CLINTRIAL™

Enter, Resolve, and Retrieve

release 4.7.1

Part Number: E27575-01

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.

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 25.

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

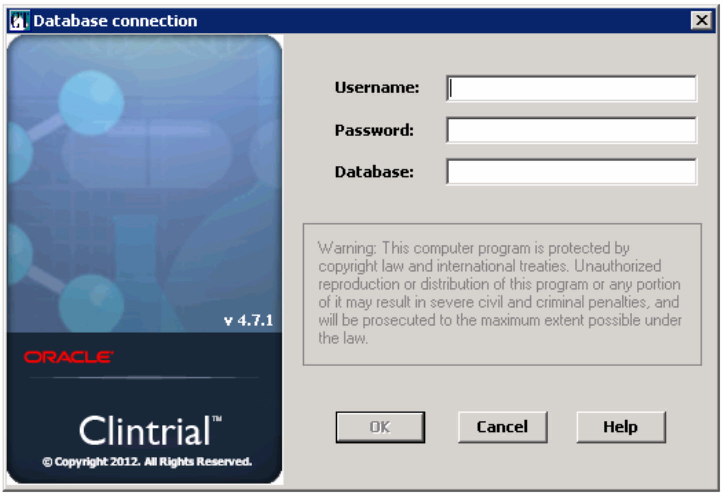
Use the Resolve extended module to manage discrepancy and resolution capabilities that support the work you perform in other Clintrial software modules. You can:

- 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 **Oracle Health Sciences**, the Clintrial program group, then the module.

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



Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program or any portion of it may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

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

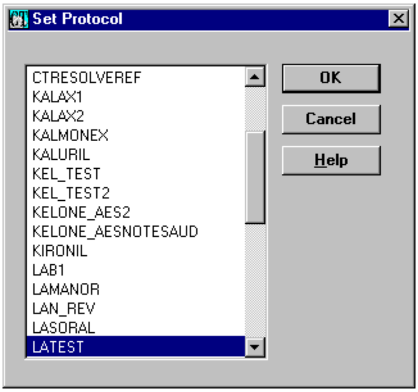
You can start any of the installed Clintrial software modules from the Switchboard **Run** menu. When you start the Switchboard, you provide a user name, password, and database service name. Each time that you start a different Clintrial software module from Switchboard, the Clintrial software uses the 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:



CTRESOLVEREF
KALAX1
KALAX2
KALMONEX
KALURIL
KEL_TEST
KEL_TEST2
KELONE_AES2
KELONE_AESNOTESAUD
KIRONIL
LAB1
LAMANOR
LAN_REV
LASORAL
LATEST

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

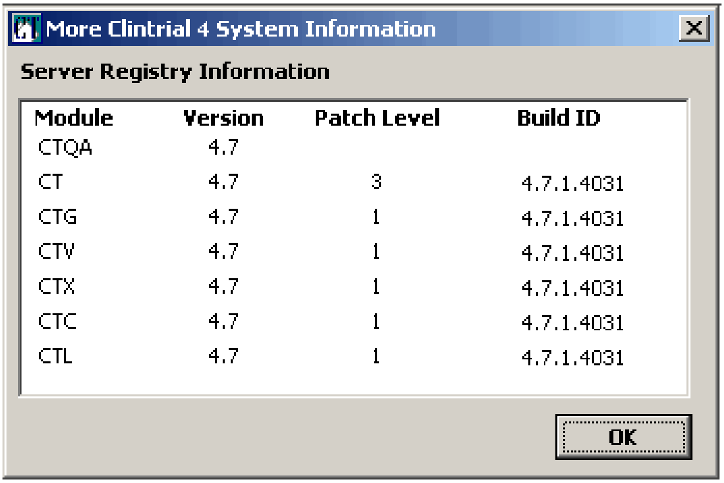
To modify the password for your user account, from the **File** menu, select **Password**. More complete instructions appear in the Clintrial software Basics Help.

How to access the Server Registry Information

You can access the Server Registry 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:


More Clintrial 4 System Information
Server Registry Information
OK

Module	Version	Patch Level	Build ID
CTQA	4.7		
CT	4.7	3	4.7.1.4031
CTG	4.7	1	4.7.1.4031
CTV	4.7	1	4.7.1.4031
CTX	4.7	1	4.7.1.4031
CTC	4.7	1	4.7.1.4031
CTL	4.7	1	4.7.1.4031

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

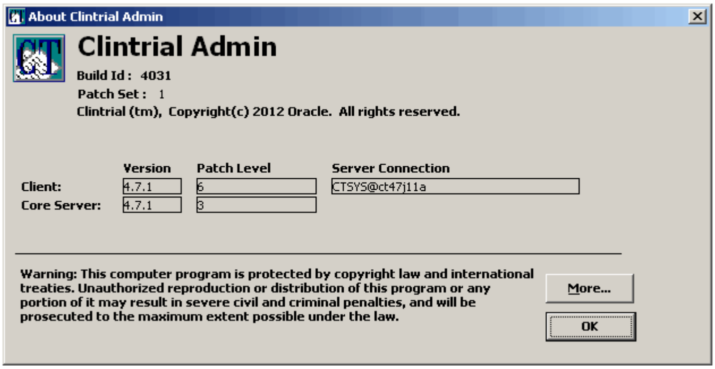
Each Clintrial software module is delivered with Help. Help includes:

- 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:



About Clintrial Admin

Clintrial Admin

Build Id : 4031

Patch Set : 1

Clintrial (tm), Copyright(c) 2012 Oracle. All rights reserved.

	Version	Patch Level	Server Connection
Client:	4.7.1	6	CTSYS@ct47j11a
Core Server:	4.7.1	3	

Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program or any portion of it may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

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**.

Enter, Resolve, and Retrieve

Enter

2 *Introduction to Enter*

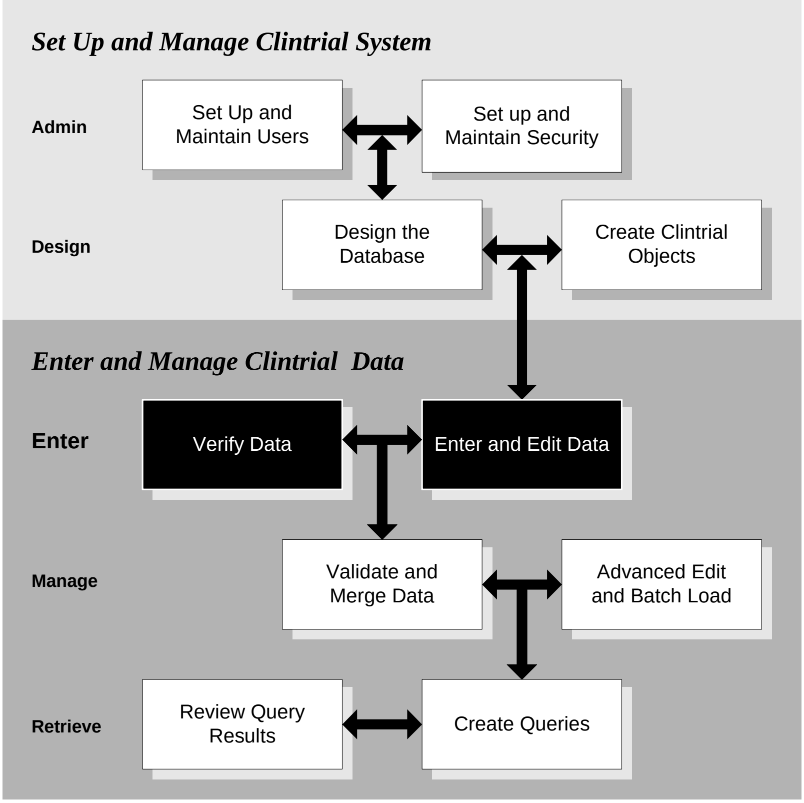
Overview

This chapter introduces you to Enter and describes how to begin an Enter session, the access rights required to use Enter, and the user preferences you can set to customize your Enter environment.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

How Enter fits in the Clintrial software workflow

The following figure shows Enter's place in the Clintrial software workflow:



What are the Enter tasks?

The Clintrial software Enter module is the core module that you use to enter, edit, and verify clinical data. In Enter you can:

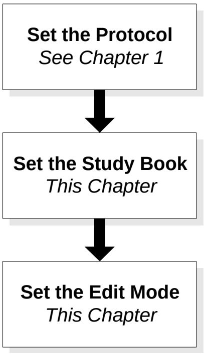
- Enroll subjects in a study.
- Enter and edit clinical data.
- Verify clinical data.
- Create reports about clinical data.

Who are Enter users?

Data-entry operators and data managers are the main users of Enter.

Beginning an Enter session

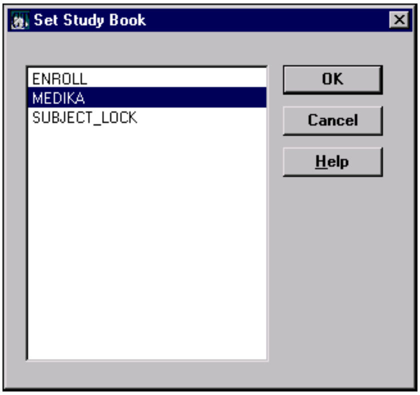
After starting Enter and connecting to the Clintrial software database, you must perform the following tasks to begin your Enter session:



Setting the study book

After setting the protocol, you must set the study book that you want to use in your Enter session. You can only work with one study book at a time.

The first time you start Enter from your workstation, the Set Study Book dialog box opens automatically after you set the protocol. For all subsequent logins, Enter automatically chooses the study book that you used during your last session. Enter displays a message asking you to confirm that this is the study book you want to use. To use a different study book, click **No**. The Set Study Book dialog box opens:



During an Enter session, you can select a different study book to use without closing Enter. To change the study book from within an active session:

1. Close any open study pages.
2. From the **File** menu, select **Set Study Book**. The Set Study Book dialog box opens.

Setting the edit mode

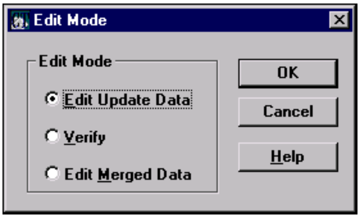
Before you can enter, edit, or verify data, you must set the appropriate edit mode for your Enter session. The edit mode determines the type of data entry that you can perform. There are three edit modes:

- The *Edit Update Data* edit mode allows you to enter new data or edit unmerged data. If you edit unmerged data with the status *Validated* or *Validation Error*, the record is automatically revalidated.
- The *Verify* edit mode allows you to verify previously entered data.
- The *Edit Merged Data* edit mode allows you to edit data that has been validated and moved to the data table. When you edit and save merged data, validation is rerun automatically. If you change data and validation fails, the changes are not saved.

By default, when you start Enter, the edit mode is set to **Edit Update Data**. If you want to perform verification or edit merged data, you must reset the edit mode to either **Verify** or **Edit Merged Data**. The name of the current edit mode is displayed in the study page title bar.

To set the edit mode:

1. Close all open study pages.
2. From the **File** menu, select **Edit Mode**. The Edit Mode dialog box opens:



Edit Mode

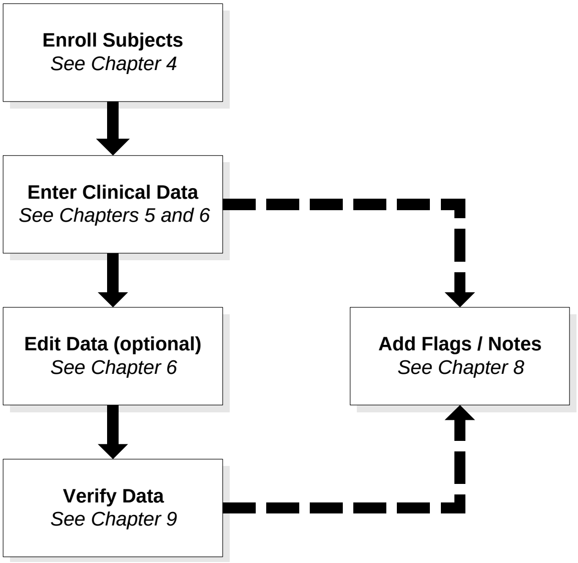
● Edit Update Data
○ Verify
○ Edit Merged Data

3. Select the edit mode you want and click **OK**.

The edit mode you select stays in effect throughout the current session unless you change it.

Sample workflow in Enter

The following figure shows a typical Enter workflow:



Required access rights and access levels

Your ability to enter, modify, or view data in Enter is determined by the *access rights* assigned to you by the Clintrial software administrator. The Clintrial software administrator specifies the level of access you have to each protocol's unmerged, merged, and enrollment data.

The following table shows the available levels of access in Enter for clinical data protocols:

Access right:	Access level:	Use to:
Unmerged	Full	Enter, edit, and delete data in the update tables.
	No Delete	Enter and edit data in the update tables.

Access right:	Access level:	Use to:
	Read	View data in the update tables. ***Note:*** For View protocols, Read is the highest available access level.
	None	No access to data in the update tables.
Merged	Full	Enter, edit, and delete data in the data tables.
	No Delete	Enter and edit data in the data tables.
	Read	View data in the data tables. ***Note*****:** For View protocols, Read is the highest available access level.
	None	No access to data in the data tables.
Enroll	Full	Enroll subjects in a study.
	Read	View enrollment data.
	None	No access to enrollment data.

