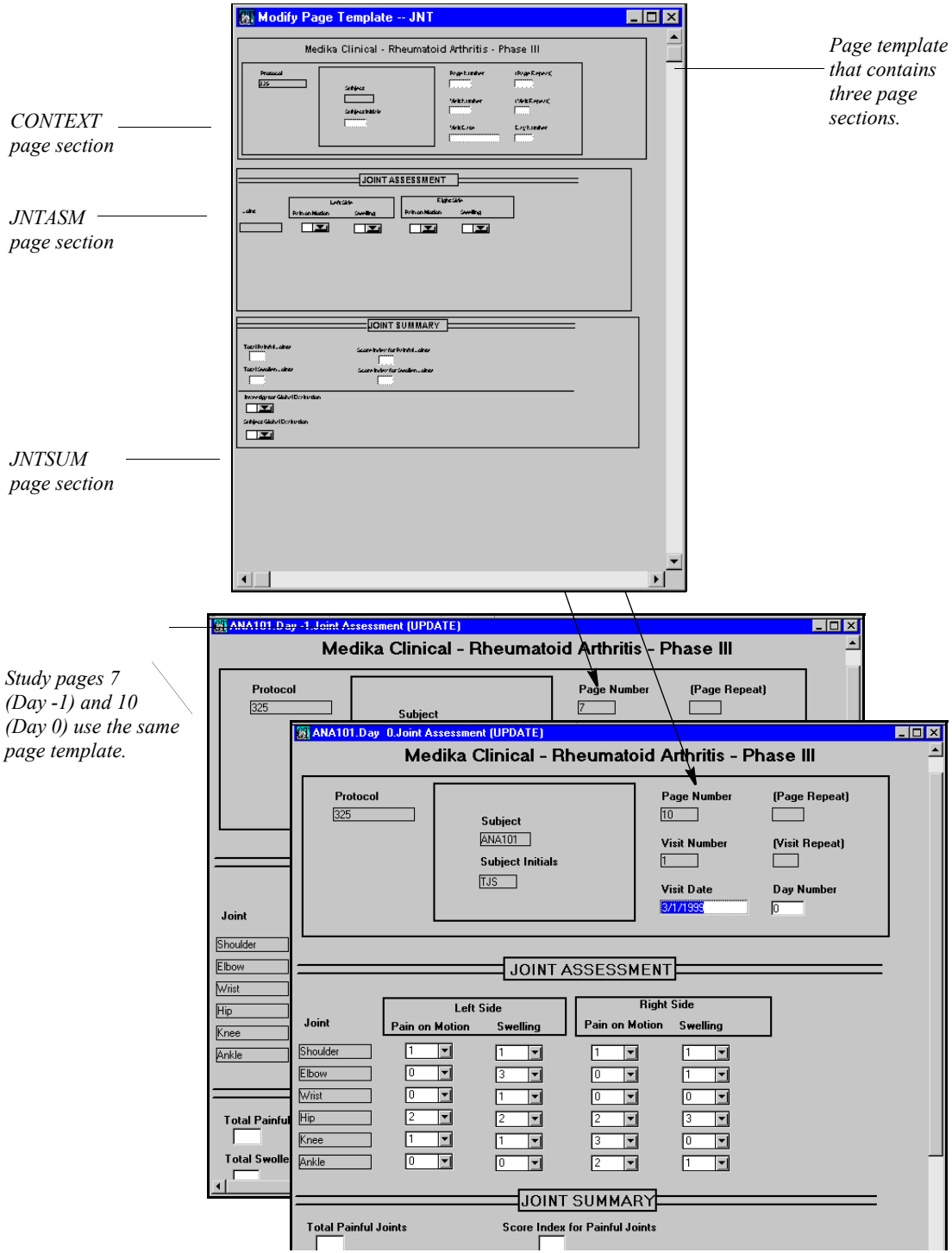
  Page section layouts — For more information, see Chapter 17.

  Page templates — For more information, see Chapter Chapter 16.

- Objects that build a study book and order specific pages within a study book:

  Study books (including blocks) and study pages — For more information, see Chapter 18.

The following figure shows the relationship among page sections, page templates, and study pages in a study book:


Modify Page Template -- JNT
Medika Clinical - Rheumatoid Arthritis - Phase III
JOINT ASSESSMENT
JOINT SUMMARY
Page template that contains three page sections.
CONTEXT page section
JNTASM page section
JNTSUM page section
Study pages 7 (Day -1) and 10 (Day 0) use the same page template.
ANA101.Day -1.Joint Assessment (UPDATE)
ANA101.Day 0.Joint Assessment (UPDATE)
Medika Clinical - Rheumatoid Arthritis - Phase III
Protocol
325
Subject
ANA101
Subject Initials
TJS
Page Number
(Page Repeat)
Visit Number
(Visit Repeat)
Visit Date
Day Number
3/1/1995
JOINT ASSESSMENT
Left Side
Right Side
Pain on Motion
Swelling
Shoulder
Elbow
Wrist
Hip
Knee
Ankle
JOINT SUMMARY
Total Painful Joints
Score Index for Painful Joints

Creating page sections

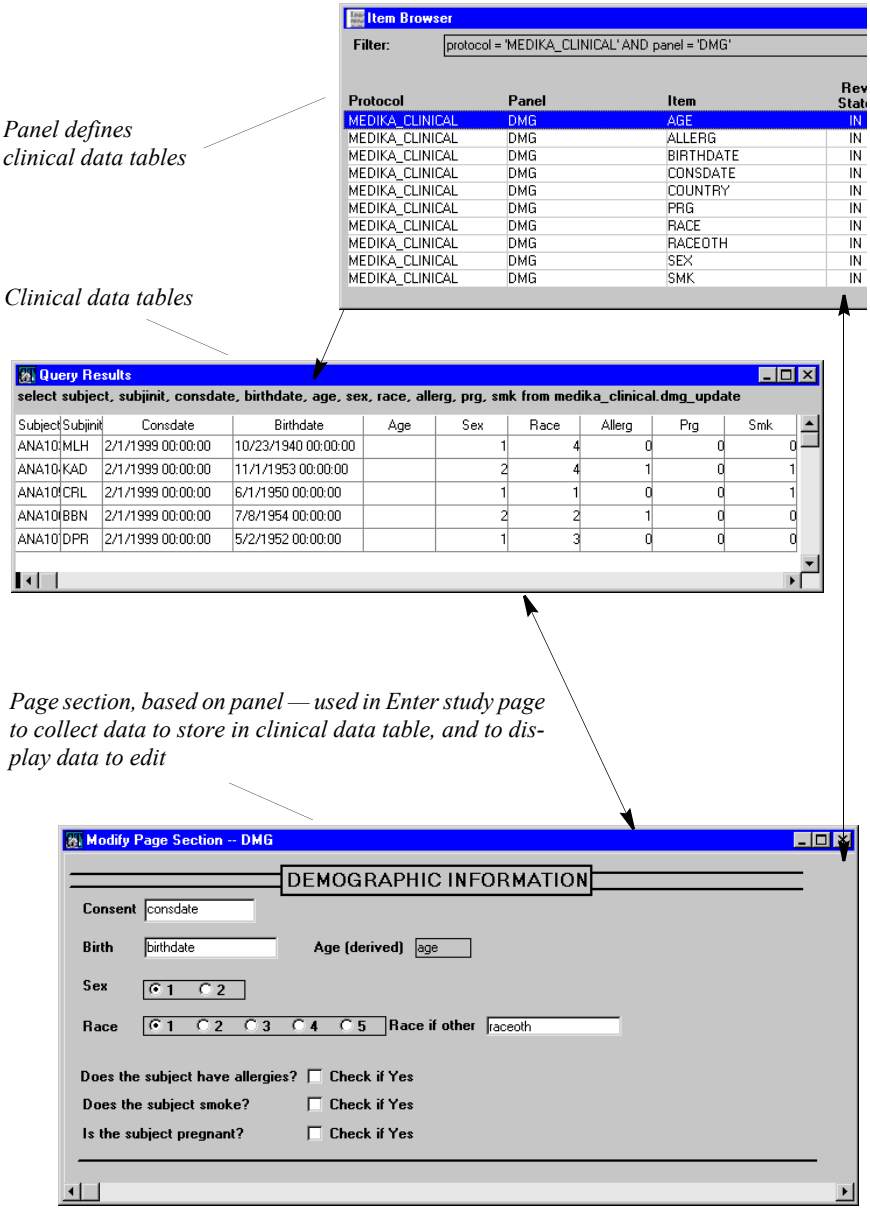
What is a page section?

A *page section* is a section of a study page that collects a particular type of clinical data. The data is then stored in clinical data tables defined by a panel that corresponds to the page section.

For example, a study page might include a page section for demographic data (DMG panel), a page section for vital signs (VITAL panel), and a page section for physical examination results (PHYEXM panel).

How to collect data through a page section

The following figure shows the relationship of a panel and its items to a page section in which data is entered, and to the tables that store the clinical data:



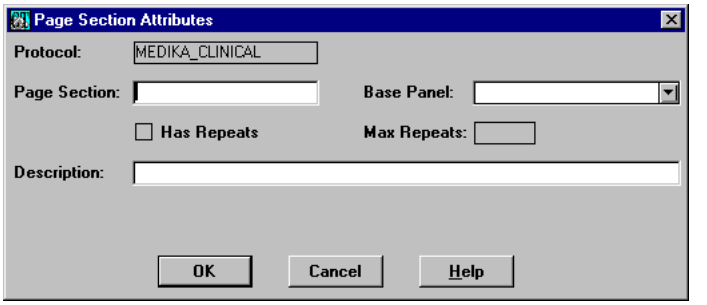
The page section called Demographic Information is based on the DMG panel. During data entry, the data is entered in the page section and stored in clinical data tables also defined by the panel DMG.

How to create a page section

To create a page section, you do the following:

1. With the Page Section Browser open, from the **Page Section** menu, select **Create**.

   The Page Section Attributes dialog box opens:



2. Specify the attributes and click **OK**.

   The Modify Page Section window opens.

3. Edit (optionally) and **Save** the page section layout.

For information on page section attributes, see "Page section attributes" on page 487.

For information on editing the page section layout, see Chapter Chapter 17.

How to modify page section attributes

To modify page section attributes:

1. In the Page Section Browser, select the page section for which you want to modify the attributes.
2. From the **Page Section** menu, select **Modify**.
3. From the **Controls** menu, select **Page Section Attributes**.

   The Page Section Attributes dialog box opens.

4. Modify the attributes as needed and then click **OK** to save them, or click **Cancel** to cancel the modifications.

Attaching procedures to page section fields

What is a value changed procedure?

A *value changed procedure* is a type of data-entry processing procedure that runs automatically when the data-entry person changes the value of a field and then exits the field. You attach a value changed procedure to a field in a page section.

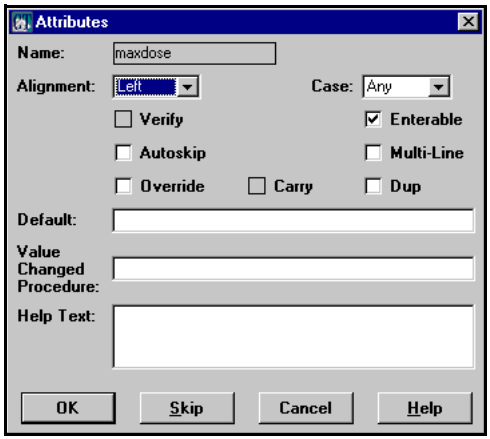
For more information on data-entry processing procedures, see "Design menu commands" on page 388, "What is a data-entry processing procedure?" on page 367, and the *Reference Guide*.

How to specify a value changed procedure

To attach a value changed procedure to a field in a page section:

1. From the **Page Section** menu, select **Modify**.
2. Select the field to which you want to attach the procedure, and, from the **Design** menu, select **Attributes**.

   The Modify Object Attributes dialog box opens:



3. In the Value Changed Procedure field, enter the name of the procedure. Click **OK** to save the procedure.
4. Test the procedure in page template Test Mode, or in Enter.

Page sections and panels

How do page sections relate to panels?

Each page section must be based on a single panel. However, multiple page sections can be based on the same panel. Base multiple page sections on the same panel when you want the data stored in a single table (defined by the panel), but you want the display to vary.

For example, in the MEDIKA study book, the DRGCMP panel is the base panel for the following four page sections:

- BOTDISP — For information about drug bottles dispensed at Day 0
- BOTRET — For information about drug bottles returned and dispensed at Day 30, Day 60, and any unscheduled visits
- BOTRET_FINAL — For information about drug bottles returned at the final visit (Day 90)
- DRGCMP — For information about drug use compliance

Each of the four page sections contains different text, and uses a different set of items from the panel. You modify the page sections by editing the page section layout.

Caution: Be sure that you select a base panel of the correct type. If you use a page section in a study book in a way that is not compatible with the panel type, the result can be unusable data. For example, if you create a page section based on a Type 1 panel, and then use it on more than one study page (in which case it should have been based on a Type 3 panel), data entered in the two page sections for the same patient fails to merge. The unique constraint on the SUBJECT_ID system item for a Type 1 panel causes the merge to fail when it encounters another record with the same value for the SUBJECT_ID.

For information on page section layouts, see Chapter 17.

For information on deleting items from a page section layout, see "How to delete items and text" on page 400.

For information on adding an item into a page section layout, see "Controls menu commands" on page 401.