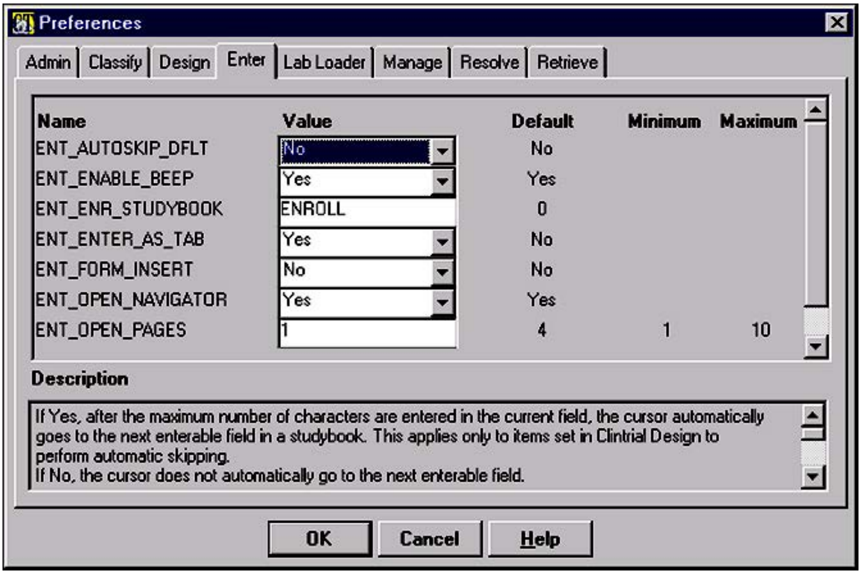
Note: If you open Enter and cannot access a particular study page, it may be because the designer has modified a section on that page but failed to update the section’s status to *Valid*. Contact your designer or Clintrial software administrator.

Setting your user preferences

User preferences are parameters that you can use to tailor your Clintrial software working environment. Default values for user preferences are set automatically when the Clintrial software is installed and can later be added to or modified by your Clintrial software administrator. You can set your own preferences to override these defaults.

To set your Enter user preferences:

1. From the **File** menu, select **Preferences**. The Preferences dialog box opens:



Name	Value	Default	Minimum	Maximum
ENT_AUTOSKIP_DFLT	No	No		
ENT_ENABLE_BEEP	Yes	Yes		
ENT_ENR_STUDYBOOK	ENROLL	0		
ENT_ENTER_AS_TAB	Yes	No		
ENT_FORM_INSERT	No	No		
ENT_OPEN_NAVIGATOR	Yes	Yes		
ENT_OPEN_PAGES	1	4	1	10

Description

If Yes, after the maximum number of characters are entered in the current field, the cursor automatically goes to the next enterable field in a studybook. This applies only to items set in Clintrial Design to perform automatic skipping.
If No, the cursor does not automatically go to the next enterable field.

2. On the Enter tab, enter your preferences:
   - For fields with drop-down lists, select a value from the list.
   - For text-entry fields, be sure that you use the correct data type and syntax. The Clintrial software does not validate your entry except for minimum/maximum range.
   - For fields with a minimum/maximum range, your entry must fall within the specified range.
3. Click **OK**.

Note: Changes to preferences do not affect active Clintrial software sessions. You must restart the module for new preferences to take effect.

The following table describes the user preferences that are available in Enter.

Preference:	**Description:**
ENT_AUTOSKIP_DFLT Default: No	Determines whether autoskip is enabled when you start Enter: Yes: Autoskip is enabled. No: Autoskip is not enabled.

Preference:	**Description:**
ENT_ENABLE_BEEP Default: No	Determines whether an audible sound is heard when a data entry or verify error occurs: Yes: Dialog box and sound occur. No: Only dialog box is displayed.
ENT_ENR_STUDYBOOK Default: None	Displays value of current enrollment study book. Determines the enrollment study book opened when you select the **File** menu’s **New Subject** command.
ENT_ENTER_AS_TAB Default: No	Determines whether the **Enter** key acts the same as the **Tab** key: Yes: The **Enter** key acts as a **Tab** key. No: The **Enter** key is ignored except in the last field on a study page where it moves the cursor to the field on the next study page. ***Note:*** When using Clintrial in Japanese, this preference should be set to **No** to use the **Enter** key for navigating from one field to the next if there are study pages which have a multiline field at the bottom of the page. In Japanese, **Control** + **Enter** key in the last field of a page will always trigger a message asking if you want to save the entries on the current page and will then navigate to the first field on the next page.
ENT_FORM_INSERT Default: No	Determines whether data is saved automatically when you close a study page: Yes: Data is saved automatically. No: Data is saved only on user request.
ENT_OPEN_NAVIGATOR Default: Yes	Determines whether the Navigator opens automatically when you open a clinical data study book: Yes: The Navigator opens automatically. No: The Navigator does not open automatically.

Preference:	**Description:**
ENT_OPEN_PAGES Default: 4	Determines the number of study pages that you can have open at one time in Enter and Retrieve. Minimum: 1 Maximum: 10
ENT_TAB_PAGE_FORWARD Default: No	Determines what happens when you press the **Tab** key when the cursor is in the last field on a study page: Yes: The cursor advances to the first field on the next study page. No: The cursor advances to the first field on the current study page.

3 How Enter Collects and Stores Clinical Data

Overview

This chapter describes the basic objects and structures the Clintrial software uses to collect and store data for a protocol.

How Enter collects clinical data

Clinical data is data collected during a clinical trial. In Enter, you enter clinical data in study books.

What is a study book?

Study books are online representations of case report forms (CRFs). Each study book contains an ordered lists of *study pages* that correspond to the pages in a CRF. When you open a study book, you gain access to its study pages, which you can then work on and navigate as a set.

In addition to study books used to record clinical data, Enter uses two other types of study books:

- *Enrollment study books* are study books used to enter subject enrollment data. Unlike clinical data study books, enrollment study books contain only one study page.
- *Non-subject study books* are study books used to enter nonclinical data; that is, data not related to a particular subject, such as standard coding thesauruses or laboratory normal ranges. Like enrollment study books, non-subject study books contain only one study page.

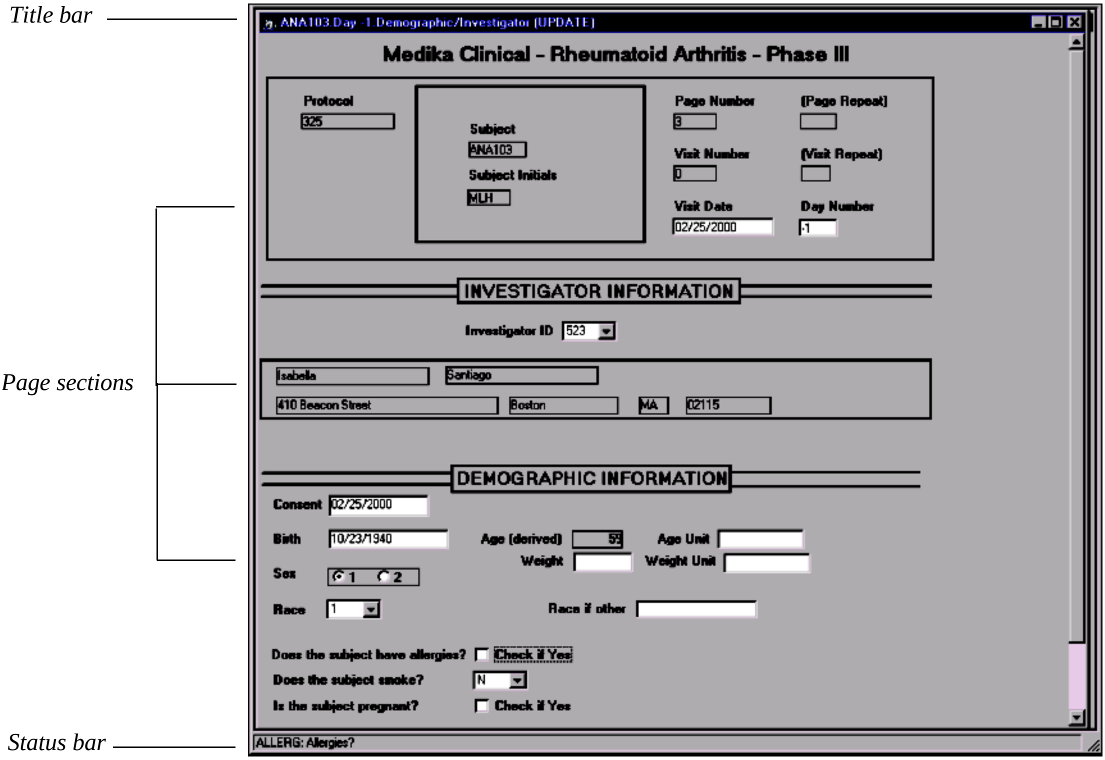
What is a page section?

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.

Study pages in clinical data study books always contain a context page section and one or more page sections for clinical data. Typically, page sections that collect clinical data have a title identifying the type of information they collect, for example, “Demographic Information”.

Study pages in enrollment and non-subject study books contain only one page section. Consequently, page sections in these types of study books are often not labeled.

The following figure shows the parts of a typical study page:



ANA103 Day -1 Demographic/Investigator (UPDATE)

Medika Clinical - Rheumatoid Arthritis - Phase III

Protocol: 325

Subject: ANA103

Subject Initials: MLH

Page Number: 3

(Page Repeat):

Visit Number: 0

(Visit Repeat):

Visit Date: 02/25/2000

Day Number: -1

INVESTIGATOR INFORMATION

Investigator ID: 523

Isabella | Santiago

410 Beacon Street | Boston | MA | 02115

DEMOGRAPHIC INFORMATION

Consent: 02/25/2000

Birth: 10/23/1940

Age (derived): 59

Age Unit:

Weight:

Weight Unit:

Sex: ● 1 ○ 2

Race: 1

Race if other:

Does the subject have allergies? ☐ Check if Yes

Does the subject smoke? N

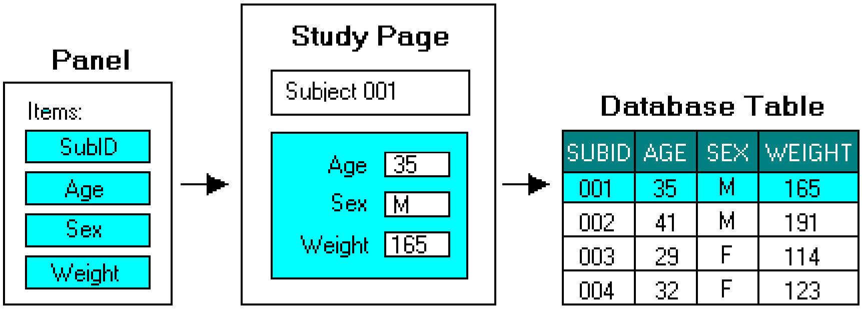
Is the subject pregnant? ☐ Check if Yes

ALLERG: Allergies?

What is a panel?

A *panel* is a collection of logically or clinically related items. It serves as the basis for a page section on a study page. For example, the designer might create a panel to collect demographic information for study subjects. When a panel is installed, three database tables are created for storing its clinical data. See page 45.

The following figure shows the relationship between a panel, its corresponding section on a study page, and its database table:



SUBID	AGE	SEX	WEIGHT
001	35	M	165
002	41	M	191
003	29	F	114
004	32	F	123

What is an item?

Within each panel, the designer creates *items*. An item is an object that stores a particular piece of clinical data, such as a subject's age, sex, or weight. Items serve as the basis for the fields that appear on a study page.

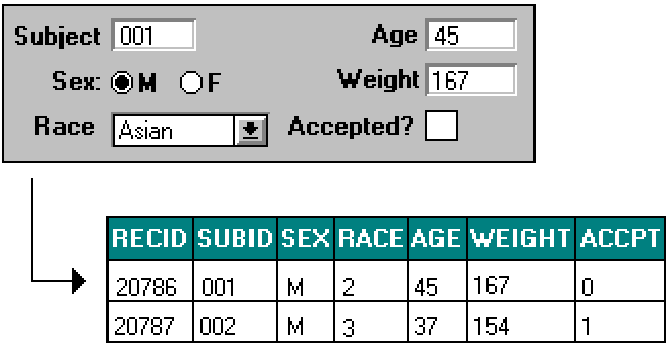
In addition to items that store clinical data, the Clintrial software also uses context items and system items:

- *Context items* are items in a context panel that provide contextual information for records in clinical data tables. Special context items (for example, Subject ID) are used to uniquely identify records.
- *System items* are items internal to the Clintrial software that are attached automatically to every record; for example, record status.

Item names and field labels

Sometimes a field's label on a study page is identical to the field's underlying item name; however, sometimes it is not. When performing certain Clintrial software activities (for example, creating an SQL restriction or specifying retrieval criteria for a report), you must specify actual item names, not field labels. This requires that you know the names of the underlying Clintrial software objects used in your study. You can use Enter's Item Help to reference a field's underlying item name.

The following figure shows how the labels on fields can vary from the fields' corresponding item names:



Subject 001 Age 45

Sex: ● M ○ F Weight 167

Race Asian Accepted? □

RECID	SUBID	SEX	RACE	AGE	WEIGHT	ACCPT
20786	001	M	2	45	167	0
20787	002	M	3	37	154	1

Similarly, certain activities may require you to supply a panel or database table name. Consult your data manager or designer if you need help.

How Enter stores clinical data

When the designer installs a panel, the Clintrial software automatically creates Oracle database tables to store the data it collects. Database tables are organized into rows and columns.

- A column is created in the table for each item in the panel.
- Each row of data stored in the table represents a single *clinical data record*.