The CONTEXT page section and CONTEXT panel

All subject-related items in the CONTEXT page section that are displayed on this page section are nonenterable. Enter populates the subject-related items (whether displayed or not) from the enrollment panel each time a new study page is created. In addition, global change keeps the data consistent across all pages if the enrollment panel's data is ever edited.

Repeating items

What is a repeating item?

A *repeating item* is an item for which multiple values can be entered within a page section.

Which panels can have repeating items?

You can specify repeating items only in page sections based on panels containing non-patient data or on panels that allow for more than one record for a subject or subject visit. Therefore, only panels of the following types can serve as base panels for pages sections with repeating items:

- Type 0 — Non-Patient Data
- Type 2 — >1 Record per Patient
- Type 4 — >1 Record per Patient Visit

Note: There is no requirement that a page section based on one of the preceding panel types must contain repeating items. However, if you create such a page section and do not specify any repeating items, the page section layout may sometimes display multiple rows of the non-repeating record. (For example, this may occur because a within-panel master-detail relationship exists to a page section with repeating items.) You can hide the superfluous rows by sizing the page section layout to display only the first row.

For more information about page section layouts, see Chapter 17.

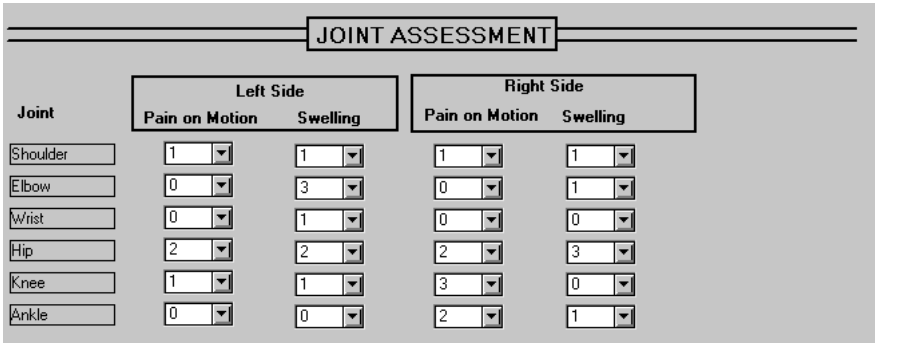
For more information on within-panel master-detail relationships, see Chapter 16.

How to specify a repeating item

When you create or modify a page section, you can specify that the page section contains repeating items. You also can specify the number of repeats allowed. A page section can contain repeating items or non-repeating items, but not both.

To specify repeating items, in the Page Section Attributes dialog box, select the Has Repeats attribute when you create the page section.

The following figure shows repeating items in the JNTASM page section, as displayed in Enter:



In this example, Joint is a repeating item, as are the four items containing values assessing the severity of pain or swelling.

For information on how to create a page section, see "How to create a page section" on page 480.

Grouping page sections

How to use page sections multiple times

You can use a single page section multiple times. For example, in the MEDIKA study book the page section for Vital Signs (VITAL) appears five times, once for each visit.

How to group page sections in page templates

Page sections are grouped into page templates. One or more page templates can use a single page section. For example, in the MEDIKA study book the SIGNATURE page section appears in three different page templates — LABLNG, BOTRET, and TERMINATION.

Moreover, by establishing master-detail relationships between panels or between page sections based on the same panel, you can place multiple page sections based on the same panel on the same page template.

For more information on how master-detail relationships affect page section grouping, see "Panels with master-detail keys" on page 212, "Creating page templates with master and detail page sections" on page 370, and the following section.

For more information on page templates, see Chapter 16.

Master page sections, detail page sections, and subset page sections

What is a master page section?

A *master page section* is a page section based on the panel containing the master key item. A master page section may allow repeating items. The master page section is placed on a page template with the associated detail page section. In a study page, when a record is selected in the master page section, the associated records are displayed in the detail page section.

What is a detail page section?

A *detail page section* is a page section based on the panel containing the detail key item. A detail page section may allow repeating items. The detail page section is placed on a page template with the associated master page section. In a study page, when a record is selected in the master page section, the associated detail records are displayed in the detail page section.

What is a subset page section?

A *subset page section* is a page section that contains a subset of records from the base panel. To create a subset page section, you specify a subset key to determine which set of records to display in the subset page section. You can use subset page sections to place multiple page sections based on the same panel on a single page template, and to create a within-panel master-detail relationship among these page sections.

Summary of master-detail relationship

To create a master-detail relationship among records, do the following:

1. Create a panel or panels to contain the records in a master-detail relationship. For panels in a cross-panel master-detail relationship, you must specify master key items and detail key items. For a panel to contain a within-panel master-detail relationship, create the panel without master or detail key items. Or, create the panel with subset key items.

   For information on setting up panels for a cross-panel master-detail relationship, see "How to define a master-detail relationship" on page 214.

   For information on setting up a panel with a subset key item for a within-panel master-detail relationship, see "How to specify an item as a subset key" on page 217.

2. Create a master page section and a detail page section.

   For a cross-panel master-detail relationship, create a master page section based on a panel with a master key item, and create a detail page section based on a detail key item.

   Note: You can set up the master key item so that Enter automatically generates a unique value for each record. To do this, specify the data type of the master key item as FIXED, and clear the master page section's Enterable attribute. Otherwise, the data-entry person enters the value for the master key.

   For a within-panel master-detail relationship, create two page sections based on the same panel. Master key items and detail key items are not needed for within-panel master-detail relationships.

3. Place the master page section and detail page section on a page template.

   For information on completing the cross-panel master-detail setup on the page template, see "Creating page templates with master and detail page sections" on page 370.

   For information on completing the within-panel master-detail setup on the page template, see "Repeating items in master-detail relationships" on page 381.

Page section attributes

The following table lists the page section attributes:

Attribute:	Description:
Protocol	Name of the protocol.
Page Section	Name of the page section.
Base Panel	Name of the panel that defines the items used by the page section.
Has Repeats	Specifies whether the panel contains repeating items. Available only if the page section is based on a Type 2 or Type 4 panel (> 1 record).
Max Repeats	Limits the number of repeats for a repeating item.
Description	Description of the page section.

When you select the **Page Section** menu's **Show** command, the Clintrial software displays the previously described page section attributes as read-only, and also displays the following additional read-only attributes:

Attribute:	Description:
Has Sequences	Sequences are specified in the page section layout.
Status	Status of the page section layout. Values are: • Valid — Page section can be used on a page template. • Old Layout — Page section has a layout created in a previous version of the Clintrial software and cannot be used. An error message is displayed in Enter. • Invalid — Page section is unusable but no error message is displayed. • No Layout — Page section does not have a layout. An error message is displayed in Enter.

Attribute:	Description:
Modifiable Status	Modifiable if the page section can be modified. Not Modifiable if the page section cannot be modified. Unmodifiable Copy if the page section was created by a connected copy from a page section.
Width	Width of the page section.
Height	Height of the page section.
Modification Date	Date and time of the last modification to the panel.
Modified by	User account that last modified the page section.
Object ID	Unique internal identifier for page section layout.

23 Metadata Reports

Connected Object Report 530

Overview

This chapter describes Clintrial software metadata reports.

What is a metadata report?

A *metadata report* displays in report format the metadata associated with a particular type of database object. For example, the Item and Coding Target Report lists item names, item attributes, and coding targets for all selected panels. The set of Clintrial software metadata reports includes all the primary metadata currently in the Clintrial software data model.

Metadata reports enable you to produce comprehensive descriptions of the Clintrial software metadata associated with a clinical study. This allows for convenient viewing of metadata structures. In addition, producing the complete set of Clintrial software metadata reports simplifies archiving the metadata component of a clinical study either online or on paper.

Types of Clintrial software metadata reports

Reports are available in Design on the following types of metadata:

- Protocols — Reports on protocol attributes, protocol hierarchy, and protocol parameter audits for selected protocols.
- Codelists — Reports on codelist attributes, and codelist and checklist values and usage for selected codelists.
- Panels — Reports on panel attributes and objects for panels in the current protocol.
- Display objects — Reports on page sections, page templates, and study books in the current protocol.
- Connected objects — Reports on connected objects in the current protocol.

Metadata reports are also available in Admin and Manage. For more information on these reports, see Chapter 7, and the *Manage* section of *Manage, Classify, and Lab Loader*.

Some reports can be produced with the following options. See the following examples):

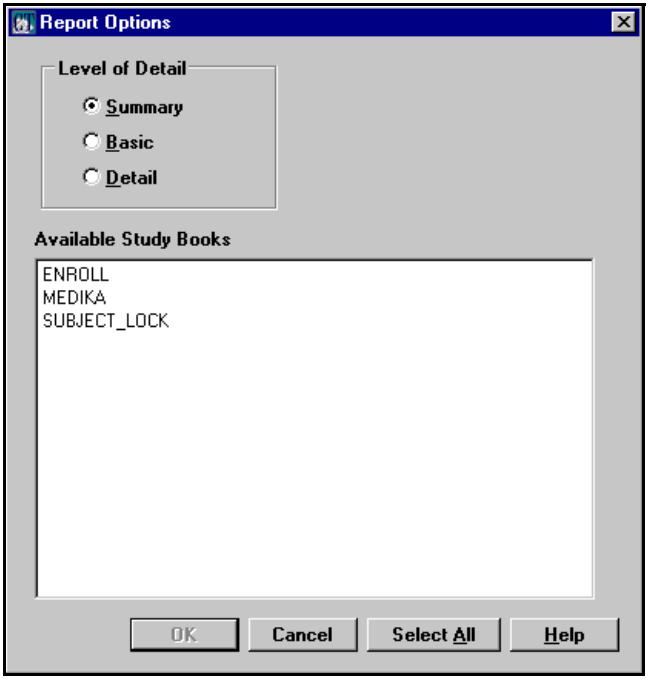
- Level of Detail — The following levels of detail may be available:
  - Summary — Provides a minimal level of detail, typically listing only essential attributes of the type of metadata object reported on.
  - Basic — Provides a listing of basic object attributes.
  - Detail — Provides a complete listing of all object attributes.
- Sort Option — Where there are multiple containers associated with a metadata object, you can choose to sort a report so that the level of detail corresponds to an object type. For example, the Codelist Usage by Codelist Report can be sorted by protocol (least level of detail), panel (more detail), or item (greatest detail).
- Include Unimplemented Panels — For panel reports, you can choose between a report on panels marked for revision but not implemented or for the panel state prior to being marked for revision.

Formatting varies according to the specific report, and generally consists of a modified columnar form. Typically, reports with Detail level of detail are formatted according to logical groupings of attributes.

Note: Most reports with the Level of Detail option provide only the Basic and Detail selections.

Selecting report options

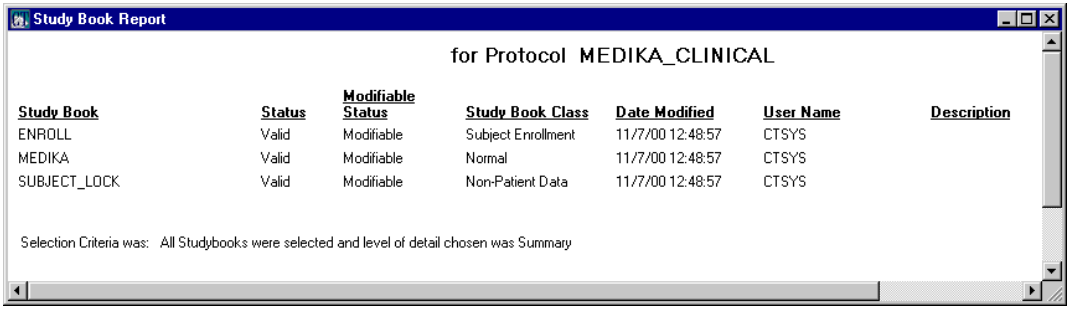
If a particular type of report has options, a dialog box enables you to select report options.For example, if, from the **Reports** menu, you select **Display Objects >> Study Books**, the Report Options dialog box opens:



The following sections show examples of reports with Summary, Basic, and Detail levels of detail.

Example of report with Summary level of detail

The following example shows a Study Book Report with the Summary level of detail:



Study Book Report

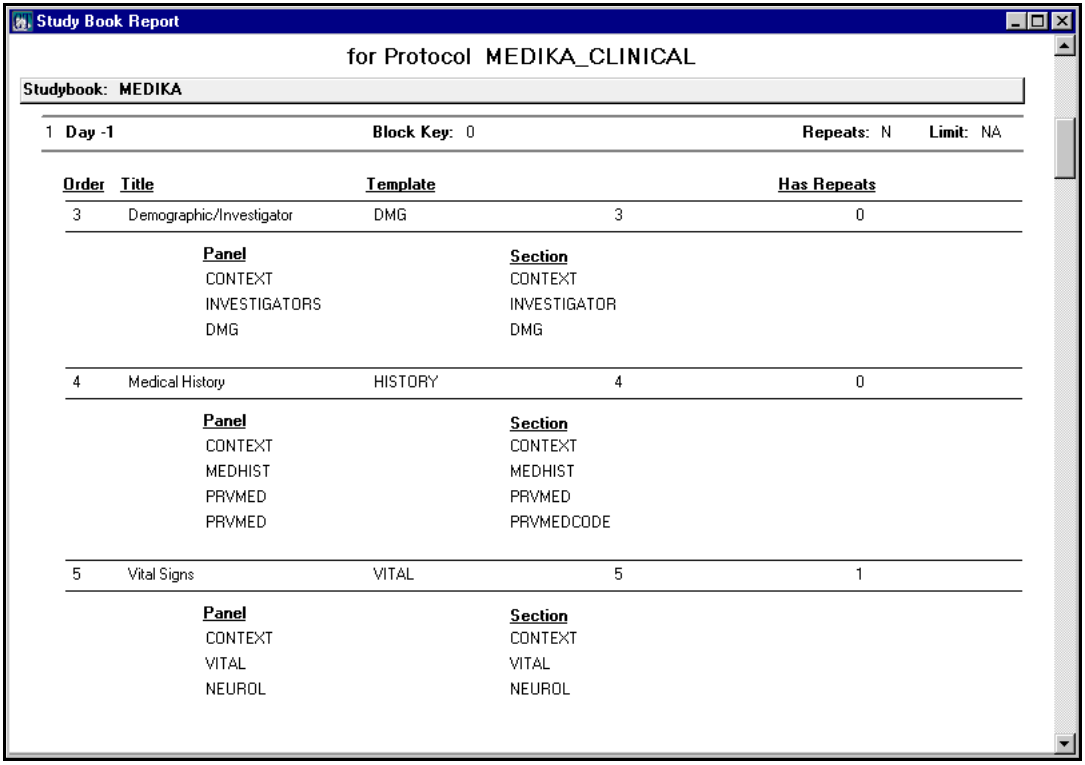
for Protocol MEDIKA_CLINICAL

Study Book	Status	Modifiable Status	Study Book Class	Date Modified	User Name	Description
ENROLL	Valid	Modifiable	Subject Enrollment	11/7/00 12:48:57	CTSYS	
MEDIKA	Valid	Modifiable	Normal	11/7/00 12:48:57	CTSYS	
SUBJECT_LOCK	Valid	Modifiable	Non-Patient Data	11/7/00 12:48:57	CTSYS	

Selection Criteria was: All Studybooks were selected and level of detail chosen was Summary

Example of report with Basic level of detail

The following example shows a Study Book Report with the Basic level of detail:



Study Book Report

for Protocol MEDIKA_CLINICAL

Studybook: MEDIKA

1 **Day -1** **Block Key:** 0 **Repeats:** N **Limit:** NA

Order	Title	Template		Has Repeats
3	Demographic/Investigator	DMG	3	0

Panel	Section
CONTEXT	CONTEXT
INVESTIGATORS	INVESTIGATOR
DMG	DMG

4	Medical History	HISTORY	4	0

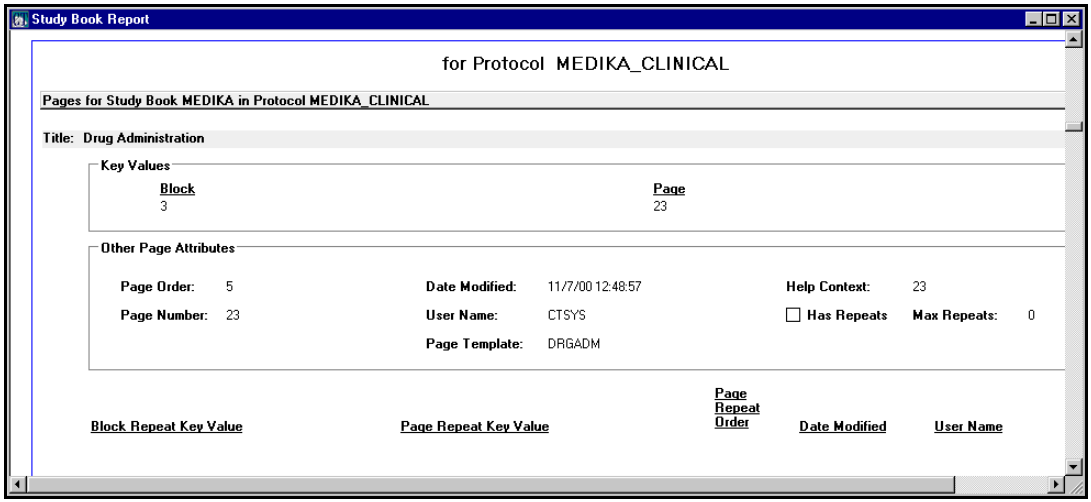
Panel	Section
CONTEXT	CONTEXT
MEDHIST	MEDHIST
PRVMED	PRVMED
PRVMED	PRVMEDCODE

5	Vital Signs	VITAL	5	1

Panel	Section
CONTEXT	CONTEXT
VITAL	VITAL
NEUROL	NEUROL

Example of report with Detail level of detail

The following example shows a Study Book Report with the Detail level of detail:



For more information on the Study Book Report, see "Study Book" on page 528.

Producing and saving metadata reports

In Design, you produce a report by selecting a report type from the **Reports** menu, and then selecting the report from a submenu. Metadata reports can be saved to a .prn file, and then printed out.

Available options for particular types of reports are listed at the beginnings of the sections that describe these reports.

Note: For Clintrial software metadata reports:

- To be included in a report, metadata objects cannot belong to a protocol still listed in the Protocol Import Browser. For its objects to be included in a metadata report, a protocol must be released and displayed in the Protocol Browser.
- If you are using Multisite and want to produce a metadata report on objects in a distributed protocol, the protocol must be accepted, or be a distribution master, at the site to which you are connected.

For more information on how to produce, print, and save a metadata report, see the Design Help.

Protocol reports

The types of protocol reports with the available options are:

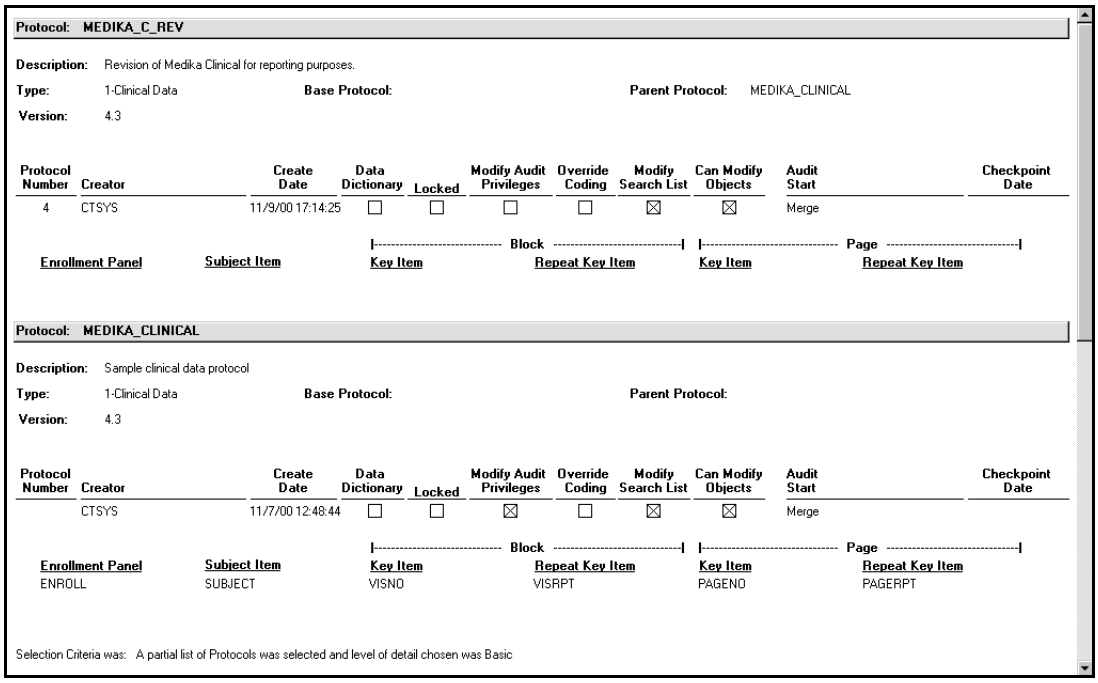
- Protocol Attribute Report — Level of Detail:
  - Basic
  - Detail
- Protocol Hierarchy Report — No options.
- Protocol Parameter Audit Report — Available protocols.

For more information on the Level of Detail options, see "Report options and formats" on page 492.

Protocol Attributes

For selected protocols, the Protocol Attribute Report displays the protocol attributes included in the Show Protocol window, and, with “Level of Detail — Detail” selected, additional attributes including protocol parameters and their values the table on page 498.

The following example shows the Protocol Attributes Report with the Basic level of detail:



Protocol: MEDIKA_C_REV

Description: Revision of Medika Clinical for reporting purposes.

Type: 1-Clinical Data **Base Protocol:** **Parent Protocol:** MEDIKA_CLINICAL

Version: 4.3

Protocol Number	Creator	Create Date	Data Dictionary	Locked	Modify Audit Privileges	Override Coding	Modify Search List	Can Modify Objects	Audit Start	Checkpoint Date
4	CTSYS	11/9/00 17:14:25	☐	☐	☐	☐	☒	☒	Merge	

		Block		Page	
Enrollment Panel	Subject Item	Key Item	Repeat Key Item	Key Item	Repeat Key Item

Protocol: MEDIKA_CLINICAL

Description: Sample clinical data protocol

Type: 1-Clinical Data **Base Protocol:** **Parent Protocol:**

Version: 4.3

Protocol Number	Creator	Create Date	Data Dictionary	Locked	Modify Audit Privileges	Override Coding	Modify Search List	Can Modify Objects	Audit Start	Checkpoint Date
	CTSYS	11/7/00 12:48:44	☐	☐	☒	☐	☒	☒	Merge	

		Block		Page	
Enrollment Panel	Subject Item	Key Item	Repeat Key Item	Key Item	Repeat Key Item
ENROLL	SUBJECT	VISNO	VISRPT	PAGENO	PAGERPT

Selection Criteria was: A partial list of Protocols was selected and level of detail chosen was Basic

For more information on protocol attributes, see Chapter 9, and the Design Help.

The Protocol Attribute Report provides the following information:

Report column:	Description:
Protocol	The name of the protocol.
• Protocol attributes	Level of Detail — Basic • Description • Type • Base Protocol • Parent Protocol • Version • Protocol Number • Creator • Create Date • Data Dictionary (check box) • Locked (check box) • Modify Audit Privileges (check box) • Override Coding (check box) • Modify Search List (check box) • Can Modify Objects (check box) • Audit Start • Checkpoint Date • Enrollment Panel • Subject Item • Key Item (block) • Repeat Key Item (block) • Key Item (page) • Repeat Key Item (page)

Report column:	Description:
Protocol attributes cont'd	Level of Detail — Detail • Type • Creator • Create Date • Protocol Number • Base Protocol • Parent Protocol • Description • Panel and Subject Information (Enrollment Panel, Block Key Item, Page Key Item, Subject Item, Block Repeat Key Item, Page Repeat Key Item) • Audit Information (Audit Start, Merge, Checkpoint Date, Modify Audit Privileges, Audit Sponsor Notes, Audit Investigator Notes) • Export Site • Error Log Item • Help Context • Import Date • Data Space • Help File Name • Index Space • Other Attributes (Data Dictionary, Locked, Can Modify Objects, Modify Search List, Override Coding, Version) • View Restriction • Parameters (Name, Value, Default)

For more information on protocol attributes, see Chapter 9, and the Design Help.

For more information on protocol parameters, see Chapter 5, and the Design Help.

Protocol Hierarchy

For all protocols in the database instance, the Protocol Hierarchy Report uses indentation to show parent-child protocol relationships and base-view protocol relationships. An indented protocol is the child of, or a view or a view on, a preceding protocol with lesser indentation.

The following example shows the Protocol Hierarchy Report:

The Protocol Hierarchy Report provides the following information:

Report column:	Description:
Protocol	Name of the protocol.
Type	Protocol type.

For more information on protocol hierarchy and protocol types, see Chapter 9, and the Design Help.

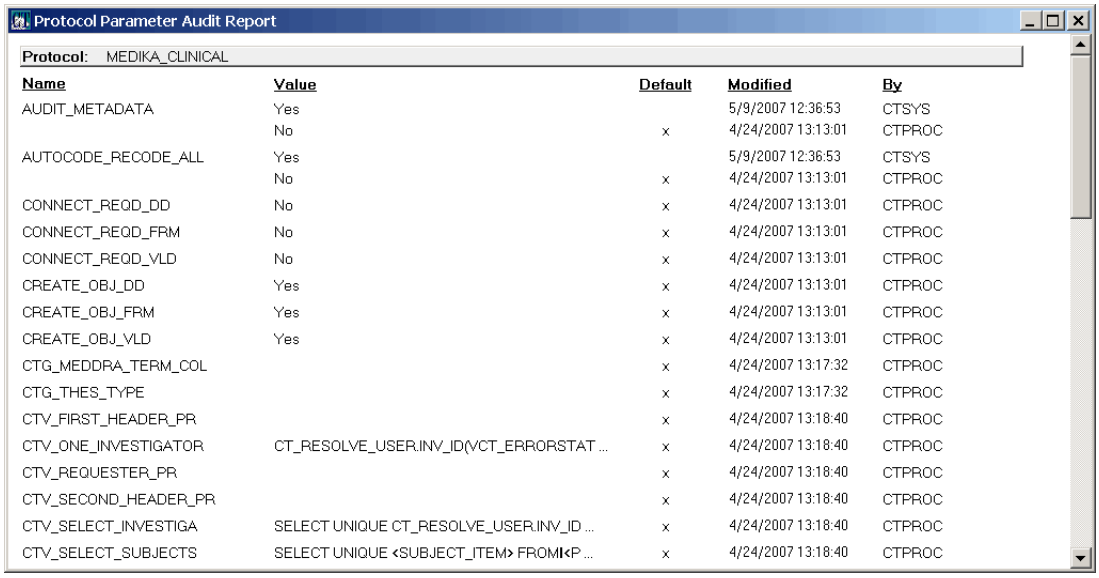
Protocol Parameter Audit

For selected protocols in the database instance, the Protocol Parameter Audit Report shows the protocol parameters along with the history of their settings.