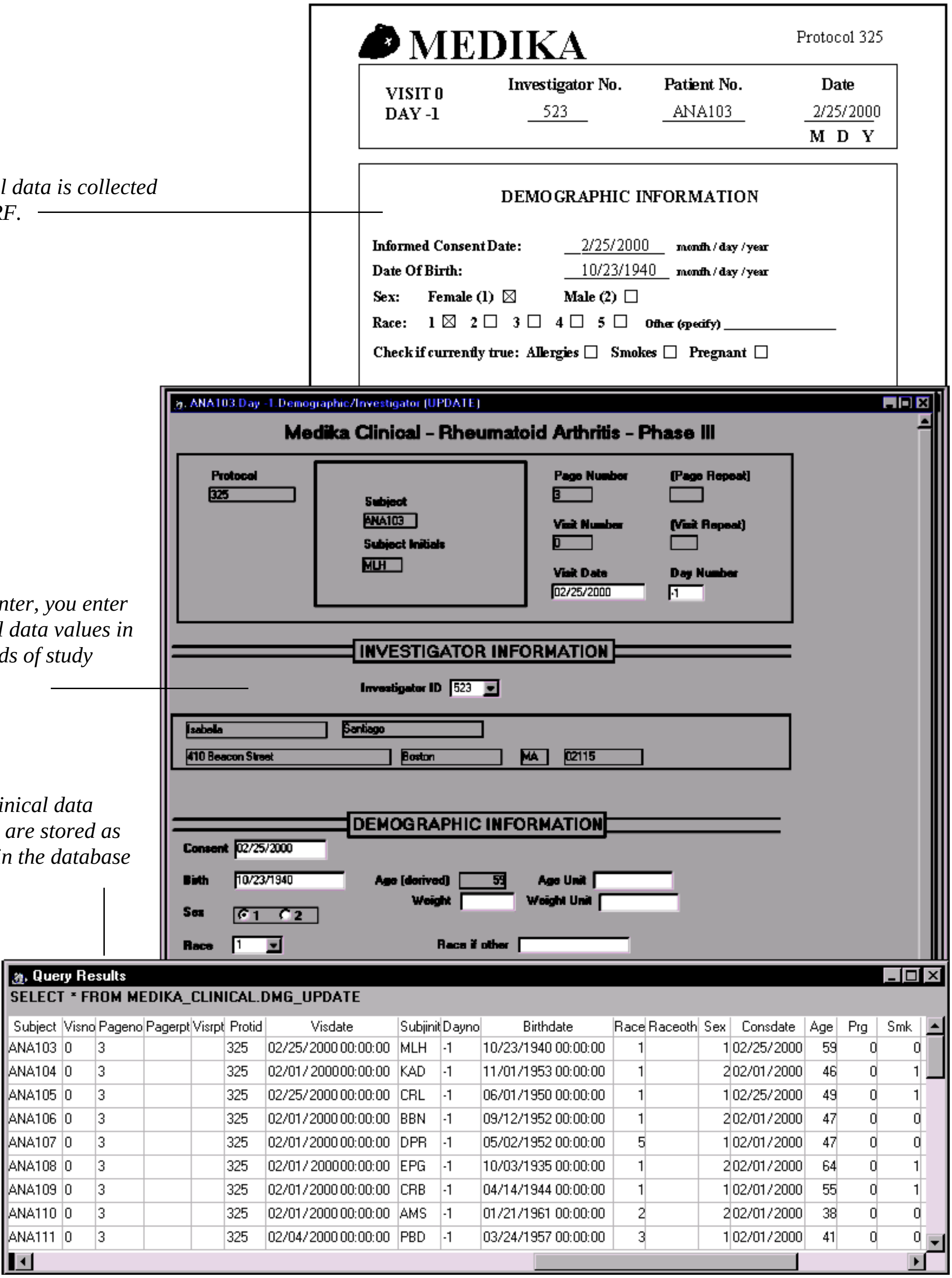
When you enter and save values in a particular page section on a study page, you are creating a record in a database table.

The following figure shows the relationship between a CRF, a study page, and a database table:

Clinical data is collected in a CRF.

With Enter, you enter clinical data values in the fields of study pages.

The clinical data values are stored as items in the database tables.



MEDIKA

Protocol 325

VISIT 0
DAY -1

Investigator No.	Patient No.	Date
523	ANA103	2/25/2000 M D Y

DEMOGRAPHIC INFORMATION

Informed Consent Date: 2/25/2000 month / day / year

Date Of Birth: 10/23/1940 month / day / year

Sex: Female (1) ☒ Male (2) □

Race: 1 ☒ 2 □ 3 □ 4 □ 5 □ Other (specify) ______________

Check if currently true: Allergies □ Smokes □ Pregnant □

ANA103.Day -1.Demographic/Investigator (UPDATE)

Medika Clinical – Rheumatoid Arthritis – Phase III

Protocol: 325

Subject: ANA103

Subject Initials: MLH

Page Number: 3

(Page Repeat):

Visit Number: 0

(Visit Repeat):

Visit Date: 02/25/2000

Day Number: -1

INVESTIGATOR INFORMATION

Investigator ID: 523

Isabella | Santiago
410 Beacon Street | Boston | MA | 02115

DEMOGRAPHIC INFORMATION

Consent: 02/25/2000

Birth: 10/23/1940

Age (derived): 59

Age Unit:

Weight:

Weight Unit:

Sex: ● 1 ○ 2

Race: 1

Race if other:

Query Results

SELECT * FROM MEDIKA_CLINICAL.DMG_UPDATE

Subject	Visno	Pageno	Pagerpt	Visrpt	Protid	Visdate	Subjinit	Dayno	Birthdate	Race	Raceoth	Sex	Consdate	Age	Prg	Smk
ANA103	0	3			325	02/25/2000 00:00:00	MLH	-1	10/23/1940 00:00:00	1		1	02/25/2000	59	0	0
ANA104	0	3			325	02/01/2000 00:00:00	KAD	-1	11/01/1953 00:00:00	1		2	02/01/2000	46	0	1
ANA105	0	3			325	02/25/2000 00:00:00	CRL	-1	06/01/1950 00:00:00	1		1	02/25/2000	49	0	1
ANA106	0	3			325	02/01/2000 00:00:00	BBN	-1	09/12/1952 00:00:00	1		2	02/01/2000	47	0	0
ANA107	0	3			325	02/01/2000 00:00:00	DPR	-1	05/02/1952 00:00:00	5		1	02/01/2000	47	0	0
ANA108	0	3			325	02/01/2000 00:00:00	EPG	-1	10/03/1935 00:00:00	1		2	02/01/2000	64	0	1
ANA109	0	3			325	02/01/2000 00:00:00	CRB	-1	04/14/1944 00:00:00	1		1	02/01/2000	55	0	1
ANA110	0	3			325	02/01/2000 00:00:00	AMS	-1	01/21/1961 00:00:00	2		2	02/01/2000	38	0	0
ANA111	0	3			325	02/04/2000 00:00:00	PBD	-1	03/24/1957 00:00:00	3		1	02/01/2000	41	0	0

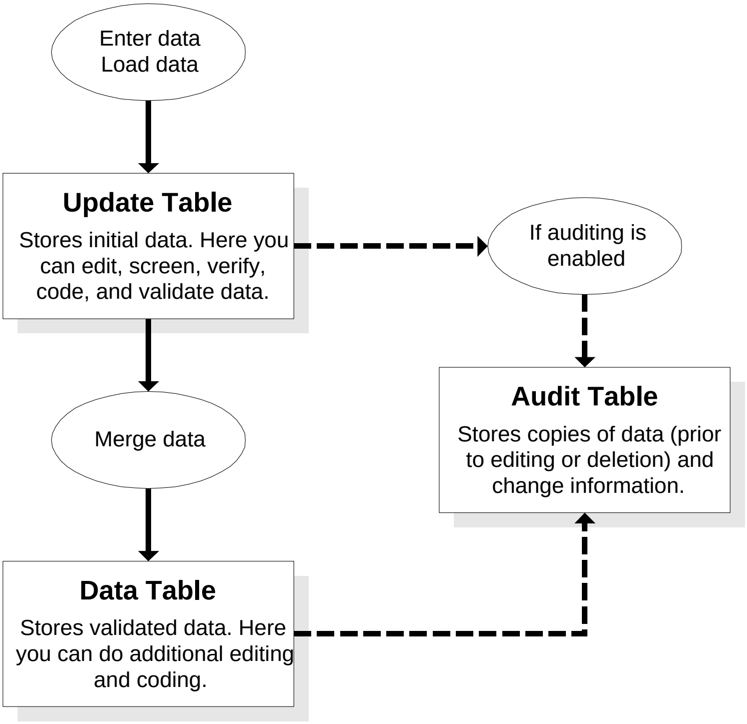
Types of Clintrial software database tables

For each panel your designer installs, the Clintrial software creates the following three database tables:

- The *update table* stores clinical data when it is first entered in the Clintrial software.
- The *data table* stores data that has passed validation and has been merged, or moved, from the update table.
- The *audit table* stores copies of clinical data records as they existed prior to modification or deletion.

For example, if the designer creates a panel named VITALS to collect data pertaining to subjects' vital signs, the Clintrial software creates three database tables for the panel named VITALS_UPDATE, VITALS_DATA, and VITALS_AUDIT.

The following figure shows the data flow between the update table, data table, and audit table:


Enter data
Load data
Update Table
Stores initial data. Here you can edit, screen, verify, code, and validate data.
If auditing is enabled
Merge data
Audit Table
Stores copies of data (prior to editing or deletion) and change information.
Data Table
Stores validated data. Here you can do additional editing and coding.

What is record status?

Record status is a status that the Clintrial software assigns to records to track them internally as they go through various stages of data management. The Clintrial software updates record status automatically. For example, when a record is first entered, its status is 2 (*Unverified*). When the record is verified, its status changes automatically to 1 (*Verified*).

While a record is in the update table, its status can change several times. However, once the record is in the data table, its status cannot change.

The following table lists the statuses used for records in the update and data tables. Only records with the status *Validated* (0) appear in the data table:

Status:	**Assigned to records that have:**	**Table:**	**Code:**
Unscreened	Been entered by batch loading in Manage but have not yet been screened.	Update	3
Unverified	Been entered interactively in Enter but have not yet been verified.	Update	2
Verified	Passed verification or screening.	Update	1
Validated	Passed validation.	Update or data	0
Validation Error	Failed validation or merging.	Update	– 1
Verification Error	Failed verification.	Update	– 2
Screening Error	Failed screening.	Update	– 3

4 Enrolling Subjects in a Study

Overview

This chapter describes how to enroll subjects in a study.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

What is enrollment?

Before you can enter clinical data for a subject, you must add the subject to your study. Adding new subjects to a study is called *enrollment*.

There are two ways to enroll new subjects:

- In a clinical data study book
- In an enrollment study book

Access rights for enrollment

To enter or view enrollment data, you must have the necessary level of access for the Enroll access right. For each protocol on which you are working, your Clintrial software administrator specifies the level of access you will have to that protocol's enrollment data.

The following table lists the different access levels for the Enroll access right:

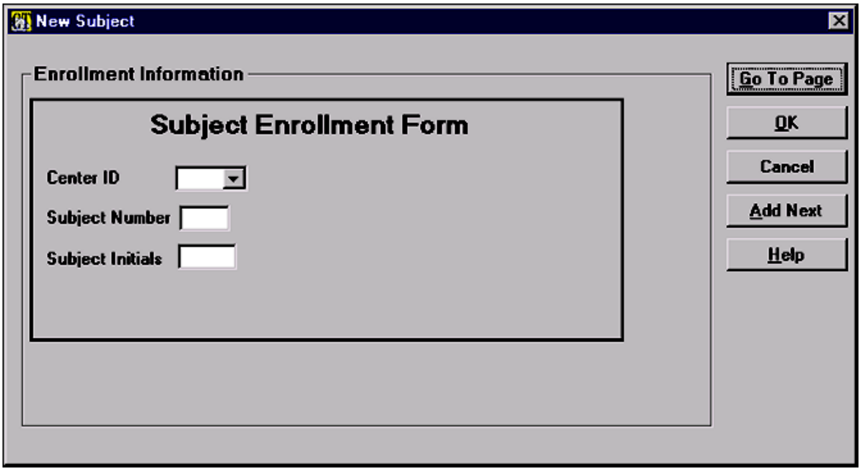
If your access level is:	**You can:**
Full	Enroll subjects.
Read	Open and view enrollment study books.
None	You cannot enroll subjects or open enrollment study books.

Enrolling subjects in a clinical data study book

You can enroll new subjects in a clinical data study book. This method of enrollment is effective when you are only adding a few subjects at a time. When you need to add many subjects, use the enrollment study book.

To enroll a new subject in a clinical data study book:

1. From the **File** menu, select **New Subject**. The New Subject dialog box opens.



   Note: If there is more than one valid enrollment study book for the current protocol, and the ENT_ENR_STUDYBOOK user preference does not specify a valid enrollment study book, the Select Enrollment Study Book dialog box opens. You must first select a valid enrollment study book for the current protocol and session, then the New Subject dialog box opens.

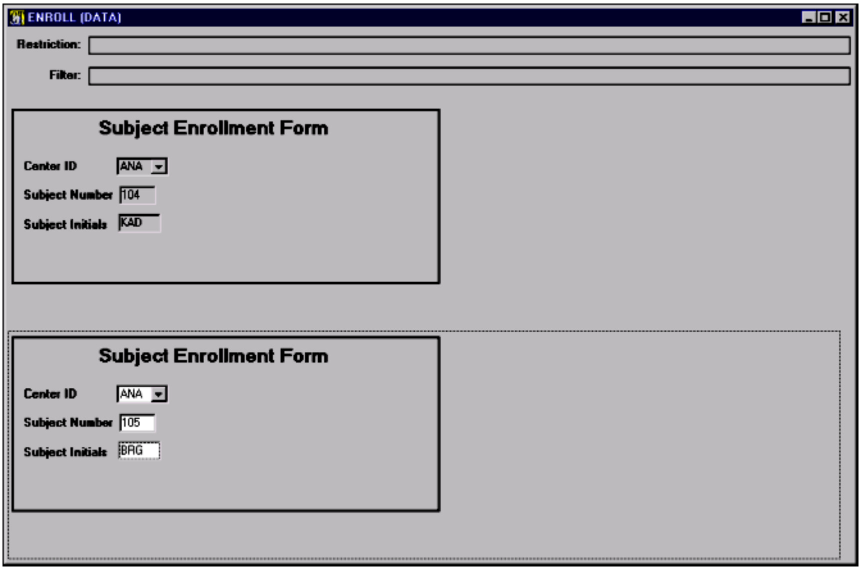
2. Complete the fields in the Enrollment Information section. These fields will vary according to the requirements of your study. There is always a field (for example, Subject Number) that serves as a unique identifier for the new subject.
3. After completing the subject's enrollment information, click:
   - **Go To Page** to save the subject and go to the subject’s first study page.
   - **OK** to save the subject and close the New Subject dialog box.
   - **Add Next** to save the subject and continue adding additional subjects.

Working with enrollment study books

Enrollment study books are study books used exclusively to enroll new subjects. Every clinical data protocol contains at least one enrollment study book. Enrollment study books contain only one study page that opens automatically when you open the study book.

Using an enrollment study book to enroll subjects is effective when you are enrolling many subjects at once, or when you want to view or print a list of enrolled subjects.