Rows are grouped by protocol. A new header line appears for each protocol. Within each protocol, rows are sorted by Name and Date descending.

At least one row exists for each parameter, which displays the current value of the parameter. An “X” indicates which rows are for the default settings.

The following example shows the Protocol Parameter Audit Report:



Protocol Parameter Audit Report

Protocol: MEDIKA_CLINICAL

Name	Value	Default	Modified	By
AUDIT_METADATA	Yes		5/9/2007 12:36:53	CTSYS
	No	x	4/24/2007 13:13:01	CTPROC
AUTOCODE_RECODE_ALL	Yes		5/9/2007 12:36:53	CTSYS
	No	x	4/24/2007 13:13:01	CTPROC
CONNECT_REQD_DD	No	x	4/24/2007 13:13:01	CTPROC
CONNECT_REQD_FRM	No	x	4/24/2007 13:13:01	CTPROC
CONNECT_REQD_VLD	No	x	4/24/2007 13:13:01	CTPROC
CREATE_OBJ_DD	Yes	x	4/24/2007 13:13:01	CTPROC
CREATE_OBJ_FRM	Yes	x	4/24/2007 13:13:01	CTPROC
CREATE_OBJ_VLD	Yes	x	4/24/2007 13:13:01	CTPROC
CTG_MEDDRA_TERM_COL		x	4/24/2007 13:17:32	CTPROC
CTG_THES_TYPE		x	4/24/2007 13:17:32	CTPROC
CTV_FIRST_HEADER_PR		x	4/24/2007 13:18:40	CTPROC
CTV_ONE_INVESTIGATOR	CT_RESOLVE_USER.INV_ID(VCT_ERRORSTAT ...	x	4/24/2007 13:18:40	CTPROC
CTV_REQUESTER_PR		x	4/24/2007 13:18:40	CTPROC
CTV_SECOND_HEADER_PR		x	4/24/2007 13:18:40	CTPROC
CTV_SELECT_INVESTIGA	SELECT UNIQUE CT_RESOLVE_USER.INV_ID ...	x	4/24/2007 13:18:40	CTPROC
CTV_SELECT_SUBJECTS	SELECT UNIQUE <SUBJECT_ITEM> FROMI<P ...	x	4/24/2007 13:18:40	CTPROC

The Protocol Parameter Audit Report provides the following information:

Report column:	Description:
Name	Name of the protocol parameter.
Value	A value, if any, for the protocol parameter.
Default	Blank — The setting listed is not the default setting. x — The setting listed is the default setting.
Modified	Date and time of modification.
By	Name of user account that modified the protocol parameter.

Note: If a protocol is imported from a database with a later time-zone, it is possible that modifications made locally after import will appear to have occurred earlier.

For more information on protocol parameters, see Chapter 5, and the Design Help.

Codelist and checklist reports

The types of codelist and checklist reports with the available options are:

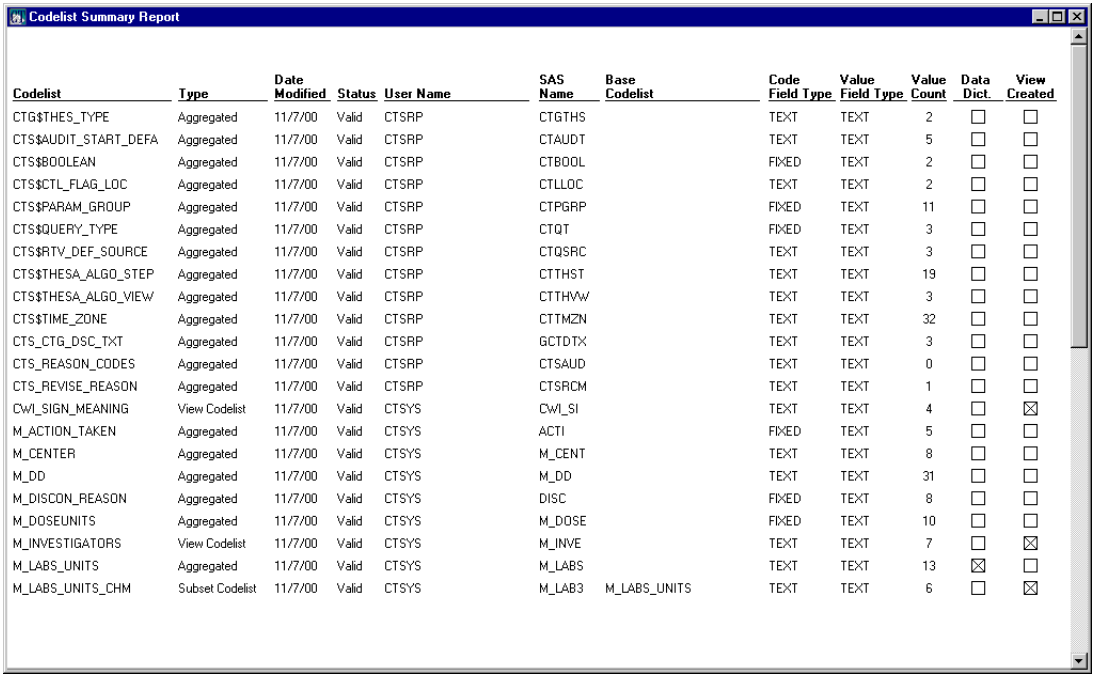
- Codelist Summary — Level of Detail option:
  - Basic
  - Detail
- Codelist Values — No options
- Codelist Usage by Codelist — Sort option:
  - By Protocol
  - By Panel
  - By Item
- Codelist Usage by Protocol — No options
- Codelist Value Usage — No options
- Checklist Value Usage — No options
- Codelist Hierarchy — No options

For more information on the Level of Detail and Sort options, see "Report options and formats" on page 492.

Codelist Summary

For the selected codelists in the database instance, the Codelist Summary Report displays all the codelist metadata as well as a count of the codelist values. This report also displays the additional attributes of view codelists, including base/subset relationships.

The following example shows the Codelist Summary Report with the Basic level of detail:



Codelist Summary Report

Codelist	Type	Date Modified	Status	User Name	SAS Name	Base Codelist	Code Field Type	Value Field Type	Value Count	Data Dict.	View Created
CTG$THES_TYPE	Aggregated	11/7/00	Valid	CTSRP	CTGTHS		TEXT	TEXT	2	☐	☐
CTS$AUDIT_START_DEFA	Aggregated	11/7/00	Valid	CTSRP	CTAUDT		TEXT	TEXT	5	☐	☐
CTS$BOOLEAN	Aggregated	11/7/00	Valid	CTSRP	CTBOOL		FIXED	TEXT	2	☐	☐
CTS$CTL_FLAG_LOC	Aggregated	11/7/00	Valid	CTSRP	CTLLOC		TEXT	TEXT	2	☐	☐
CTS$PARAM_GROUP	Aggregated	11/7/00	Valid	CTSRP	CTPGRP		FIXED	TEXT	11	☐	☐
CTS$QUERY_TYPE	Aggregated	11/7/00	Valid	CTSRP	CTQT		FIXED	TEXT	3	☐	☐
CTS$RTV_DEF_SOURCE	Aggregated	11/7/00	Valid	CTSRP	CTQSRC		TEXT	TEXT	3	☐	☐
CTS$THESA_ALGO_STEP	Aggregated	11/7/00	Valid	CTSRP	CTTHST		TEXT	TEXT	19	☐	☐
CTS$THESA_ALGO_VIEW	Aggregated	11/7/00	Valid	CTSRP	CTTHVW		TEXT	TEXT	3	☐	☐
CTS$TIME_ZONE	Aggregated	11/7/00	Valid	CTSRP	CTTMZN		TEXT	TEXT	32	☐	☐
CTS_CTG_DSC_TXT	Aggregated	11/7/00	Valid	CTSRP	GCTDTX		TEXT	TEXT	3	☐	☐
CTS_REASON_CODES	Aggregated	11/7/00	Valid	CTSRP	CTSAUD		TEXT	TEXT	0	☐	☐
CTS_REVISE_REASON	Aggregated	11/7/00	Valid	CTSRP	CTSRCM		TEXT	TEXT	1	☐	☐
CWI_SIGN_MEANING	View Codelist	11/7/00	Valid	CTSYS	CWI_SI		TEXT	TEXT	4	☐	☒
M_ACTION_TAKEN	Aggregated	11/7/00	Valid	CTSYS	ACTI		FIXED	TEXT	5	☐	☐
M_CENTER	Aggregated	11/7/00	Valid	CTSYS	M_CENT		TEXT	TEXT	8	☐	☐
M_DD	Aggregated	11/7/00	Valid	CTSYS	M_DD		TEXT	TEXT	31	☐	☐
M_DISCON_REASON	Aggregated	11/7/00	Valid	CTSYS	DISC		FIXED	TEXT	8	☐	☐
M_DOSEUNITS	Aggregated	11/7/00	Valid	CTSYS	M_DOSE		FIXED	TEXT	10	☐	☐
M_INVESTIGATORS	View Codelist	11/7/00	Valid	CTSYS	M_INVE		TEXT	TEXT	7	☐	☒
M_LABS_UNITS	Aggregated	11/7/00	Valid	CTSYS	M_LABS		TEXT	TEXT	13	☒	☐
M_LABS_UNITS_CHM	Subset Codelist	11/7/00	Valid	CTSYS	M_LAB3	M_LABS_UNITS	TEXT	TEXT	6	☐	☒

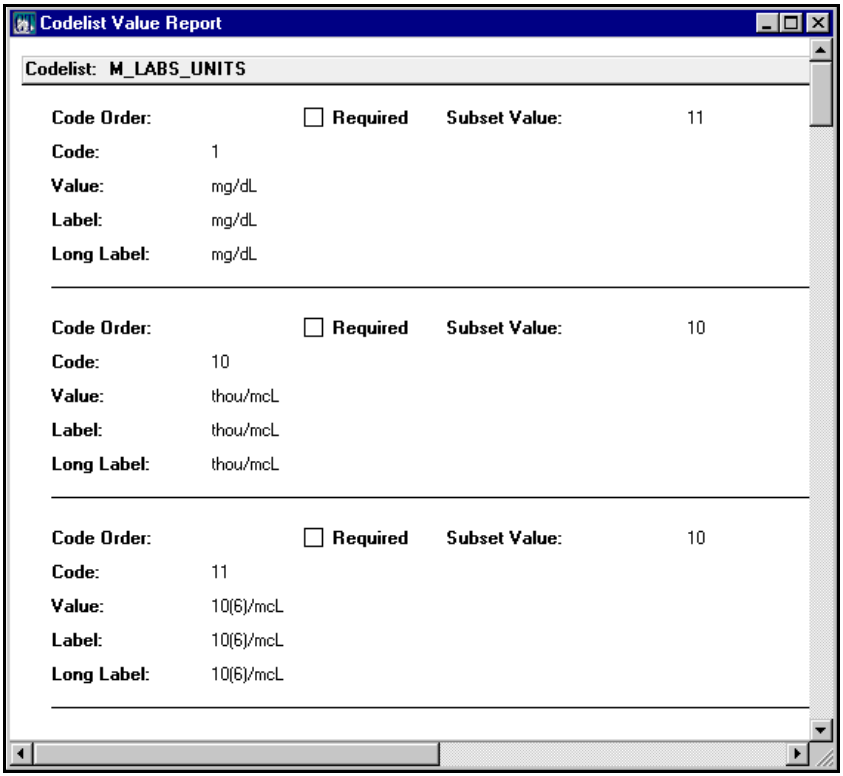
The Codelist Summary Report provides the following information:

Report column:	Description:
Codelist	Name of the codelist.
Codelist attributes	Level of Detail — Basic • Type • Date Modified • Status • User Name • SAS Name • Base Codelist (if subset codelist) • Code Field Type • Value Field Type • Value Count • Data Dict. (check box) • View Created (check box) Level of Detail — Detail • Type • Date Modified • Status • User Name • SAS Name • Code Field Type • Value Field Type • Value Count • Data Dict. (check box) • View Created (check box) • View Restriction (if view codelist — Additional Properties: Base Protocol, Base Panel, Non-Clintrial Base Table, Status, Code, Value, Label, Long Label, Status, Code Order, Subset Required, Subset Value)

For more information on codelist attributes, view codelists, and subset codelists, see Chapter 12, and the Design Help.

For selected codelists, the Codelist Value Report displays codes, values, labels, ordering, and subset attributes. View codelist values are included.

The following example shows the Codelist Value Report:



The Codelist Value Report provides the following information:

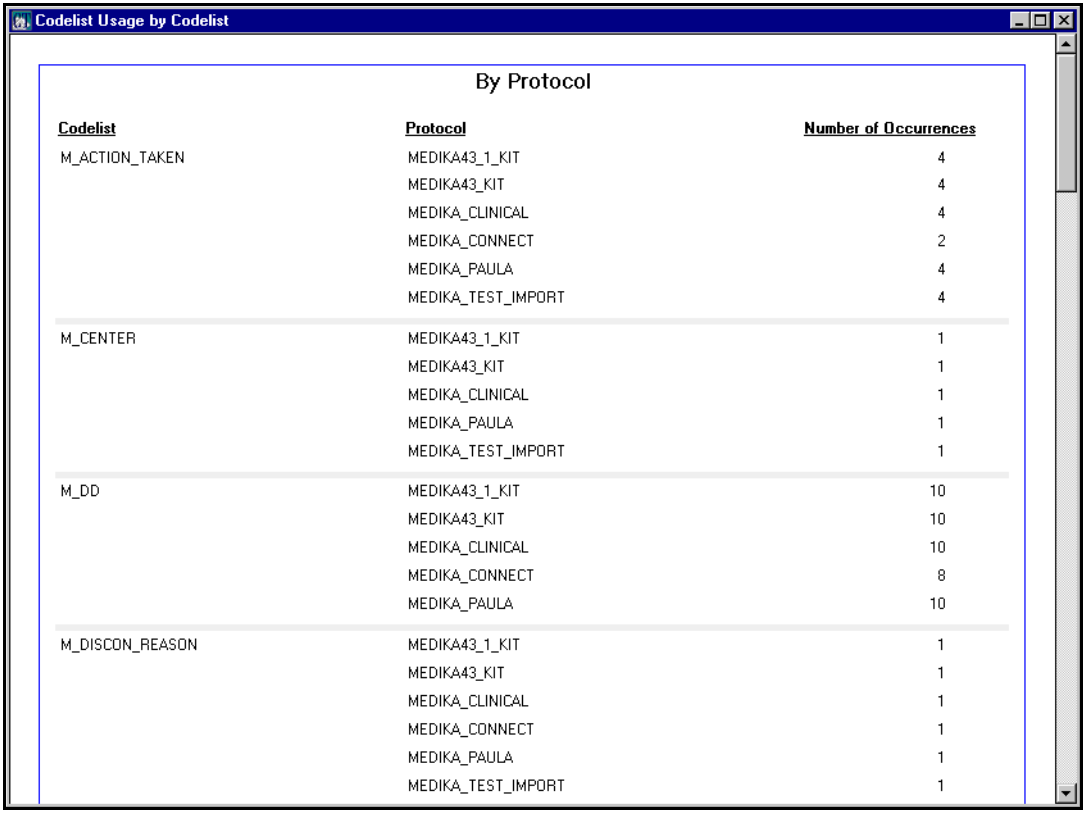
Report column:	Description:
Codelist	Name of the codelist.
Attributes	• Code Order • Code • Value • Label • Long Label • Required (check box) • Subset Value

For more information on codelist attributes, see Chapter 12, and the Design Help.

Codelist Usage by Codelist

For selected codelists, the Codelist Usage by Codelist Report displays codelists sorted by the object (protocol, panel or item) that uses the codelist, including usage as a checklist.

The following example shows the Codelist Usage by Codelist Report, sorted by protocol:



Codelist Usage by Codelist

By Protocol

Codelist	Protocol	Number of Occurrences
M_ACTION_TAKEN	MEDIKA43_1_KIT	4
	MEDIKA43_KIT	4
	MEDIKA_CLINICAL	4
	MEDIKA_CONNECT	2
	MEDIKA_PAULA	4
	MEDIKA_TEST_IMPORT	4
M_CENTER	MEDIKA43_1_KIT	1
	MEDIKA43_KIT	1
	MEDIKA_CLINICAL	1
	MEDIKA_PAULA	1
	MEDIKA_TEST_IMPORT	1
M_DD	MEDIKA43_1_KIT	10
	MEDIKA43_KIT	10
	MEDIKA_CLINICAL	10
	MEDIKA_CONNECT	8
	MEDIKA_PAULA	10
M_DISCON_REASON	MEDIKA43_1_KIT	1
	MEDIKA43_KIT	1
	MEDIKA_CLINICAL	1
	MEDIKA_CONNECT	1
	MEDIKA_PAULA	1
	MEDIKA_TEST_IMPORT	1

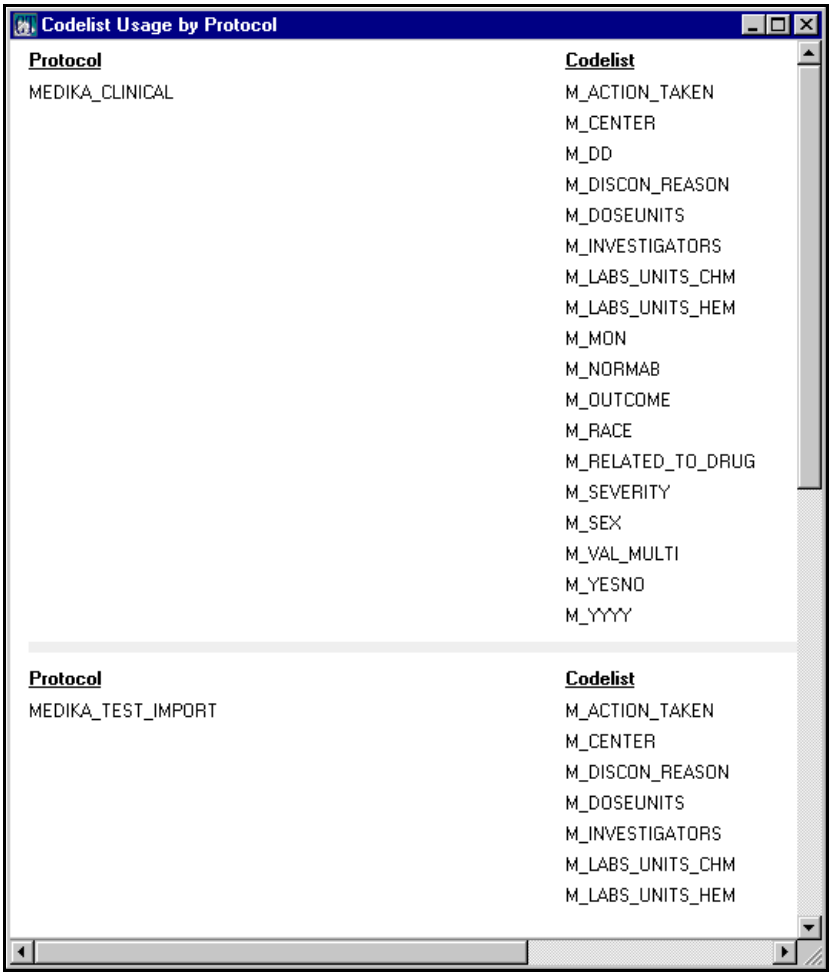
The Codelist Usage by Codelist Report provides the following information according to the sort option selected:

Sort option:	Description:
By Protocol	Sort columns: • Codelist — Name of the codelist. • Protocol — Name of the protocol containing the panel and item that use the codelist. • Number of Occurrences — Number of times the codelist is used in data entry in the Enter study pages for the listed protocol.
By Codelist	Sort columns: • Codelist — Name of the codelist. • Protocol — Name of the protocol containing the panel and item that use the codelist. • Panel — Name of the panel containing the item that uses the codelist. • Panel References — Number of times the codelist is used by items in the listed panel.
By Item	Sort columns: • Codelist — Name of the codelist. • Protocol — Name of the protocol containing the panel and item that use the codelist. • Panel — Name of the panel containing the item that uses the codelist. • Item — Name of the item that uses the codelist.

Codelist Usage by Protocol

For selected protocols, the Codelist Usage by Protocols Report lists all the codelists to which the selected protocols refer.

The following example shows the Codelist Usage by Protocol Report:



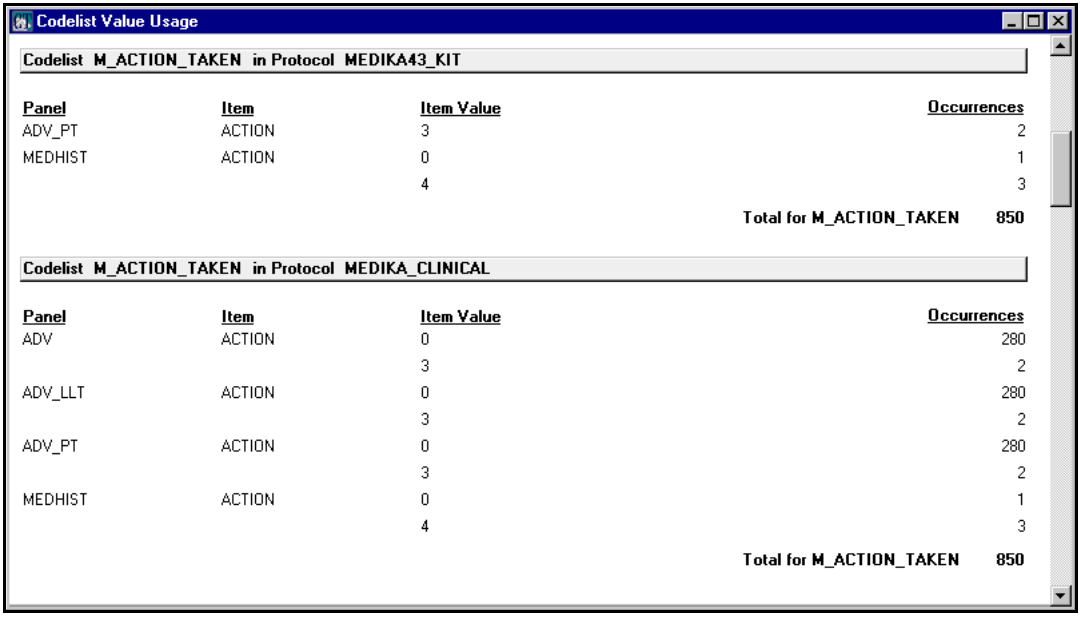
The Codelist Usage by Protocol Report provides the following information:

Report column:	Description:
Protocol	Name of the protocol referencing the codelist.
Codelist	Name of the codelist used by the protocol.

Codelist Value Usage

For selected codelists, the Codelist Value Usage Report displays the protocols, panels, and item names that use the specified codelists, as well as the codelist codes found in the clinical data, and a count of the number of occurrences of each item value in Enter study pages.

The following example shows the Codelist Value Usage Report:



Codelist Value Usage

Codelist M_ACTION_TAKEN in Protocol MEDIKA43_KIT

Panel	Item	Item Value	Occurrences
ADV_PT	ACTION	3	2
MEDHIST	ACTION	0	1
		4	3
		Total for M_ACTION_TAKEN	850

Codelist M_ACTION_TAKEN in Protocol MEDIKA_CLINICAL

Panel	Item	Item Value	Occurrences
ADV	ACTION	0	280
		3	2
ADV_LLT	ACTION	0	280
		3	2
ADV_PT	ACTION	0	280
		3	2
MEDHIST	ACTION	0	1
		4	3
		Total for M_ACTION_TAKEN	850

The Codelist Value Usage Report provides the following information:

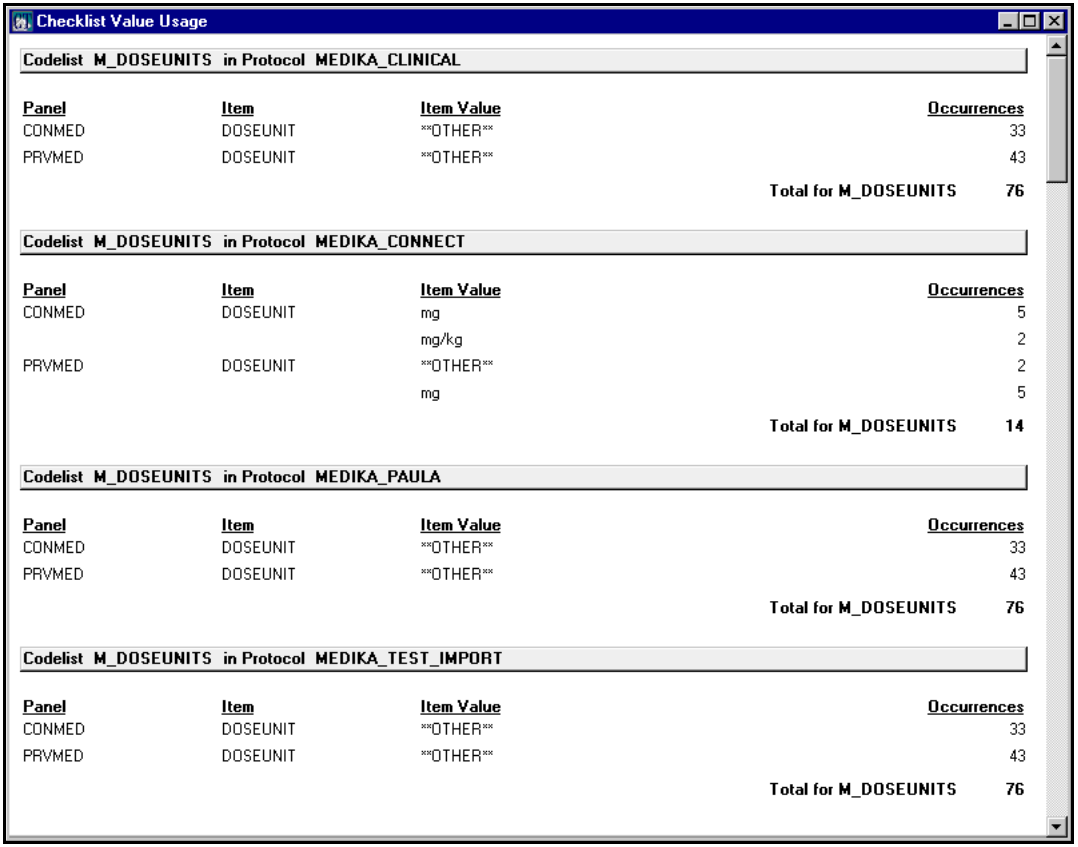
Report column:	Description:
Codelist <*X*> in Protocol <*Y*>	The name of the codelist containing a codelist value used in study pages with clinical data, and the name of the protocol containing panels with items that use this codelist value.
Panel	Name of the panel containing items that use the listed codelist value.
Item	Name of the item that uses the codelist value.
Item Value	The specific codelist value that is used in study pages with clinical data. The number of times this value is used is the number that appears in the Occurrences column.
Occurrences	Number of times an item value is used in study pages with clinical data.