The following figure shows a study page from an enrollment study book:



Restriction:

Filter:

Subject Enrollment Form

Center ID: ANA

Subject Number: 104

Subject Initials: KAD

Subject Enrollment Form

Center ID: ANA

Subject Number: 105

Subject Initials: BRG

Enrolling a subject

To enroll a subject in an enrollment study book:

1. From the **Edit** menu, select **Add Subject**.
2. Complete the fields in the study page. These fields will vary according to the requirements of your study. There is always a field (for example, Subject Number) that serves as a unique identifier for the new subject.
3. Repeat these steps to add additional subjects, as needed.
4. Save the study page.

Modifying and deleting a subject

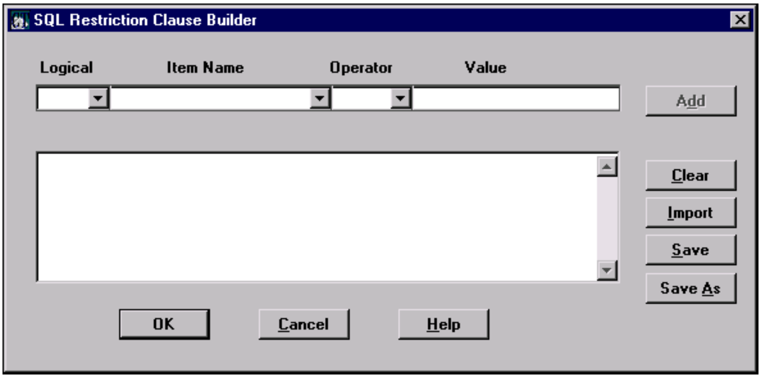
After you save a subject's enrollment data, you cannot modify or delete it in Enter; however, you can remove a subject that has been added but not saved. To remove a subject that has been added but not saved:

1. Place the cursor in a field belonging to the subject you want to remove.
2. From the **Edit** menu, select **Delete Subject**.

Using an SQL restriction

You can use an SQL restriction to specify which subject records the study book should retrieve from the database. For example, you might use an SQL restriction to specify that only female subjects should be retrieved.

To use an SQL restriction, from the **File** menu, select **Respecify**. The SQL Restriction Clause Builder opens:



Note: The SQL Restriction Clause Builder also opens automatically when you first open an enrollment study book.

Filtering and sorting

You can filter and sort subjects in an enrollment study page. Filtering and sorting allow you to determine which subjects are displayed in the study page and the order in which they appear.

- To filter subjects, from the **View** menu, select **Filter**. The Specify Filter dialog box opens.

- To sort subjects, from the **View** menu, select **Sort**. The Specify Sort Columns dialog box opens.

  Note: Filtering does not limit the retrieval of records from the database; filtering only determines which retrieved records are visible in the study page.

Saving data in an enrollment study page

You can save the data in an enrollment study page at any time by selecting **Save** from the **File** menu. If you try to close the study page without saving it, the Clintrial software prompts you to save it.

If the ENT_FORM_INSERT user preference is set to **Yes**, data is saved automatically when you close a study page. No confirmation to save is required on your part. The Clintrial software default for this preference is **No**. To close a study page without saving it, from the **File** menu, select **Close Without Saving**.

Printing an enrollment study page

To print an enrollment study page, from the **File** menu, select **Print**. All records appearing on the study page are printed. Prior to printing, you can use the **Filter** and **Sort** commands to specify which records appear on the page and the order in which they should be listed.

Note: Clintrial prints what can fit on a page, as determined by paper size, page orientation (landscape vs. portrait), etc.

5 Working with Clinical Data Study Books

Overview

This chapter describes the structure of clinical data study books and how to navigate in them.

- For information on entering, editing, and deleting data in a study book, see Chapter 6.
- For information on using enrollment study books, see Chapter 4.
- For information on using non-subject study books, see Chapter 11.

For step-by-step instructions on any tasks described in this chapter, see the Enter Help.

How are clinical data study books organized?

What is a clinical data study book?

In Enter, you record data collected during clinical trials in *clinical data study books*. Study books are online representations of case report forms (CRFs). Each study book contains an ordered list of *study pages* that correspond to the pages in a paper CRF.

What is a block?

A *block* is a grouping 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.

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. Every study page in a clinical data study book contains these page sections:

- A *context page section* that displays basic contextual information such as the subject's ID, the visit number, the date, and so on.

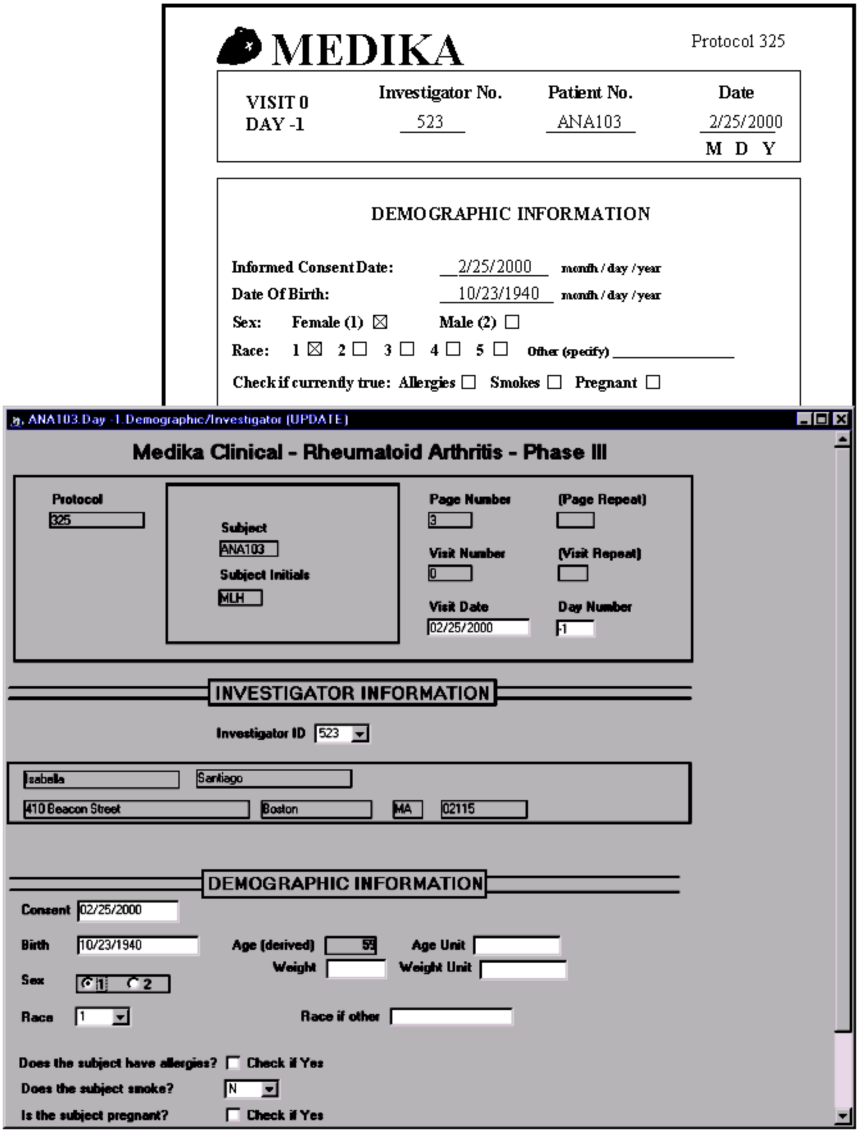
- One or more *clinical data page sections* used to collect a particular type of clinical data, such as a subject's demographic information or vital signs. Typically, page sections that collect clinical data have a title identifying the type of information they collect (for example, "Demographic Information").

Title bar

The title bar on each study page displays the following information:

- The current edit mode, if other than Edit Update Data.
- The ID of the study subject displayed on the study page.
- The name of the block to which the study page belongs.
- The name of the study page.
- The Clintrial software database table type (update or data) in which data entered on the study page is being recorded.

The following figure shows a page from a CRF and its corresponding Clintrial software study page:



MEDIKA

Protocol 325

VISIT 0 DAY -1	Investigator No.	Patient No.	Date
	523	ANA103	2/25/2000 M D Y

DEMOGRAPHIC INFORMATION

Informed Consent Date: 2/25/2000 month / day / year

Date Of Birth: 10/23/1940 month / day / year

Sex: Female (1) ☒ Male (2) □

Race: 1 ☒ 2 □ 3 □ 4 □ 5 □ Other (specify) ____________

Check if currently true: Allergies □ Smokes □ Pregnant □

Title bar

ANA103.Day -1.Demographic/Investigator (UPDATE)

Medika Clinical - Rheumatoid Arthritis - Phase III

Context page section

Protocol: 325

Subject: ANA103

Subject Initials: MLH

Page Number: 3

(Page Repeat):

Visit Number: 0

(Visit Repeat):

Visit Date: 02/25/2000

Day Number: -1

INVESTIGATOR INFORMATION

Investigator ID: 523

Isabella | Santiago

410 Beacon Street | Boston | MA | 02115

Clinical data page section

DEMOGRAPHIC INFORMATION

Consent: 02/25/2000

Birth: 10/23/1940 Age (derived): 59 Age Unit:

Weight: Weight Unit:

Sex: ● 1 ○ 2

Race: 1 Race if other:

Does the subject have allergies? □ Check if Yes

Does the subject smoke? N

Is the subject pregnant? □ Check if Yes

Navigating between study pages and subjects

There are four different methods you can use to navigate between the subjects and study pages in a clinical data study book:

- The Navigator
- Menu commands
- Enter and Tab keys
- The Go To dialog box

The Navigator

The *Navigator* is a browser that contains a hierarchical display of the blocks and pages in a clinical data study book. You can open a study page by clicking its icon in the Navigator. For more information on using the Navigator, see page 59.

Menu commands

The **Navigate** menu contains commands that allow you to move quickly through a sequence of pages and subjects.

- Select **Next Subject** or **Previous Subject** to open a copy of the current page for the next or previous subject.
- Select **Next Page** or **Previous Page** to open the next or previous page for the current subject.
- Depending on your navigation order setting, select **Next** or **Previous** to open either the next or previous page for the current subject, or a copy of the current page for the next or previous subject.
- The **Next** and **Previous** commands work like other **Navigate** menu commands, except when you are navigating to or from the last page of a list. Where the **Next Page**/**Previous Page**/**Next Subject/ Previous Subject** commands move you to the beginning of the list for the current page or subject, the **Next** and **Previous** commands move you to the first page of the next list.

Enter and Tab keys

With the cursor in the last field on a study page, you can open the next page in the navigation order by pressing the **Enter** or **Tab** key.

- If the navigation order is set to **Navigate By Page**, press **Tab** or **Enter** to open the next page for the current subject.
- If the navigation order is set to **Navigate By Subject**, press **Tab** or **Enter** to open a copy of the current page for the next subject.

Note: To use the **Tab** key for study page navigation, your ENT_TAB_PAGE_FORWARD user preference must be set to **Yes**. The default is **No**.

The Go To dialog box

The Go To dialog box contains fields in which you can specify the ID of the page and subject you want to open. To open the Go To dialog box, from the **Navigate** menu, select **Go To**.

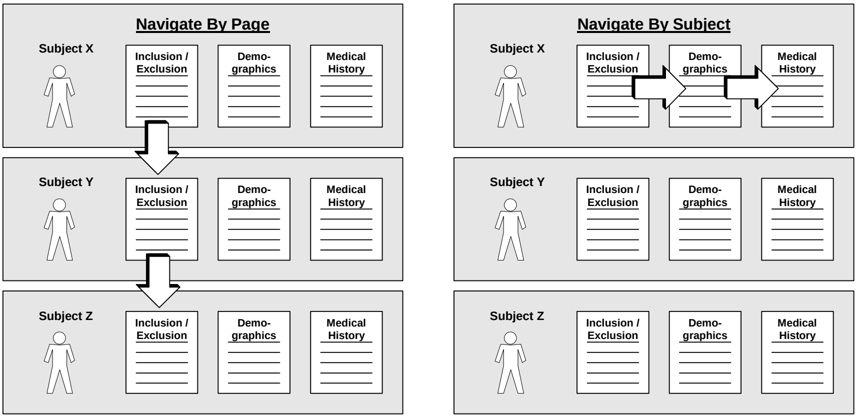
Note: Labels in Go To dialog boxes are truncated if the label is greater than forty characters.

What is navigation order?

Navigation order is a user-modifiable setting that determines what happens when you use **Enter** and **Tab** key navigation, or the **Next** and **Previous** menu commands. When navigation order is set to:

- **Navigate By Subject** (the default), you open a copy of the current study page for the next subject. That is, you navigate from page 1 for subject X, to page 1 for subject Y.
- **Navigate By Page**, you open the next study page for the current subject. That is, you navigate from page 1 for subject X, to page 2 for subject X.

The following figure shows how Navigate By Subject and Navigate By Page work:



To set the navigation order, from the **Navigate** menu, select **Navigate By Subject** or **Navigate By Page**.

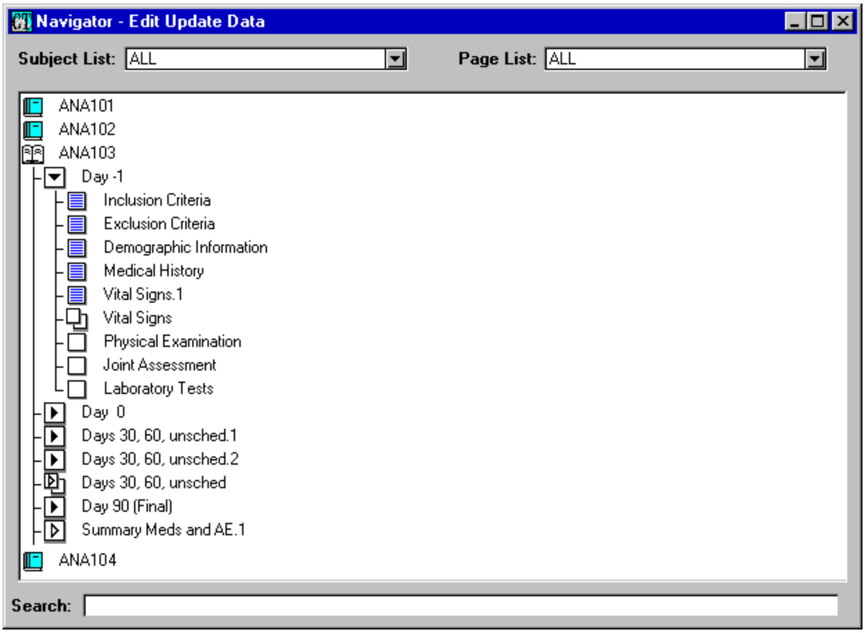
Note: The navigation order also affects how subjects are displayed in the Navigator. (See page 61).

Using the Navigator

The *Navigator* is a browser that contains a graphical display of the blocks and study pages in a clinical data study book. You can open a study page from the Navigator by clicking its icon.

In the Navigator, study pages are arranged in an expandable hierarchy. Depending on your navigation order setting, the Navigator groups pages by subject or by some other definable block (for example, visits). The Navigator uses different icons for subjects, blocks, and pages. See page 61. Shading distinguishes icons containing data from those that do not.

The following figure shows the Navigator:



Navigator - Edit Update Data

Subject List: ALL

Page List: ALL

ANA101
ANA102
ANA103
Day -1
Inclusion Criteria
Exclusion Criteria
Demographic Information
Medical History
Vital Signs.1
Vital Signs
Physical Examination
Joint Assessment
Laboratory Tests
Day 0
Days 30, 60, unsched.1
Days 30, 60, unsched.2
Days 30, 60, unsched
Day 90 (Final)
Summary Meds and AE.1
ANA104

Search:

Opening the Navigator

If your ENT_OPEN_NAVIGATOR user preference is set to **Yes,** the Navigator opens automatically whenever you open a clinical data study book. Otherwise, to open the Navigator, from the **Navigate** menu, select **Navigator**.

Navigator icons

The Navigator uses icons to uniquely identify:

- Study books
- Blocks
- Repeating blocks
- Study pages
- Repeating study pages

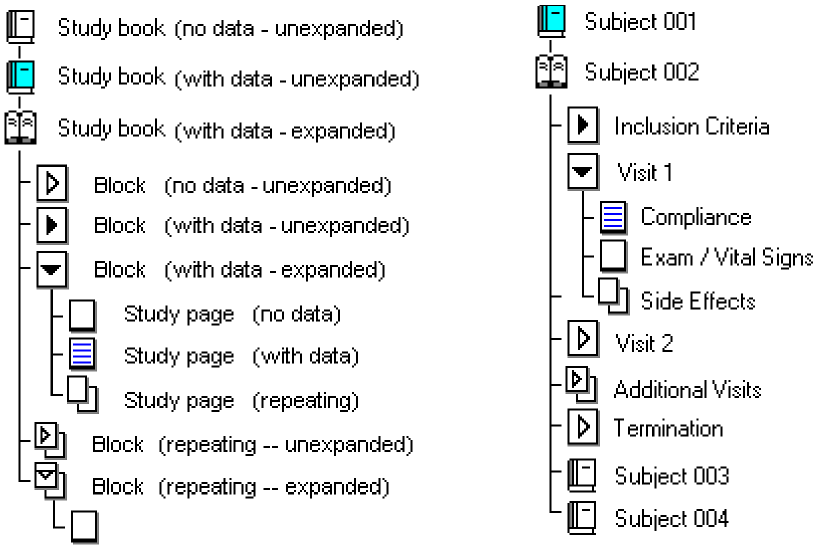
Note: When a repeating block or repeating study page has reached the maximum number of allowable repeats, the repeating icon is replaced by a nonirritating icon. For more information on working with repeating blocks and repeating study pages, see page 63 and page 64.

To expand or collapse a study book or block, double-click its icon. Double-clicking a study page icon opens the study page.

The following figure shows the icons that appear in the Navigator:

Navigator icons in their different display states:

Navigator icons as they appear in a typical study:


Study book (no data - unexpanded)
Study book (with data - unexpanded)
Study book (with data - expanded)
Block (no data - unexpanded)
Block (with data - unexpanded)
Block (with data - expanded)
Study page (no data)
Study page (with data)
Study page (repeating)
Block (repeating -- unexpanded)
Block (repeating -- expanded)
Subject 001
Subject 002
Inclusion Criteria
Visit 1
Compliance
Exam / Vital Signs
Side Effects
Visit 2
Additional Visits
Termination
Subject 003
Subject 004

Note: If your site is a Multisite replication site, a icon will appear next to subjects that were enrolled at other sites. You can view but not edit data belonging to these subjects.

How navigation order affects the Navigator display

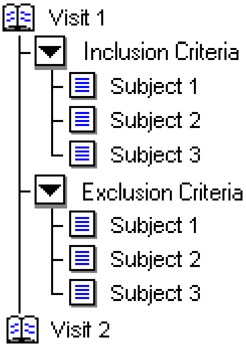
The current navigation order affects how pages are grouped and displayed in the Navigator. If the navigation order is set to:

- **Navigate By Subject**, the Navigator groups subjects under individual pages.

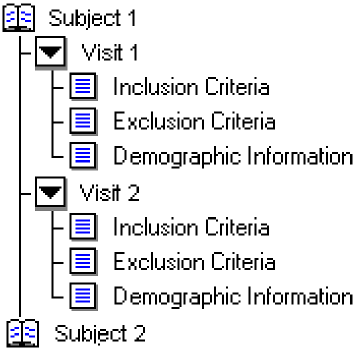
- **Navigate By Page**, the Navigator groups individual pages under each subject.

The following figure shows how navigation order affects the Navigator display:

Navigate By Subject:



Navigate By Page:



Sorting subjects in the Navigator

Subjects can be listed in the Navigator in either ascending or descending order. The default is ascending. To change the order, from the **Navigate** menu, select **Sort Subjects** >> **Ascending** or **Sort Subjects** >> **Descending**.

Using subject lists and page lists

You can use the Subject or Page drop-down lists to specify a particular list of subjects or pages for the Navigator to display. When you select a subject list or page list, only those subjects or pages belonging to the list you select will be displayed and used in the navigation order. The default for both is ALL.

For more information on working with subject lists and page lists, see Chapter 7.

Using the Navigator’s Search field

You can search for a particular block or study page by entering text in the Navigator's Search field at the bottom of the Navigator browser. (See the figure on page 60.) The search begins at the currently selected Navigator icon and continues within the hierarchy until the first match is found. You can only search expanded icons.

For example, to find the next occurrence of the Laboratory Examinations study page in Visit 1, expand Visit 1 and then enter “L” in the Search field. The search begins as soon as you enter the text.

Working with repeating blocks

A *repeating block* is a single block that you can use multiple times.

For example, a study book may contain a block called EXAM that contains study pages to record data on subject examinations. Anticipating that different subjects will need different numbers of exams, the designer could make EXAM a repeating block. You can then record results of subsequent examinations as needed in repeats of the EXAM block.

Your designer can limit the number of times a block can repeat. When this limit is reached, you cannot create additional blocks.

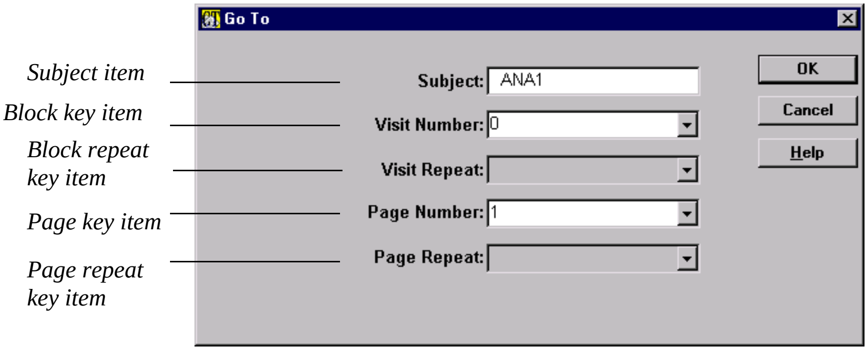
Note: The study pages that appear in a repeating block are determined by the designer. You cannot include or exclude individual pages from the block.

Using a repeating block

Your designer may have renamed the fields shown in the following figure. If the fields have been renamed, the field labels (but not their positions) will differ from those listed here.

To use a repeating block:

1. From the **Navigate** menu, select **Go To**. The following example shows the Go To dialog box in the Medika study book:



Subject: ANA1
Visit Number: 0
Visit Repeat:
Page Number: 1
Page Repeat:

2. In the Subject (Subject Item) field, enter the ID of the subject for whom you want to create a new repeating block.
3. In the Visit Number (Block Key Item) field, enter or select the repeating block on which to base the new one. In this example, the blocks are visits. You must select a block that is repeating, otherwise the Block Repeat Key Item (next) field will be unenterable.
4. In the Visit Repeat (Block Repeat Key Item) field, enter or select a value for the new repeating block. If the maximum number of repeats for this block has been reached, this command is unavailable and you cannot create a new repeating block.
5. In the Page Number (Page Key Item) field, enter or select the study page that you want to go to after the block is created.
6. If the page you selected in Step 5 is a repeating study page, in the Page Repeat (Page Repeat Key Item) field, enter or select the repeating study page that you want to go to after the block is created.
7. Click **OK**. The new block opens to the study page that you selected.

Working with repeating study pages

A *repeating study page* is a single study page that you can use multiple times.

For example, a block may include a study page called VITALS that records data on a subject’s vital signs. Anticipating that vital signs will need to be recorded several times during a visit for some subjects, the designer can make VITALS a repeating study page. You can then record results of subsequent readings as needed in repeats of the VITALS study page.

Your designer can limit the number of times a study page can repeat. When this limit is reached, you cannot create additional repeating study pages.

Repeats of a study page all use the same page key value, and are distinguished by page repeat key values. The designer can either predefine page repeat key values for repeats of the study page, or allow these values to be defined during data entry.

Using a repeating study page

Your designer may have renamed the fields listed here. If this has been done, the field labels (but not their position) will differ from those described here.

To use a repeating study page:

1. From the **Navigate** menu, select **Go To**. The Go To dialog box opens.
2. In the Subject (Subject Item) field, enter the ID of the subject for whom you want to create a new repeating page.
3. In the Visit Number (Block Key Item) field, enter or select the block in which the new page is located.
4. If the block you selected in Step 3 is a repeating block, in the Visit Repeat (Block Repeat Key Item) field, enter or select the repeating block in which the new page is located.
5. In the Page Number (Page Key Item) field, enter or select the repeating study page on which to base the new one. You must select a page that is repeating, otherwise the Page Repeat Key (next) field is unavailable.
6. In the Page Repeat (Page Repeat Key Item) field, enter or select a value for the new repeating page. If the maximum number of repeats for this page has been reached, this command is unavailable and you cannot create a new repeating page.
7. Click **OK**. The new study page opens.

What is a new repeating study page?

If your study book contains repeating study pages for which the designer has predefined a set of page repeat key values, you can use the **New Repeating Page** command to open a new repeating study page.

When you select **New Repeating Page**, you are opening an empty repeating study page that uses a predefined page repeat key. If the maximum number of repeating study pages has already been reached, or if your designer has not predefined page repeat keys, you cannot use this command.

To use the **New Repeating Page** command, first open the study page that you want to repeat. Then, from the **File** menu, select **New Repeating Page**.

Printing clinical data study pages

To print a study page while it is open, from the **File** menu, select **Print**.

You can also print study pages from the Navigator without having to open them. Only study pages that contain data are printed. Only study pages and subjects included in the current page list and subject list can be selected for printing.

Note: Clintrial prints what can fit on a page, as determined by paper size, page orientation (landscape vs. portrait), etc.

1. Select the page(s) that you want to print. You can select:
   - One or more individual pages.
   - A particular block.
   - All the pages in the study book by selecting **Select All** from the **Edit** menu.
2. From the **File** menu, select **Print**.

The Navigator provides several options for selecting groups of subjects and pages to print. These options differ depending on the navigation order setting that you are using.

If you are using **Navigate By Page**, select a:

- Subject, to print all study pages for that subject.
- Block, to print all study pages in the block for that subject.
- Study page, to print that study page for that subject.

If you are using **Navigate By Subject**, select a:

- Block, to print all study pages in the block for all subjects.
- Study page, to print that study page for all subjects.
- Subject, to print that study page for that subject.

6 Entering, Editing, and Deleting Data

Overview

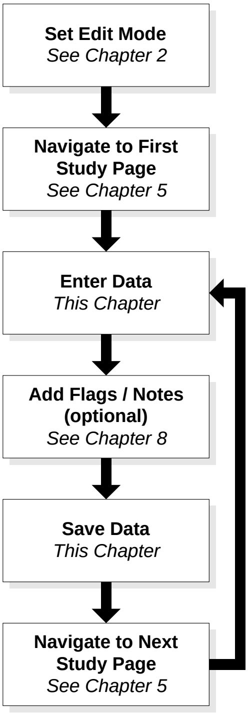
This chapter describes how to enter, edit, and delete data in study books.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

Data-entry workflow

After you set the protocol, study book, and edit mode in which you want to work, you can begin to enter data.