For more information on codelist values, see Chapter 12, and the Design Help.

Checklist Value Usage

For selected codelists, the Checklist Value Usage Report displays the protocols, panels, and items that use the specified codelist as a checklist, as well as codelist values found in the clinical data, and an occurrence count.

The following example shows the Checklist Value Usage Report:



Checklist Value Usage

Codelist M_DOSEUNITS in Protocol MEDIKA_CLINICAL

Panel	Item	Item Value	Occurrences
CONMED	DOSEUNIT	**OTHER**	33
PRVMED	DOSEUNIT	**OTHER**	43
		Total for M_DOSEUNITS	76

Codelist M_DOSEUNITS in Protocol MEDIKA_CONNECT

Panel	Item	Item Value	Occurrences
CONMED	DOSEUNIT	mg	5
		mg/kg	2
PRVMED	DOSEUNIT	**OTHER**	2
		mg	5
		Total for M_DOSEUNITS	14

Codelist M_DOSEUNITS in Protocol MEDIKA_PAULA

Panel	Item	Item Value	Occurrences
CONMED	DOSEUNIT	**OTHER**	33
PRVMED	DOSEUNIT	**OTHER**	43
		Total for M_DOSEUNITS	76

Codelist M_DOSEUNITS in Protocol MEDIKA_TEST_IMPORT

Panel	Item	Item Value	Occurrences
CONMED	DOSEUNIT	**OTHER**	33
PRVMED	DOSEUNIT	**OTHER**	43
		Total for M_DOSEUNITS	76

The Checklist Value Usage Report provides the following information:

Report column:	**Description:**
Codelist <*X*> in Protocol <*Y*>	The name of the codelist that functions as a checklist having a value used in study pages with clinical data, and the name of the protocol containing panels with items that use this checklist value.
Panel	Name of the panel containing items that use the listed checklist value.
Item	Name of the item that uses the checklist value.

Report column:	Description:
Item Value	The checklist value that is used in study pages with clinical data. Values not specified in the codelist itself (that is, entered manually instead of selected from the list of values) are grouped together as **OTHER** under Item Value for the occurrence count.
Occurrences	Number of times an item value is used in study pages with clinical data.

For more information on checklists, see Chapter 10, Chapter 12, and the Design Help.

Codelist Hierarchy

The Codelist Hierarchy Report uses indentation to display a hierarchical view of subset codelists, their subset restrictions, and the codelists on which they are based. A codelist that is indented is a subset codelist based on the preceding codelist with less indentation.

The following example shows the Codelist Hierarchy Report:

The Codelist Hierarchy Report provides the following information:

Report column:	Description:
Codelist	Name of the codelist that is the base codelist for subset codelists and (indented) the names of all subset codelists that have that codelist as a base codelist.
Dict. (check box)	If selected, indicates that the codelist is a dictionary codelist. (A base codelist for subset codelists must be a dictionary codelist.)
Subset Restriction	Subset restriction clause that defines the subset codelist.

Panel reports

The types of panel reports with the available options are as follows:

- Panel Detail Report — Level of detail and implementation:
  - Basic
  - Detail
  - Include Unimplemented Panels
- Item and Coding Target Report — Level of detail and implementation:
  - Basic
  - Detail
  - Include Unimplemented Items and Coding Targets
- Rules — Include Unimplemented Rules
- Derivations — Include Unimplemented Derivations

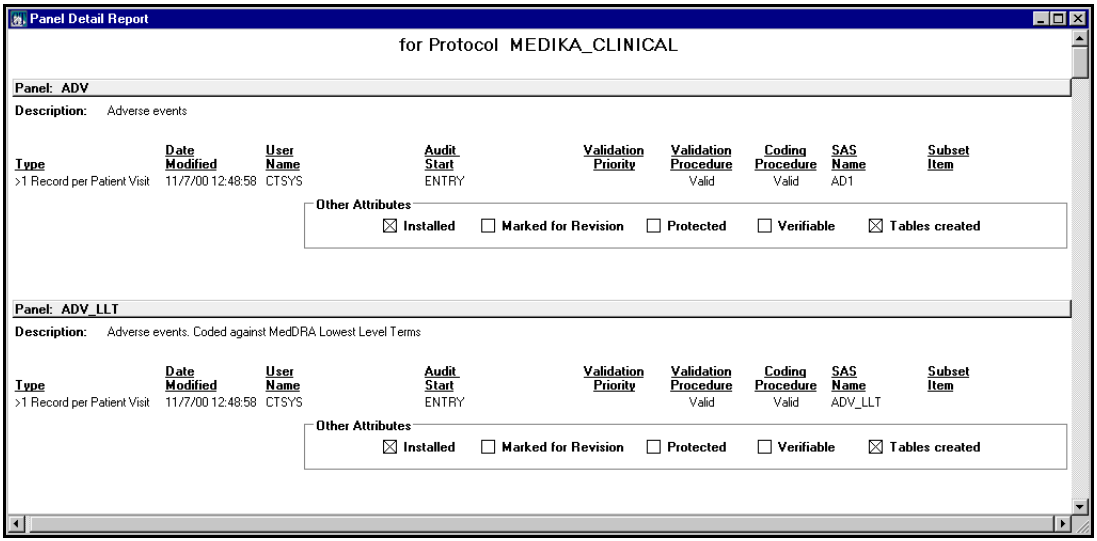
Note: If you select the option to include unimplemented objects, the panel report displays information for panels marked for revision in their current state of revision.

For more information on the Level of Detail options, see "Report options and formats" on page 492.

Panel Detail

The Panel Detail Report lists all the panels in the current protocol and displays their attributes, including master-detail relationships. For view panels, this report displays the restriction clause belonging to the panel or to the containing protocol, if either type of restriction clause exists. This report also displays a list of the items that each panel contains, and the status of each panel's validation and encoding procedures.

The following example shows the Panel Detail Report:



The Panel Detail Report provides the following information:

Report column:	Description:
Panel	Name of the panel.
Description	Description of panel.

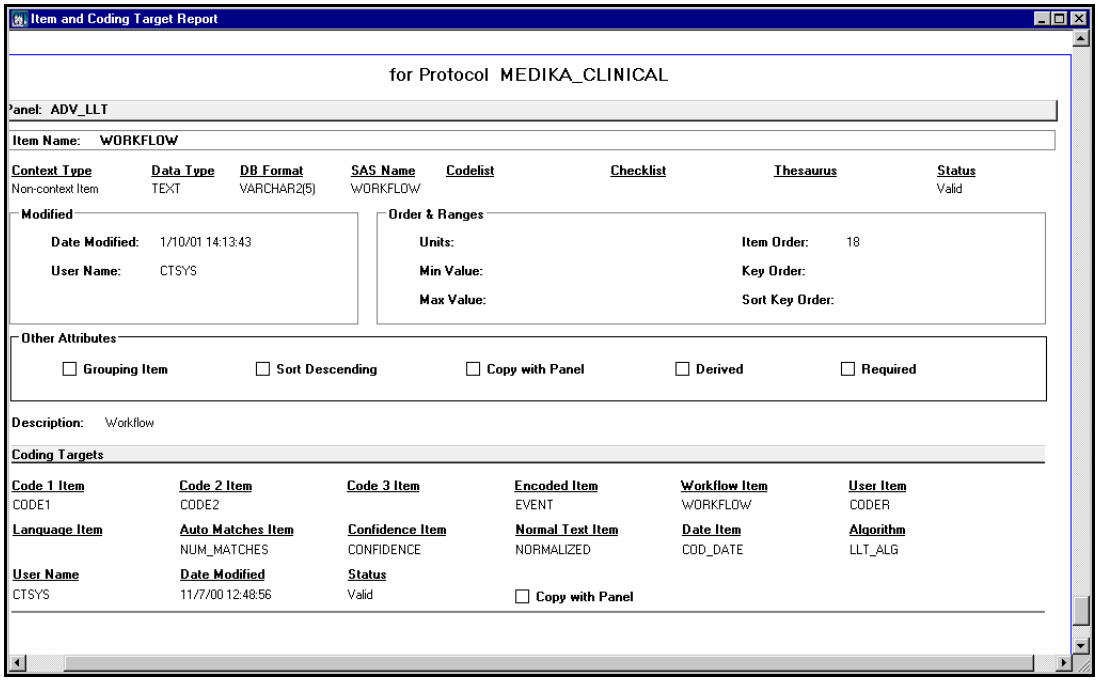
Report column:	Description:
Panel attributes	Level of Detail — Basic: • Type • Date Modified • User Name • Audit Start • Validation Priority • Validation Procedure • Coding Procedure • SAS Name • Subset Item • Other Attributes (check boxes — Installed, Marked for Revision, Protected, Verifiable, Tables created) Level of Detail — Detail • Type • User Name • Date Modified • Audit Start • SAS Name • Subset Item • Validation and Coding (Validation Priority, Validation Procedure, Coding Procedure) • Other Attributes (check boxes — Installed, Verifiable, Protected, Marked for Revision, Tables created) • View Restriction • Panel Items

For more information on panel attributes, see Chapter 10, and the Design Help.

Item and Coding Target

The Item and Coding Target Report displays the attributes of the selected items and coding targets for the current protocol.

The following example shows the Item and Coding Target Report:



The Items and Coding Target Report provides the following information:

Report column:	Description:
Panel	Name of panel.
Item attributes	Level of Detail — Basic: • Item Name • Data Type • DB Format • Required • Derived • Codelist • Checklist • SAS Name • Key Order • Description Level of Detail — Detail • Item Name • Context Type • Data Type • DB Format • SAS Name • Codelist • Checklist • Thesaurus • Status • Modified (Date Modified, User Name) • Order and Ranges (Units, Min. Value, Max. Value, Item Order, Key Order, Sort Key Order) • Other Attributes (Check boxes — Grouping Item, Sort Descending, Copy with Panel, Derived, Required) • Description

Report column:	Description:
Coding Target Attributes	Level of detail — Detail • Code 1 Item • Code 2 Item • Code 3 Item • Encoded Item • Workflow Item • User Item • Language Item • Auto Matches Item • Confidence Item • Normalized Text Item • Date Item • Algorithm • User Name • Date Modified • Status • Copy with Panel (check box)

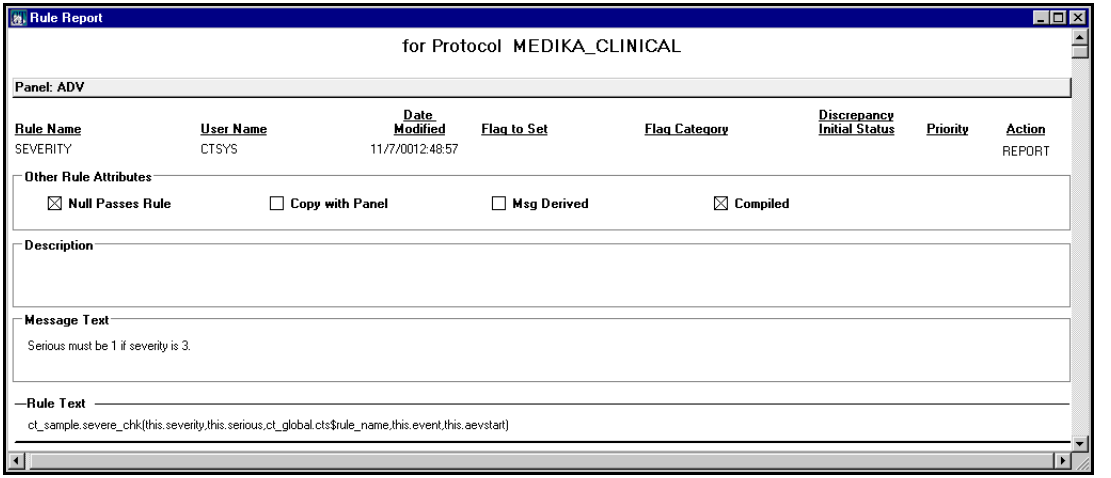
Note: To display coding target attributes, in the Report Options dialog box, you must select Detail level of detail and the Include Unimplemented Items and Coding Targets check box.

For more information on item attributes, see Chapter 10, and the Design Help.

For more information on coding target attributes, see Chapter 23, and the Design Help.

The Rule Report displays the attributes and the text of the rules for the current protocol.

The following example shows the Rule Report:



The Rule Report provides the following information:

Report column:	Description:
Panel	Name of panel containing the listed rules.
Rule attributes	• Rule Name • User Name • Date Modified • Flag to Set • Flag Category • Discrepancy Initial Status • Priority • Action • Other Rule Attributes (check boxes — Null Passes Rule, Copy with Panel, Msg Derived, Compiled) • Description • Message Text • Rule Text

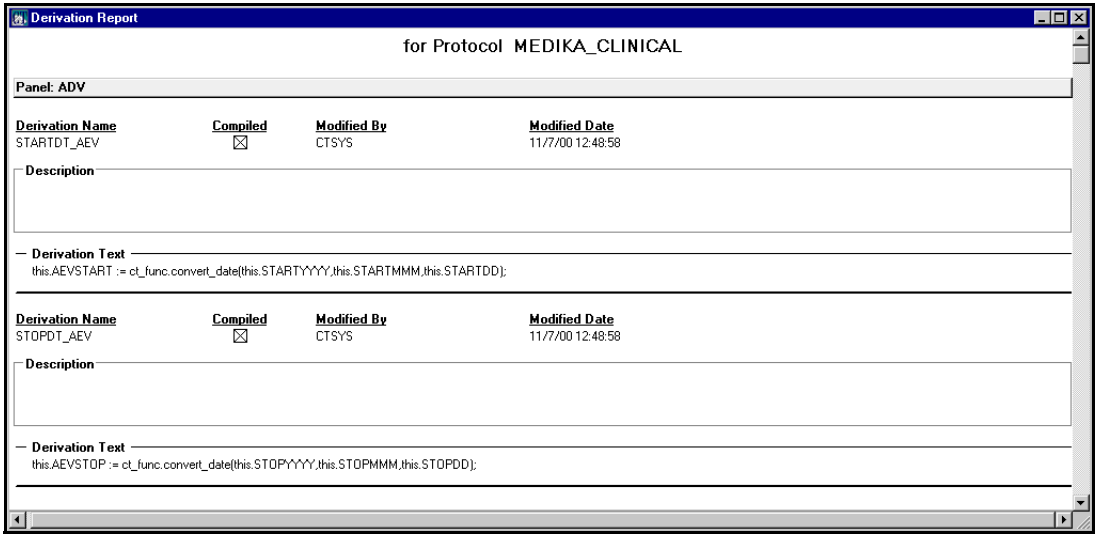
For more information on rule attributes, see Chapter 11, and the Design Help.

Derivation

The Derivation Report displays the attributes and the text of the derivations for the current protocol.

Note: The Derivation Report also runs in Manage.

The following example shows the Derivation Report:



The Derivation Report provides the following information:

Report column:	Description:
Panel	Name of panel containing the listed derivations.
Derivation attributes	• Derivation Name • Compiled (check box) • Modified By • Modified Date • Description • Derivation Text

For more information on derivation attributes, see Chapter 11, and the Design Help.

Display Objects reports

The types of display object reports with the available options are as follows:

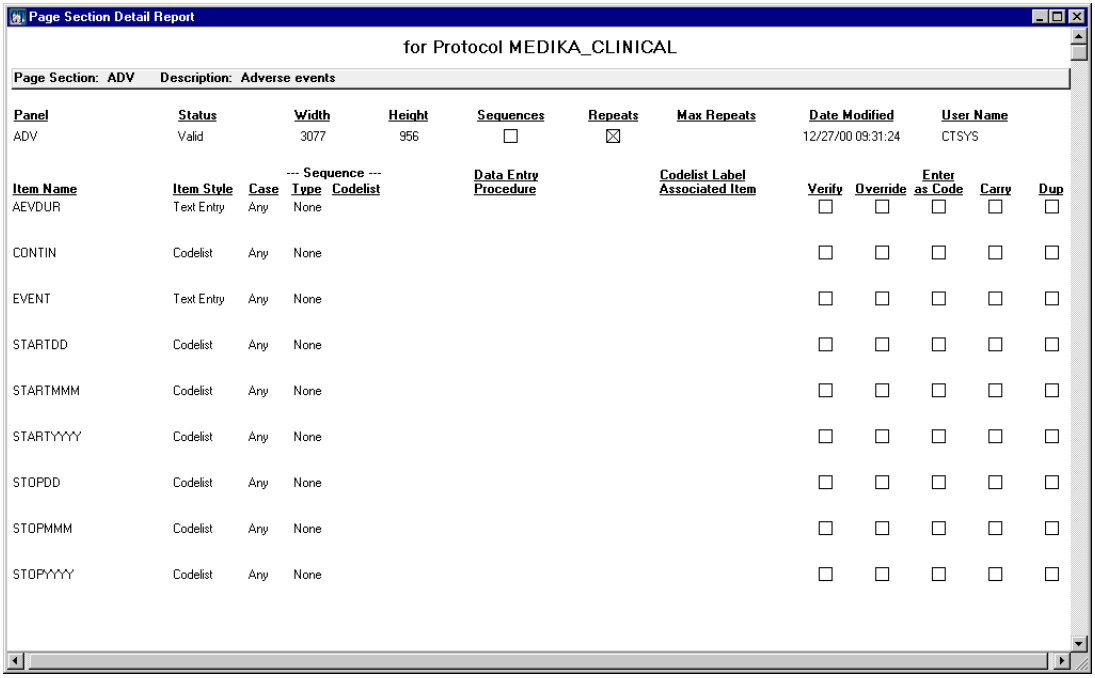
- Page Section Detail Report — No options.
- Page Template Summary — No options.
- Page Template Detail — No options.
- User Procedure Report — No options.
- Study Book Report — Level of Detail:
  - Summary
  - Basic
  - Detail

For more information on the Level of Detail options, see "Report options and formats" on page 492.

Page Section Detail

The Page Section Detail Report displays all attributes of page sections and page section items for the current protocol.

The following example shows the Page Section Detail Report:



Page Section Detail Report

for Protocol MEDIKA_CLINICAL

Page Section: ADV Description: Adverse events

Panel	Status	Width	Height	Sequences	Repeats	Max Repeats	Date Modified	User Name
ADV	Valid	3077	956	☐	☒		12/27/00 09:31:24	CTSYS

Item Name	Item Style	Case	--- Sequence --- Type	Codelist	Data Entry Procedure	Codelist Label Associated Item	Verify	Override	Enter as Code	Carry	Dup
AEVDUR	Text Entry	Any	None				☐	☐	☐	☐	☐
CONTIN	Codelist	Any	None				☐	☐	☐	☐	☐
EVENT	Text Entry	Any	None				☐	☐	☐	☐	☐
STARTDD	Codelist	Any	None				☐	☐	☐	☐	☐
STARTMMM	Codelist	Any	None				☐	☐	☐	☐	☐
STARTYYYY	Codelist	Any	None				☐	☐	☐	☐	☐
STOPDD	Codelist	Any	None				☐	☐	☐	☐	☐
STOPMMM	Codelist	Any	None				☐	☐	☐	☐	☐
STOPYYYY	Codelist	Any	None				☐	☐	☐	☐	☐

The Page Section Detail Report provides the following information:

Report column:	**Description:**
Page Section	Name of the page section.
Description	A description of the page section.

Report column:	Description:
Page section attributes	• Panel • Status • Width • Height • Sequences (check box) • Repeats (check box) • Max Repeats • Date Modified • User Name Sequence • Item Name • Item Style • Case • Type • Codelist • Data Entry Procedure • Codelist Label Associated Item • Verify (check box) • Override (check box) • Enter as Code (check box) • Carry (check box) • Dup (check box)

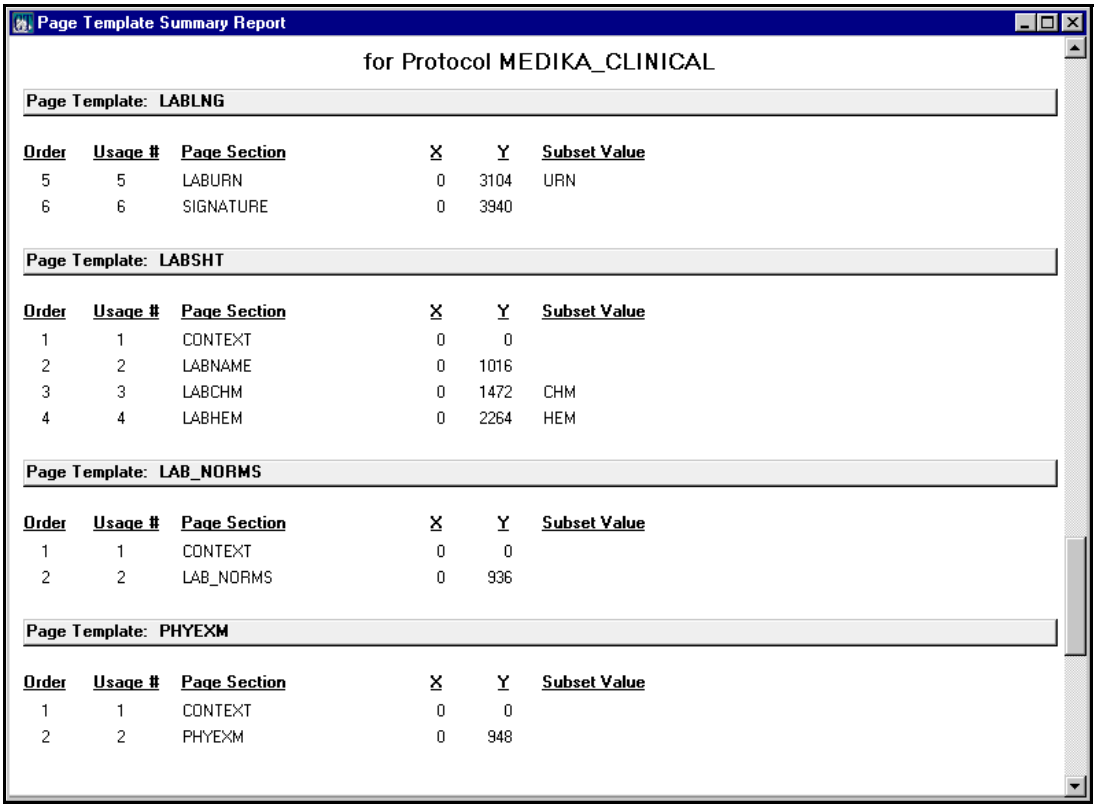
For information on page section attributes, see Chapter 22.

Page Template Summary

For the current protocol, the Page Template Summary Report displays the page sections and panels used by each page template. In addition, this report includes any subset values used by the page section.

Note: You can also display this report in Manage.

The following example shows the Page Template Summary Report:



Page Template Summary Report

for Protocol MEDIKA_CLINICAL

Page Template: LABLNG

Order	Usage #	Page Section	X	Y	Subset Value
5	5	LABURN	0	3104	URN
6	6	SIGNATURE	0	3940	

Page Template: LABSHT

Order	Usage #	Page Section	X	Y	Subset Value
1	1	CONTEXT	0	0	
2	2	LABNAME	0	1016	
3	3	LABCHM	0	1472	CHM
4	4	LABHEM	0	2264	HEM

Page Template: LAB_NORMS

Order	Usage #	Page Section	X	Y	Subset Value
1	1	CONTEXT	0	0	
2	2	LAB_NORMS	0	936	

Page Template: PHYEXM

Order	Usage #	Page Section	X	Y	Subset Value
1	1	CONTEXT	0	0	
2	2	PHYEXM	0	948	

The Page Template Summary Report provides the following information:

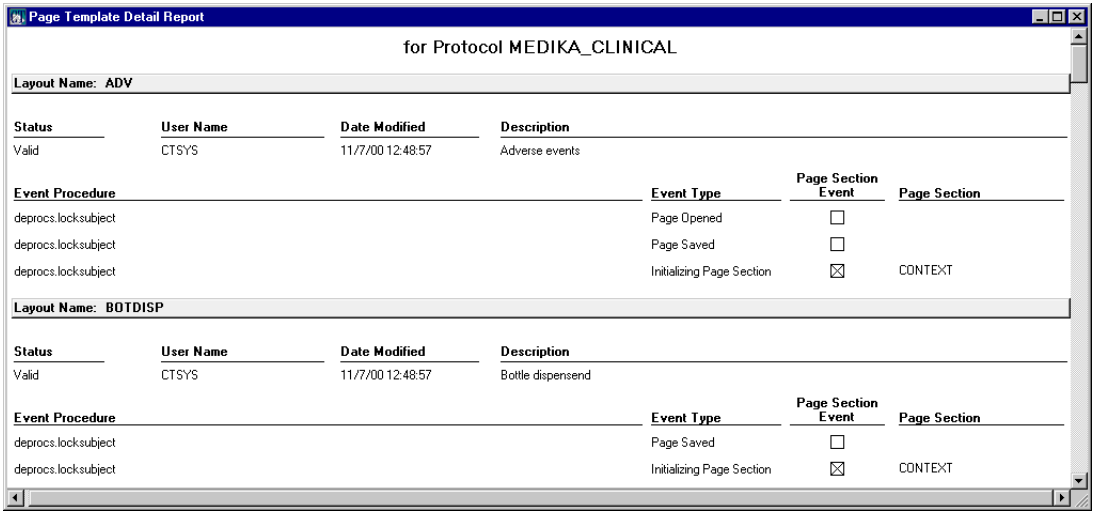
Report column:	Description:
Page Template	Name of page template.
Page template attributes	• Order • Usage # • Page Section • X [coordinate] • Y [coordinate] • Subset Value

For information on page template attributes, see Chapter 16.

Page Template Detail

For the current protocol, the Page Template Detail Report displays the attributes of the selected page templates, as well as page template events and page section usage events.