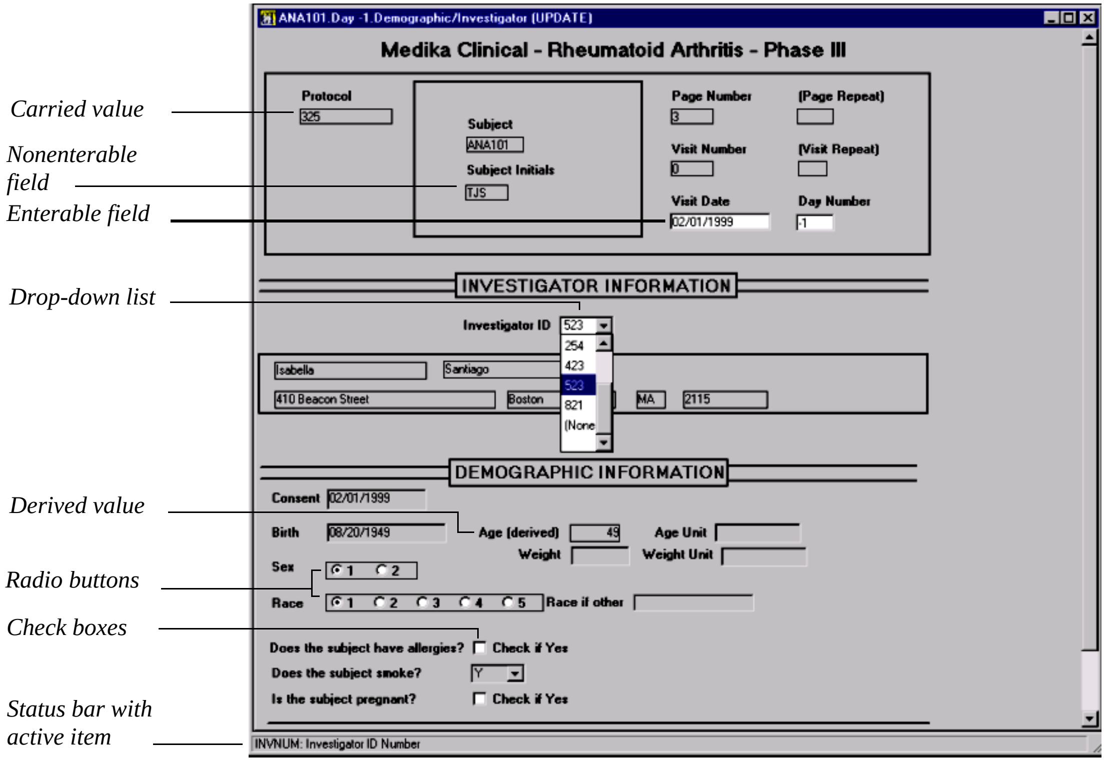
The following figure shows a typical workflow for entering clinical data:



Characteristics of study page fields

A *field* is the data-entry area in which values for an item are entered on a study page.

The following figure shows some of the types of fields that appear on a study page:



ANA101.Day -1.Demographic/Investigator (UPDATE)

Medika Clinical - Rheumatoid Arthritis - Phase III

Protocol 325

Subject ANA101

Subject Initials TJS

Page Number 3

(Page Repeat)

Visit Number 0

(Visit Repeat)

Visit Date 02/01/1999

Day Number -1

INVESTIGATOR INFORMATION

Investigator ID 523

Isabella | Santiago

410 Beacon Street | Boston | MA | 2115

DEMOGRAPHIC INFORMATION

Consent 02/01/1999

Birth 08/20/1949

Age (derived) 49

Age Unit

Weight

Weight Unit

Sex ● 1 ○ 2

Race ● 1 ○ 2 ○ 3 ○ 4 ○ 5

Race if other

Does the subject have allergies? □ Check if Yes

Does the subject smoke? Y

Is the subject pregnant? □ Check if Yes

INVNUM: Investigator ID Number

Enterable and nonenterable fields

Enterable fields are fields in which you can enter data. Enterable fields have a three-dimensional border and a white background.

Nonenterable fields are fields in which you cannot enter data. Non-enterable fields have a colored background (usually gray) and will not accept the cursor.

Some study page fields that are normally enterable may be nonenterable. When you encounter this situation, it could be due to one of the following:

If the field:	Then:
Contains data	• If you are in the Edit Update Data edit mode, the data has already been merged. • If you are in the Edit Merged Data edit mode, the data has not yet been merged. • The protocol has been locked by your designer. ***Note*:** Some items in the context section of a study page always appear nonenterable; for example, page numbers and subject IDs.
Does not contain data	• You may not have access rights to enter or edit data in the current edit mode. • You may not have the proper access rights to work on that page or page section.

Required fields

Your designer can designate specific fields as *required*. You cannot save a study page until all of its required fields contain data. If you try to save a study page when one or more of its required fields do not contain data, the Clintrial software displays a message and highlights the field(s) in question. Fields that are not required are optional; they do not require a value for you to save the study page.

Data types

The *data type* is an item attribute that determines the type of value a field will accept. You can use Item Help to determine a field's data type. See page 76.

The following table describes the data types used in the Clintrial software:

Field type:	Data type:	You can enter:	Example:
Date	DATE	A date.	8/15/2000

Field type:	Data type:	You can enter:	Example:
Date+time	DATETIME	Date and time.	8/15/2000 21:20:20
Numeric	FIXED	An integer that cannot include a decimal point. You can preface the integer with a plus (+) or minus (-) sign.	100
Numeric	FLOAT	A number that can include a decimal point. You can preface the number with a plus or minus sign.	25.6
Text	TEXT	A character string that can include letters, numbers, punctuation marks, spaces, and special characters.	TRN-001

Note: The Clintrial software uses Oracle's database format for the numeric FLOAT data type. In this format, an item whose definition is, for example, NUMBER (4,1) will accept numbers up to four digits, with one of those digits appearing to the right of the decimal point (999.9). While Oracle allows up to 38 digits, the Clintrial software is limited to 18.

Field width

Due to space limitations, fields are sometimes narrower than the actual length of data that they can contain. When your entry exceeds the visible width of a field (is within a field's allowable limit) the cursor scrolls automatically to the right as you type. When your entry reaches the field's allowable limit, you cannot enter additional text. To view the full text of truncated entries, place the cursor in the field, and then scroll using the right arrow key.

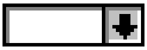
Minimum/maximum value

If a field has lower and upper bounds, the value you enter must be within the specified limits. However, if the designer has enabled the Override feature for the item, you can enter a value that overrides the lower and upper bounds.

Text case

You can enter text values in either uppercase or lowercase. Text is stored in the case in which it was entered, unless your designer has formatted the field to convert entries automatically to uppercase.

Drop-down lists



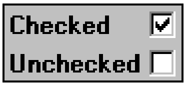
Drop-down lists are fields for which your designer has defined a specific list of acceptable values. You can recognize a drop-down list by the down arrow button on the right side of the field. Clicking the button displays the list.

If your designer set up this item as a codelist, you can only select values that appear on the list; a typed entry is not permitted. You can scroll automatically to any item in a drop-down list by pressing the keyboard key that corresponds to the item’s first letter. For example, pressing the **S** key will take you automatically to the first entry that begins with the letter S. Pressing **S** again will take you to the second **S** entry.

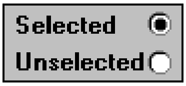
If your designer set up this item as a checklist, you can select an item from the list or you can type in an entry of your own (subject to restrictions like minimum/maximum value). Combo boxes can be distinguished from list boxes by the gap between the end of the field and the arrow button.

Note: In some instances, you may encounter items in a drop-down list that cannot be selected (displayed as red text). These items have been made unavailable by your designer, usually because they have become obsolete.

Check boxes and radio buttons



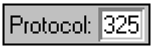
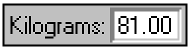
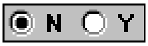
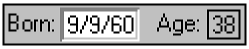
Check boxes can appear either alone or in groups of two or more. When they appear in a group, you can select as many or as few of the check boxes as you want. To select a check box, click it. To deselect (clear) a check box, click it a second time.



Radio buttons always appear in groups of two or more. You can only select one radio button in the group. To select a radio button, click it. To deselect a radio button, click it a second time, or click a different button in the group.

Display fields

Display fields are fields that automatically display a value. This value may be a default, or it may be calculated or carried over from data entered in other fields. Some display fields are enterable; some are not. There are four types of display fields:

Display field type:	**Description:**	**Example:**
Carried	A value carried over from the same field on a previous page. Carried values are used only in context items. They may or may not be editable.	 Protocol: 325 Protocol number was carried from the same field on an earlier page.
Converted	A value converted automatically from the entered value. The converted value is displayed when you exit the field.	 Kilograms: 81.00 The user entered 81 which was converted to a decimal.
Default	A default selection or value for a field. Default values are always editable.	 ● N ○ Y Radio buttons usually display a default value based on the most common response.
Derived	A display-only value calculated from a value entered in a different field. Derived values are not calculated until a record is validated. Prior to validation, they do not display.	 Born: 9/9/60 Age: 38 The Age field displays a derived value calculated from the subject's date of birth.

Displaying dates in fields

The format in which dates display in study page fields is determined by your computer's system date format setting. Windows 98 users can alter this setting using the Windows Regional Settings dialog box.

1. From the Windows **Start** menu, select **Settings >> Control Panel**.
2. In the Control Panel window, select **Regional Settings**. The Regional Settings Properties dialog box opens.
3. Select the **Date** tab.
4. To change your display format, select a format from the Short date style drop-down list.

Note: The long date style setting does not affect the Clintrial software.

Entering dates in fields

The following guidelines apply when you enter dates in study page fields:

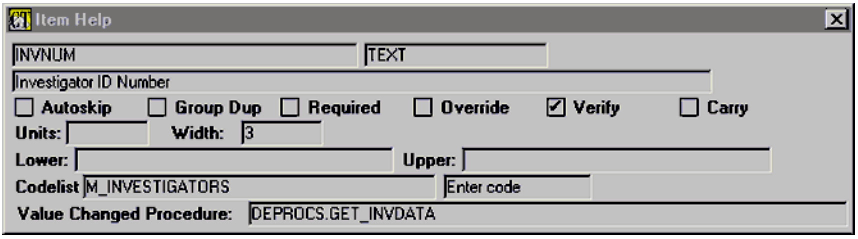
- Years can be entered as two digits or four digits. When a two-digit year is entered, the Clintrial software stores numbers 00 to 49 as 21st century (2049) and numbers 50 to 99 as 20th century (1950).
- The sequence in which day, month, and year must be entered is determined by the sequence used in your computer's system date format setting. Available sequences are day/month/year, month/day/year, and year/month/day. The Windows default is month/day/year.
- When entering days and months, leading zeros are accepted but not required.
- You can enter a month as a numeric value, a three-character abbreviation, or as a complete word.
- Date format examples:
  - Leading zeros: 01-15-00
  - No leading zeros: 1-15-00
  - Month/day/year sequence: JAN-1-2000
  - Day/month/year sequence: 1-JAN-2000
  - Year/month/day sequence: 2000-JAN-1
  - Two digit year: 1-15-00
  - Four digit year: 1-15-2000
  - Three character month: JAN-15-2000
  - Month as word: JANUARY-15-2000

- You can use any of these delimiters to separate day, month, and year values in a date:
  - Slashes (1/15/2000)
  - Hyphens (1-15-2000)
  - Periods (1.15.2000)
  - Commas (1,15,2000)
  - Spaces (1 15 2000)

Using Item Help

You can view the attributes of any enterable field on a study page by using Enter’s Item Help. *Item Help* displays important information about a field, such as the type of value the field will accept, its underlying item name, and whether the field requires verification. To open Item Help:

1. Select the field whose attributes you want to view.
2. From the **Help** menu, select **On Current Item** or press **Shift** + **F1**.
3. The Item Help dialog box opens:



INVNUM

TEXT

Investigator ID Number

□ Autoskip □ Group Dup □ Required □ Override ☑ Verify □ Carry

Units: Width: 3

Lower: Upper:

Codelist M_INVESTIGATORS Enter code

Value Changed Procedure: DEPROCS.GET_INVDATA

Item Help attributes

The following table lists the item attributes that appear in Item Help:

Attribute:	Description:
Item Name	The item’s name as it appears in its database table.
Data Type	The data type for the item.
Description	Description of the item as entered by the designer.

Attribute:	Description:
Autoskip	Indicates whether autoskip is enabled for the item.
Group Dup	Indicates whether group duplication is enabled for the item.
Required	Indicates whether the item is optional or required.
Override	Indicates whether you can override codelist and range checks on data values for the item.
Verify	Indicates whether the item requires verification.
Carry	Indicates whether a value entered for the item will be carried forward on other study pages.
Units	A text description of the item's unit of measurement.
Width	The width (in characters) of the field.
Lower/Upper	Indicates the minimum and maximum value you can enter.
Codelist	Name of the codelist used by the item (for drop-down lists, check boxes, and radio buttons only). Indicates either Enter code or Enter value.
Value Changed Procedure	Name of the data-entry processing procedure used by the item.

Other Field Help

When the cursor is in an enterable field on a study page, the status bar at the bottom of the window displays information about the field. The default field information includes the name and a description of the item. If the designer has specified the Help text item attribute for that field, this customized Help text appears.

Field navigation

This section lists the keyboard commands you can use for field navigation.

Navigating between fields

You can use the following keyboard commands to navigate between fields on a study page.

Command:	**Moves cursor to:**
Tab	Next field.
Shift + Tab	Previous field.
Ctrl + Home	First field in page section.
Ctrl + End	Last field in page section.
Enter	First field on next study page (from last field on current page).
Ctrl + Page Up	First field (in a page section with repeating items).
Ctrl + Page Down	Last field (in a page section with repeating items).
Shift + Enter	Last field on previous study page (from first field on current page).

Navigating within a single field

You can use the following keyboard commands to reposition the cursor and select text within a single field.

Command:	**Description:**
Home	Moves cursor to the beginning of the field.
End	Moves cursor to the end of the field.

Command:	**Description:**
Ctrl + Right Arrow	Moves cursor one word to the right.
Ctrl + Left Arrow	Moves cursor one word to the left.
Escape	Restores the field's original value.
Shift + Home	Selects text from cursor position to the beginning of the field.
Shift + End	Selects text from cursor position to the end of the field.
Shift + Right Arrow	Selects one character to the right.
Shift + Left Arrow	Selects one character to the left.
Ctrl + Shift + Right Arrow	Selects one word to the right.
Ctrl + Shift + Left Arrow	Selects one word to the left.

Using autoskip

Automatic skipping (*autoskip*) is a feature that advances the cursor automatically to the next enterable field when you have entered the maximum number of characters in the current field. Autoskip only affects fields for which the designer has set the Autoskip attribute. Typically, these are fields for which there is a fixed entry length (date fields, for example).

You can use the ENT_AUTOSKIP_DFLT user preference to specify whether you want autoskip to be enabled by default whenever you open a study book.

You can also enable or disable autoskip while working on a particular study book. On the **Edit** menu, check or clear **Autoskip**. When autoskipping is enabled, a check mark (✓) appears next to the menu item.

To confirm whether the designer has set the Autoskip attribute for a specific field, open Item Help for that field.