The following example shows the Page Template Detail Report:



Page Template Detail Report

for Protocol MEDIKA_CLINICAL

Layout Name: ADV

Status	User Name	Date Modified	Description
Valid	CTSYS	11/7/00 12:48:57	Adverse events

Event Procedure	Event Type	Page Section Event	Page Section
deprocs.locksubject	Page Opened	☐	
deprocs.locksubject	Page Saved	☐	
deprocs.locksubject	Initializing Page Section	☒	CONTEXT

Layout Name: BOTDISP

Status	User Name	Date Modified	Description
Valid	CTSYS	11/7/00 12:48:57	Bottle dispensend

Event Procedure	Event Type	Page Section Event	Page Section
deprocs.locksubject	Page Saved	☐	
deprocs.locksubject	Initializing Page Section	☒	CONTEXT

The Page Template Detail Report provides the following information:

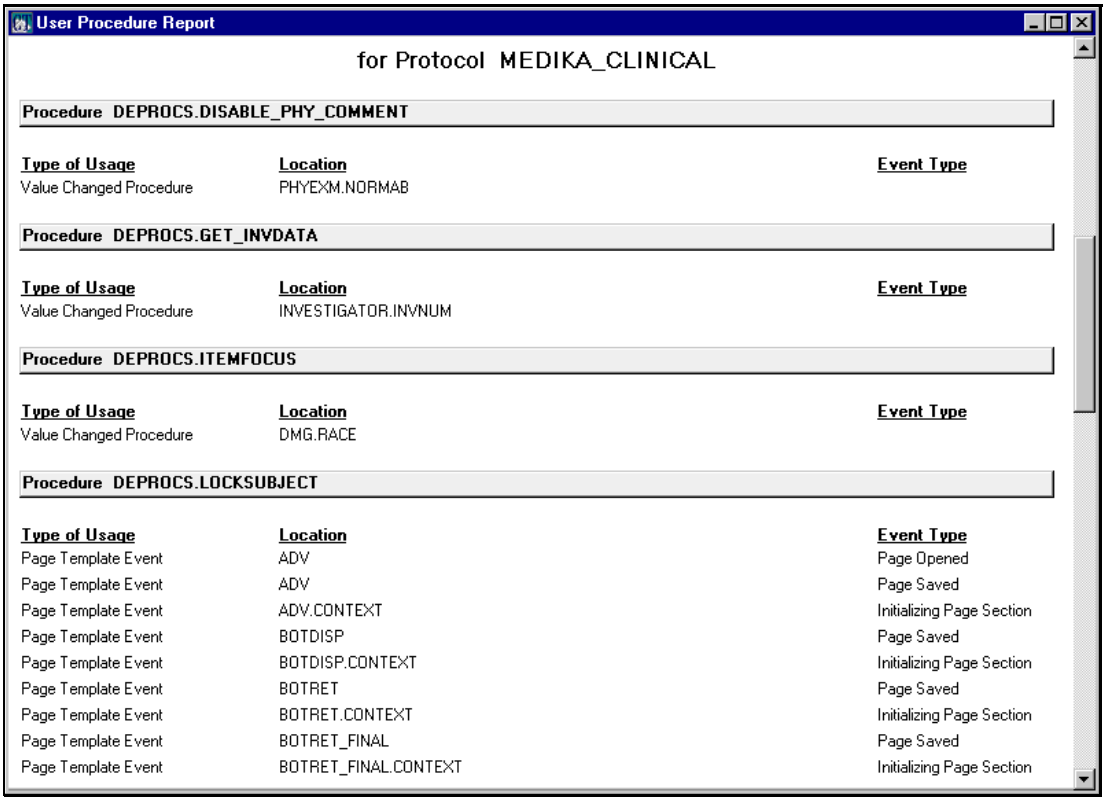
Report column:	**Description:**
Layout Name	Name of the page template.
Page template attributes	• Status • User Name • Date Modified • Description • Event Procedure • Event Type • Page Section Event (check box) • Page Section

For information on page template attributes, see Chapter 16.

User Procedure

For the current protocol, the User Procedure Report displays the types of usage for user procedures, as well as their location and the event that triggers the procedure.

The following example shows the User Procedure Report:



User Procedure Report

for Protocol MEDIKA_CLINICAL

Procedure DEPROCS.DISABLE_PHY_COMMENT

Type of Usage	Location	Event Type
Value Changed Procedure	PHYEXM.NORMAB	

Procedure DEPROCS.GET_INVDATA

Type of Usage	Location	Event Type
Value Changed Procedure	INVESTIGATOR.INVNUM	

Procedure DEPROCS.ITEMFOCUS

Type of Usage	Location	Event Type
Value Changed Procedure	DMG.RACE	

Procedure DEPROCS.LOCKSUBJECT

Type of Usage	Location	Event Type
Page Template Event	ADV	Page Opened
Page Template Event	ADV	Page Saved
Page Template Event	ADV.CONTEXT	Initializing Page Section
Page Template Event	BOTDISP	Page Saved
Page Template Event	BOTDISP.CONTEXT	Initializing Page Section
Page Template Event	BOTRET	Page Saved
Page Template Event	BOTRET.CONTEXT	Initializing Page Section
Page Template Event	BOTRET_FINAL	Page Saved
Page Template Event	BOTRET_FINAL.CONTEXT	Initializing Page Section

The User Procedure Report provides the following information:

Report column:	Description:
Procedure	Name of the procedure (in the form *PACKAGE_NAME.PROCEDURE_NAME*)
Type of Usage	Type of procedure according to usage. (For example, Page Template Event procedure.)

Report column:	Description:
Location	Depending on the value in the Type of Usage field, Location is in one of the following forms: • PAGE_TEMPLATE_NAME • PAGE_TEMPLATE_NAME.PAGE_SECTION_NAME • PAGE_SECTION_NAME.ITEM_NAME
Event Type	Type of event that triggers the user procedure.

For information on user procedures, see Chapter 11.

Study Book

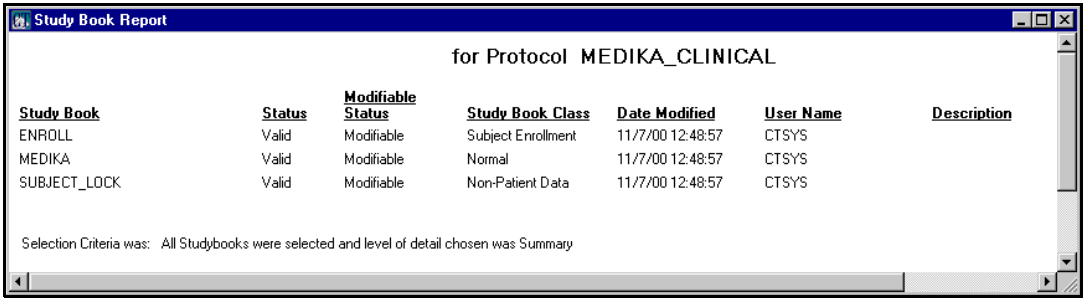
For the current protocol, the Study Book Report displays attributes of:

- Study books — Summary level of detail.
- Study blocks under each study book — Basic and Detail level of detail.
- Study pages under each study block — Basic level of detail.

The display order within each study book matches the order in the Enter Navigator. Specifically, the sort is by block order, block repeat order, page order, and page repeat order.

Note: This report can also be accessed in Manage.

The following example shows the Study Book Report with Summary level of detail:



For more information on the Level of Detail option, see "Report options and formats" on page 492.

The Study Book Report provides the following information:

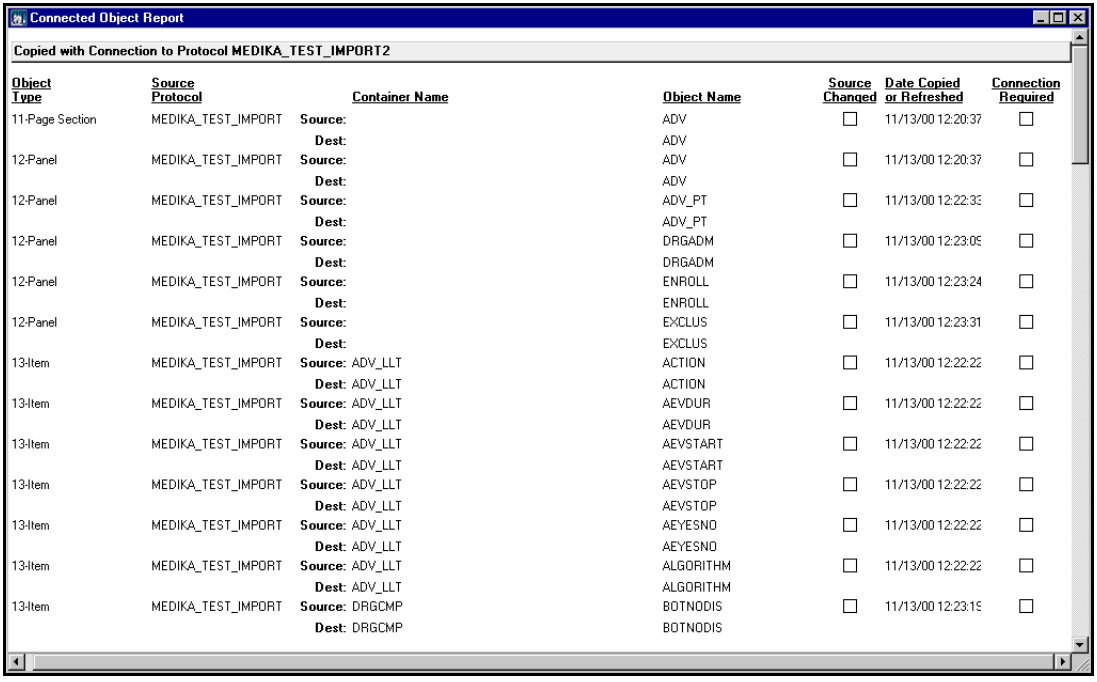
Report column:	Description:
Study Book	Name of the study book.
Study book attributes	Level of Detail — Summary: • Status • Modifiable Status • Study Book Class • Date Modified • User Name • Description Level of Detail — Basic: • Block Name • Block Key • Repeats (Y/N) • Order • Title (Study Page) • Template • Has Repeats • Panel • Section Level of Detail — Detail • Study Book Attributes (Class, Date Modified, Status, User Name, Description) • Block Title • Block Attributes (Block Key Value, Date Modified, Block Order, User Name, Has Repeats [check box], Max Repeats) • Block Repeat Key Value • Block Repeat Order • Date Modified • User Name

For information on study book attributes, see Chapter 18.

Connected Object Report

The Connected Object Report displays a list of all object connections and their statuses for the current protocol, provided that either the source object or the destination object is in the protocol. (No indirect connections outside the current protocol are displayed.)

The following example shows the Connected Object Report:



Connected Object Report

Copied with Connection to Protocol MEDIKA_TEST_IMPORT2

Object Type	Source Protocol		Container Name	Object Name	Source Changed	Date Copied or Refreshed	Connection Required
11-Page Section	MEDIKA_TEST_IMPORT	Source:		ADV	☐	11/13/00 12:20:37	☐
		Dest:		ADV			
12-Panel	MEDIKA_TEST_IMPORT	Source:		ADV	☐	11/13/00 12:20:37	☐
		Dest:		ADV			
12-Panel	MEDIKA_TEST_IMPORT	Source:		ADV_PT	☐	11/13/00 12:22:33	☐
		Dest:		ADV_PT			
12-Panel	MEDIKA_TEST_IMPORT	Source:		DRGADM	☐	11/13/00 12:23:09	☐
		Dest:		DRGADM			
12-Panel	MEDIKA_TEST_IMPORT	Source:		ENROLL	☐	11/13/00 12:23:24	☐
		Dest:		ENROLL			
12-Panel	MEDIKA_TEST_IMPORT	Source:		EXCLUS	☐	11/13/00 12:23:31	☐
		Dest:		EXCLUS			
13-Item	MEDIKA_TEST_IMPORT	Source:	ADV_LLT	ACTION	☐	11/13/00 12:22:22	☐
		Dest:	ADV_LLT	ACTION			
13-Item	MEDIKA_TEST_IMPORT	Source:	ADV_LLT	AEVDUR	☐	11/13/00 12:22:22	☐
		Dest:	ADV_LLT	AEVDUR			
13-Item	MEDIKA_TEST_IMPORT	Source:	ADV_LLT	AEVSTART	☐	11/13/00 12:22:22	☐
		Dest:	ADV_LLT	AEVSTART			
13-Item	MEDIKA_TEST_IMPORT	Source:	ADV_LLT	AEVSTOP	☐	11/13/00 12:22:22	☐
		Dest:	ADV_LLT	AEVSTOP			
13-Item	MEDIKA_TEST_IMPORT	Source:	ADV_LLT	AEYESNO	☐	11/13/00 12:22:22	☐
		Dest:	ADV_LLT	AEYESNO			
13-Item	MEDIKA_TEST_IMPORT	Source:	ADV_LLT	ALGORITHM	☐	11/13/00 12:22:22	☐
		Dest:	ADV_LLT	ALGORITHM			
13-Item	MEDIKA_TEST_IMPORT	Source:	DRGCMP	BOTNODIS	☐	11/13/00 12:23:19	☐
		Dest:	DRGCMP	BOTNODIS			

The Connected Object Report provides the following information:

Report column:	Description:
Object Type	The type of connected object. For example, item.
Source Protocol	The name of the protocol from which the object was copied.
Container Name	If there are container objects, the names of the source and destination container objects.
Object Name	The name of the connected object.

Report column:	Description:
Source Changed	Check box — If selected, the source object has been changed since the connected object was copied or refreshed.
Date Copied or Refreshed	The date that a change in the source object was accepted for the connected object. Otherwise, the date that the connected object was copied from the source object.
Connection Required	Check box — If checked, indicates whether the connection is protected against being deleted, because of the setting of a protocol parameter.

Index

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Clintrial 4.7.1
Admin and Design
Part Number: E27479-01