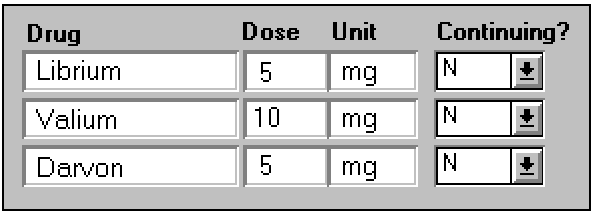
Working with repeating items

Your study may include page sections that contain repeating items. *Repeating items* are items for which multiple values can be entered. Typically, fields based on repeating items are arranged in a row. As necessary, you can insert additional rows to enter multiple values for the items in the page section. Each row of values is then stored as a separate record in the Clintrial software database.

For example, suppose you need to record information on medications being taken by a subject. If the fields used to collect this information are repeating, you can insert extra rows to record information on different medications.

When a page section contains repeating items, all of the items in the section are repeating; you will never encounter a situation in which some of the items in a page section are repeating while others are not repeating.

The following figure shows a page section with repeating items:



Drug	Dose	Unit	Continuing?
Librium	5	mg	N
Valium	10	mg	N
Darvon	5	mg	N

Your designer should provide the visual clues necessary to help you distinguish page sections with repeating items. If you are in doubt, you can determine whether an item is repeating by placing the cursor in the field. Then, look at the **Insert Repeat** command (on the **Edit** menu). If the command is enabled, the item is repeating.

Sort order in repeating items

Your designer can define sort keys for repeating items. When sort keys have been defined, records entered in the page section are sorted when you save the study page according to the criteria the designer has defined.

For example, suppose a set of repeating items collects information on medications a subject is taking. The designer can specify that those items sort alphabetically by medication name.

Note: It is important to be aware of sort order when performing verification: records may not be listed in the same order in which they were originally entered.

Adding and deleting rows

You can use the following commands to add or delete rows of repeating items:

- To add a new row below the last row, from the **Edit** menu, select **Add Repeat**.
- To insert a new row above the current row, from the **Edit** menu, select **Insert Repeat**.
- To delete the row, from the **Edit** menu, select **Delete Repeat**.

Using flags and notes

When adding flags and notes to repeating items, the flag or note level you use will depend on how you want it to be applied:

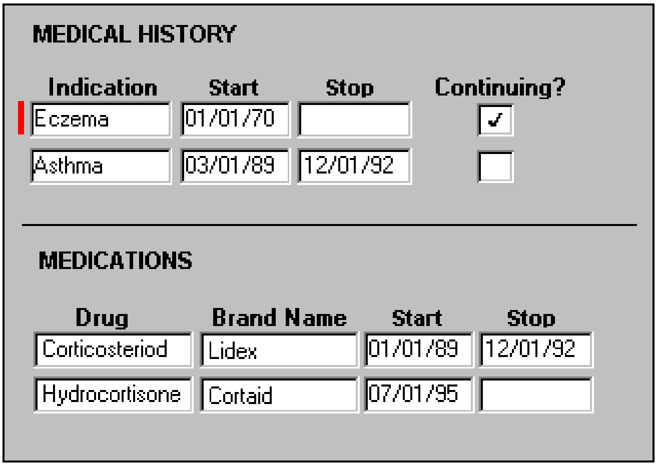
- Combined, the rows of repeating items within a page section make up one observation. Use an observation flag or note when you want the flag or note to apply to the rows as a group.
- Because each row of repeating items constitutes an individual record, use a record flag or note when you want it to apply to a particular row.
- Finally, you can use an item flag or note when you want the flag or note to apply to a single field.

Working with master records and detail records

Your study may include page sections that store master records and detail records. *Master records* and *detail records* are sets of directly related records. Each master record serves as the parent for its detail records; each detail record expands or supplements the data contained in its
master record.

Typically, master records and detail records are entered in two adjoining page sections. The Clintrial software places a red rectangle next to the master record that is currently selected. The detail page section displays only those records that belong to the current master record.

In the following example, the master record Eczema in the Medical History page section is supplemented by the two detail records in the Medications page section. The rectangle next to Eczema identifies it as the current master.



MEDICAL HISTORY

Indication	Start	Stop	Continuing?
Eczema	01/01/70		☑
Asthma	03/01/89	12/01/92	□

MEDICATIONS

Drug	Brand Name	Start	Stop
Corticosteriod	Lidex	01/01/89	12/01/92
Hydrocortisone	Cortaid	07/01/95	

At least one, and possibly both of the page sections in a master-detail relationship will contain repeating items. When working with a group of repeating master or detail records, you can add or remove rows, flags, and notes as you would in any other group of repeating items. When you delete a master record, all of its detail records are automatically deleted.

Master record and detail record navigation

You can use the following keyboard commands to navigate between rows in master records and detail records:

To move:	Use:
Between rows in the master section.	Up or Down arrow keys.
From a master record to its detail record(s).	From the last field in the master row, press **Tab**.
Between rows in the detail section.	From the last field in the current detail row, press **Tab**.

To move:	Use:
From a detail record back to its master record.	From the first field in the first detail row, press **Shift**+**Tab**.

Duplicating data in a study page

Duplication is the process of copying the values entered for an item or group of items in one study page to another.

For example, suppose a study book contains blocks for Visit 1 and Visit 2 and the study page Concomitant Medications appears in both blocks. You can copy (duplicate) the values entered for Concomitant Medications in Visit 1 to the same study page in Visit 2.

Types of duplication

There are two types of duplication:

- *Item duplication* duplicates the value for a single item. You can always duplicate data for individual items.
- *Group duplication* duplicates all items on the page for which your designer has set the Group Dup attribute. You can use Item Help to verify whether or not Group Dup has been set for a particular item. (See page 76.)

Note: When you use item duplication or group duplication for a set of repeating items, all data values are copied.

Restrictions on duplication

The source of duplicated data must always be the same subject, same study page (same repeat), previous block. You cannot skip blocks, nor can you duplicate data from one subject to another.

Duplicating data for a single field

To duplicate data in a single field:

1. Place the cursor in the field where you want to insert duplicate data.
2. From the **Edit** menu, select **Duplicate Item**. The value entered for the same subject, same study page, previous block is copied into the current field.

Note: You can duplicate data for all single items, regardless of whether the Group Dup attribute is set for the item.

Duplicating data for a study page

To duplicate data for a study page:

1. Place the cursor in any field in the study page where you want to insert duplicate data.
2. From the **Edit** menu, select **Duplicate Group**. For all fields on the page set to allow group duplication, the values entered for the same subject, same study page, previous block are copied into the current page.

Note: Only items with the Group Dup attribute setting are included in group duplication. You can use Item Help to determine if Group Dup has been set for a particular item.

Deleting data in a study page

In certain instances it may be necessary to delete all of the data that has been entered and saved on a particular study page. To do this, you must have FULL access rights for both:

- At least one panel on the study page.
- The particular edit mode in which you are working.

To delete all data from a study page:

1. Place the cursor in any field on the study page.
2. From the **Edit** menu, select **Delete Page Data**. A dialog box prompts you to confirm the deletion.

Saving data in a study page

You can save the study page you are working on at any time by selecting **Save** from the **File** menu. If you try to close a study page whose data has not been saved, the Clintrial software prompts you to save before closing the study page. You cannot save a study page until all of its required fields have been completed.

Note: Sometimes when you close a study page, the Clintrial software prompts you to save the new data even if you did not enter any new data. When this occurs, it is because the Clintrial software automatically created certain “hidden” data when the study was opened (typically, context data such as the date, the subject ID, and so on). In this situation, click on **No** when you are prompted to save the data.

Automatic saving

Two Enter user preferences allow you to save study pages automatically. When you use either user preference, no confirmation to save is required on your part.

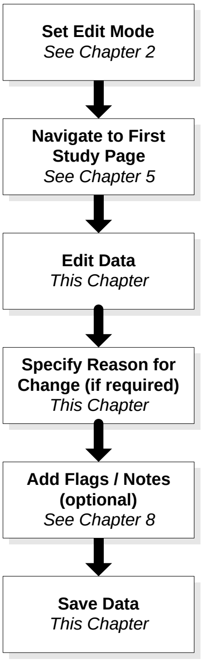
- If the ENT_FORM_INSERT user preference is set to **Yes**, data is saved automatically when you close a study page. The Clintrial software default for this preference is No.
- The ENT_OPEN_PAGES user preference allows you to specify a limit for the number of study pages that can be open at one time. Once that limit is reached, the Clintrial software saves and closes the least recently accessed study page each time you open another study page. The Clintrial software default is 4.

Editing data

Once clinical data is entered into the database, it can be edited, as necessary.

Note: If you have problems accessing data, check with your Clintrial software administrator to ensure that you have the required access rights. (For more information on access rights, see Chapter 2.)

The following figure shows a sample workflow for editing data:



What is unmerged data?

Unmerged data is data stored in the update table; that is, it is data that has not been moved (merged) to the data table. To edit unmerged data, you must set the edit mode to **Edit Update Data**. (See page 32.)

Navigate to the appropriate study book and study page, then edit the data as necessary. The title bar of the study page indicates the study page name and the type of database table that you are editing; for example, 003.Visit1.Vital Signs (UPDATE).

During the editing session, you only have access to records in the update table. If data in the current study page is nonenterable, this indicates that the data has already been merged.

When you edit unmerged data with a status of *Validated* or *Validation Error*, the record is automatically revalidated.

What is merged data?

Merged data is data that has been validated and moved (merged) to the data table. To edit merged data, you must set the edit mode to **Edit Merged Data**. (See page 32.)

Navigate to the appropriate study book and study page, then edit the data as necessary. The title bar of the study page indicates the study page name and the type of database table that you are editing; for example, 003.Visit1.Vital Signs (DATA).

During the editing session, you only have access to records in the data table. If data in the current study page is nonenterable, this indicates that the data has not yet been merged.

When you edit and save data in the data table, validation is rerun automatically. If you changed data and the validation fails, the changes are not saved.

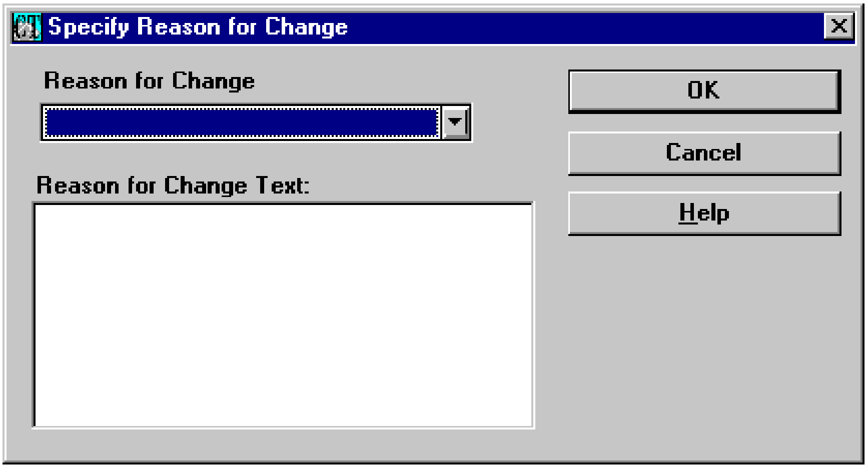
Specifying a Reason for Change

When you modify or delete data, you may be required to provide a Reason for Change. A *Reason for Change* is an explanation appended to an individual record explaining why the record was modified. Reasons for Change are stored in the audit table and included in the Manage Audit Records Report.

A Reason for Change is required when you modify data or delete data and both of the following conditions are true:

- The current protocol requires a reason for change; that is, the AUDIT_REASON_REQD Manage system parameter has been set to **Yes**.
- The audit start point for the protocol has been reached. For example, if auditing is set to begin at merging, only modifications to merged data require a Reason for Change.

The Specify Reason for Change dialog box opens automatically when you modify a record under either of the aforementioned conditions. If your protocol does not require a Reason for Change, you can still specify one by selecting **Specify Reason for Change** from the **Edit** menu.



1. Select a reason from the Reason for Change drop-down list, or enter an explanation directly into the Reason for Change Text box. The designer determines the options that appear in the Reason for Change drop-down list by adding values to the CTS_REASON_CODES codelist.

   Note: If you are planning to use Clintrial Resolve to Edit Source Data, the Reason for Change must be limited to 20 characters.

2. Click **OK**.

Note: If a Reason for Change has been specified for one record on a study page, but not for other records that also require them, when you save the study page, the Clintrial software allows you to apply that Reason for Change to those other records.

Working with scanned CRFs and DCFs

The Clintrial software supports on-screen access to scanned copies of paper Case Report Forms (CRFs) and Data Correction Forms (DCFs) that reside in third-party imaging systems. To display a scanned copy of the corresponding CRF or DCF, you can use keyboard commands, called *hot keys*, while you enter or edit patient data on a study page or view a Resolve discrepancy detail. You can also use hot keys to display the study page corresponding to the current CRF.

Using Hot keys

The Clintrial software provides hot keys that allow you to issue commands to third party imaging software products that store scanned CRFs and DCFs while viewing or editing a page of clinical data or viewing a discrepancy in the Clintrial software. Hot keys are keystrokes or combinations of keystrokes used to issue these commands. A list of the available hot keys is provided in the following table:

Key:	**Description:**
Ctrl+Alt+N	Advance to the next image in the current CRF list.
Ctrl+Alt+P	Return to the previous image in the current CRF list.
Ctrl+Alt+C	Display a CRF image corresponding to the current study page. If viewing a Resolve discrepancy detail, display a corresponding DCF image.
Ctrl+Alt+G	Display the study page corresponding to the current CRF.
Ctrl+Alt+R	This hot key may be customized to perform a function on the CRF image.

7 Working with Subject Lists and Page Lists

Overview

This chapter describes how to use subject lists and page lists.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

What are subject lists and page lists?

Large studies can contain many subjects and study pages, making study page navigation difficult. When working in clinical data study books, you can limit the subjects or study pages in the navigation order to a specific set by using subject lists and page lists.

A *subject list* is a set of specific study subjects.

A *page list* is a set of specific study pages.

What default lists appear in the Clintrial software?

Every Clintrial software protocol includes two default lists. You cannot delete or modify these lists:

- The ALL subject list includes all subjects in the study.
- The ALL page list includes all pages in the study.

Note: If your site is a Multisite replication site, there is a third default list, the SITE subject list, that includes all subjects that are owned at your site.

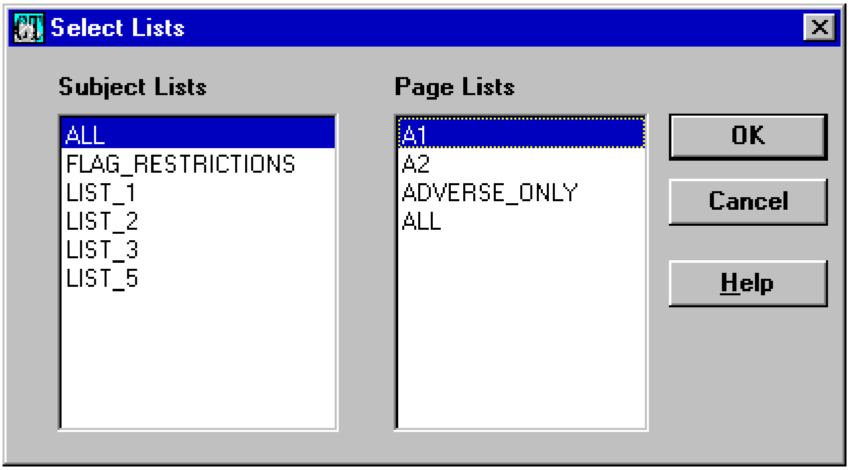
Using subject lists and page lists

When you begin an Enter session, the Clintrial software automatically uses the subject list and page list you used during your last session. At any time, you can select a different list to use. When you use a particular subject list and page list:

- Study page navigation is limited to the subjects or pages in that list.
- The Navigator display is limited to the subjects or pages in that list.

To select a subject list and page list to use:

1. From the **List** menu, choose **Select Lists**. The Select Lists dialog box opens:



2. In the Subject Lists box and the Page Lists box, select the lists that you want to use.
3. Click **OK**.

Note: You can also select the lists that you want to use from the Navigator’s Subject List and Page List drop-down lists.

Creating and modifying lists

You can create as many lists as you need. You create and modify subject lists and page lists by using the List Wizard, a series of dialog boxes that guides you through the steps of entering or modifying list criteria. When you create a list, you can:

- Select specific subjects or pages to include.
- Include or exclude subjects or pages by using a SQL restriction.
- For subject lists only, you can also create a list that includes or excludes subjects based on the presence or absence of flags or notes.

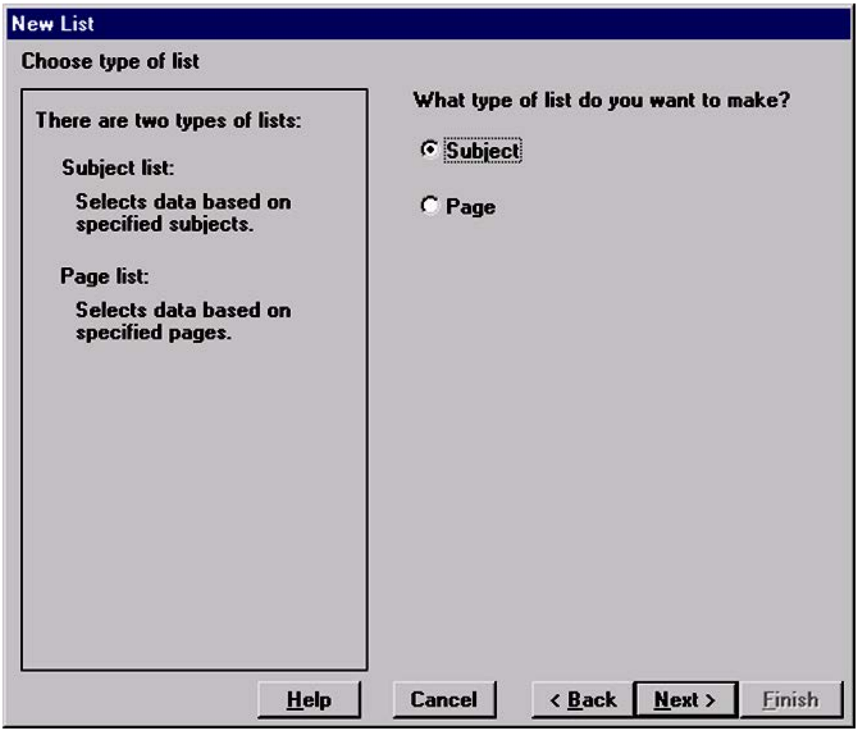
Once created, a list can be modified as needed.

Using the List Wizard

To open the List Wizard, from the **List** menu, select the appropriate command:

- New List
- Edit Subject List
- Edit Page List

The following figure shows a dialog box from the List Wizard:



New List

Choose type of list

There are two types of lists:

Subject list:
Selects data based on specified subjects.

Page list:
Selects data based on specified pages.

What type of list do you want to make?

● Subject

○ Page

What are static lists and dynamic lists?

When you create a list using a flag and note restriction or a SQL restriction, you specify whether the list should be static or dynamic.

A *static list* is a list whose membership does not change unless you edit the list directly.

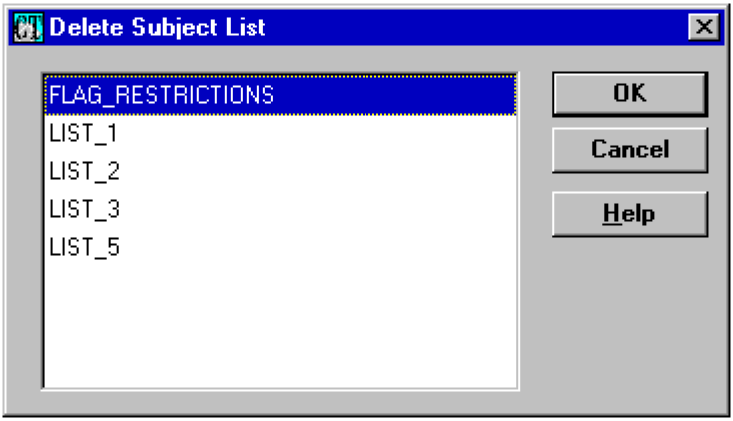
A *dynamic list* is a list whose membership is updated automatically to reflect changes in data.

For example, suppose you create a dynamic subject list for subjects with verification flags attached to their data. The list will automatically include any subjects to whose records verification flags are subsequently attached. Conversely, subjects whose verification flags are later removed will automatically be removed from the list.

Deleting a list

When a list is no longer needed, it can be deleted. To delete a subject list or page list:

1. From the **List** menu, select **Delete Page List** or **Delete Subject List**. The Delete Page List or Delete Subject List dialog box opens:



2. Select the list that you want to delete.
3. Click **OK**.

Enter

8 *Working with Flags and Notes*

Overview

This chapter describes how to use flags and notes.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

What are flags and notes?

Flags and *notes* are attachments to clinical data. You can attach flags and notes to items, records, or observations on a study page.

- *Flags* are attachments to clinical data used to label and monitor data quality problems. For example, you might attach a flag to a number that was illegible on the CRF or is out of the expected range.
- *Notes* are attachments to clinical data used to record annotations made by investigators or sponsors on a CRF. For example:
  - An investigator might comment:

    Subject did not take full regimen of drug.
  - A sponsor might state:

    Investigator confirmed measurement not taken.

Flags and notes can be modified or deleted as necessary. You can create reports showing summary and detail flag and note information. Certain Clintrial software tasks (for example, creating subject lists) allow you to retrieve records based on the flags or notes attached to them.

How are flags different from notes?

Flags are used to mark and monitor questionable or conflicting data. Notes are used strictly to record comments made by investigators and sponsors. In Manage, you can view an Audit Notes Report that summarizes changes to the notes attached to clinical data. Flags are not audited.

Levels of flags and notes

Flags and notes can be attached to individual items, records, or observations on a study page. The object to which a note is attached determines its *level*.

- Item level flags and notes

  An *item* is a Clintrial software object that stores the data collected by a single field. For example, you would choose an item flag for the Temperature field when you cannot decipher the temperature of a subject on a CRF.

- Record level flags and notes

  A *record* is data stored in one row in a database table. For example, you would choose a record note describing that the subject's blood pressure was taken while the subject was standing.

- Observation level flags and notes

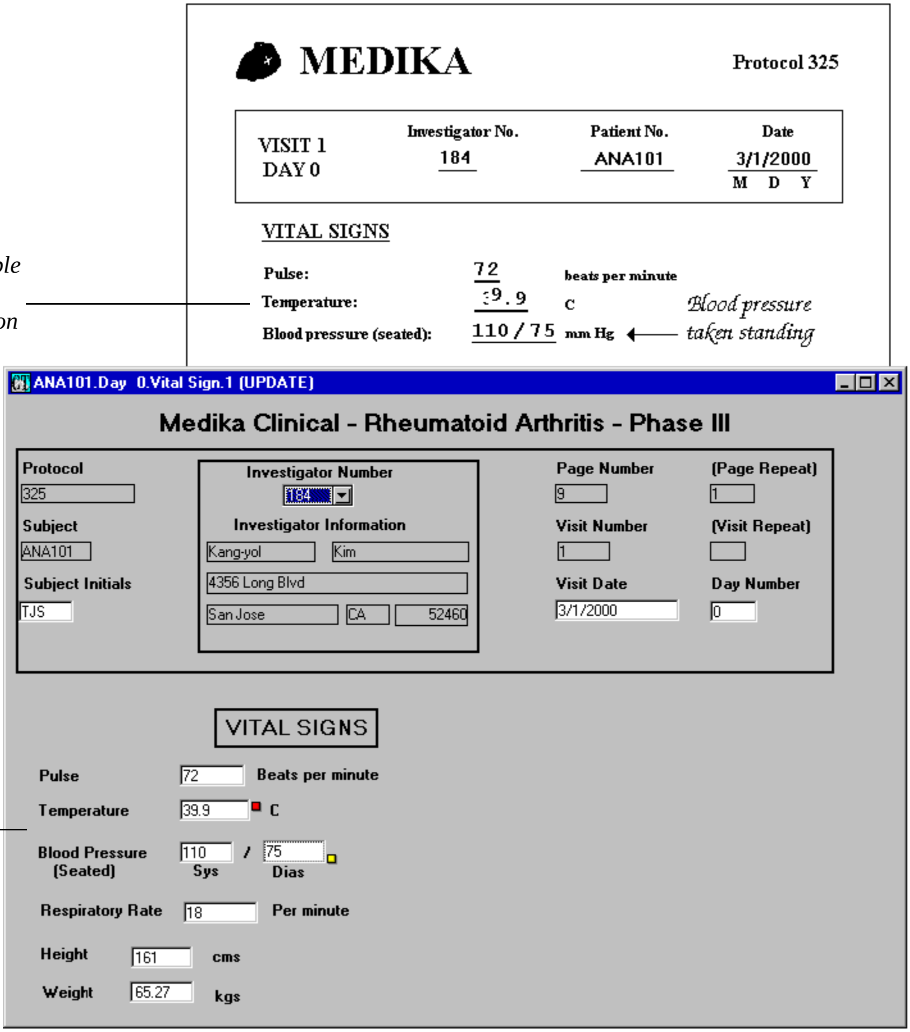
  An *observation* consists of all of the records entered in a single page section that contains repeating items.

  Note**:** Observations in non-subject study books vary from those described here.

  For more information, see page 135.

  Flags and notes can be attached to an observation, as well as to individual records within an observation. You can only use observation flags and notes in repeating page sections.

The following figure shows how information on a CRF can result in a flag and note being entered on a study page:



In this example, because of an illegible temperature and an investigator's note on the CRF, an item flag and an item note were entered on the study page.

MEDIKA **Protocol 325**

VISIT 1
DAY 0

Investigator No.	Patient No.	Date
184	ANA101	3/1/2000 M D Y

VITAL SIGNS

Pulse: 72 beats per minute
Temperature: 39.9 C
Blood pressure (seated): 110 / 75 mm Hg ← *Blood pressure taken standing*

ANA101.Day 0.Vital Sign.1 (UPDATE)

Medika Clinical - Rheumatoid Arthritis - Phase III

Protocol: 325
Subject: ANA101
Subject Initials: TJS

Investigator Number: 184
Investigator Information: Kang-yol | Kim
4356 Long Blvd
San Jose | CA | 52460

Page Number: 9 (Page Repeat): 1
Visit Number: 1 (Visit Repeat):
Visit Date: 3/1/2000 Day Number: 0

VITAL SIGNS

Pulse: 72 Beats per minute
Temperature: 39.9 🟥 C
Blood Pressure (Seated): 110 Sys / 75 Dias 🟨
Respiratory Rate: 18 Per minute
Height: 161 cms
Weight: 65.27 kgs

Flag and note icons

When a flag or note is attached, an icon is placed on the study page next to the data to which the flag or note is attached. Flag icons are red; note icons are yellow. You can open a flag or note for editing by double-clicking its icon.

The following table shows the icons used for flags and notes and where those icons appear on a study page:

Flag or note level:	Icon:	Where it appears:
Item Flag	🟥	Directly to the right of the field containing the flag.
Observation Flag	▣	Upper right corner of the page section containing the observation. Observation flags are not enterable in enrollment or non-subject data study books.
Record Flag	⊟	Variable, depending on the page section to which the record flag is attached. If more than one record flag exists in the page section, only one record-level flag icon is visible in the study page.
Item Note	🟨	Directly to the right of the field containing the note.
Observation Note	▣	Upper right corner of the page section containing the observation. Observation notes are not enterable in enrollment or non-subject data study books.
Record Note	⊟	Variable, depending upon the page section to which the record note is attached. If more than one record note exists in the page section, only one record-level note icon is visible in the study page.

Using flags and notes during verification

You can attach, modify, or delete flags and notes during verification. If your Clintrial software administrator has set the VERIFY_SHOW_TAGS system parameter to:

- **Yes**, all flags and notes are available during verification.
- **No**, only verification flags are available and visible on the study page during verification. Notes are not available.

Note: Flags and notes themselves do not require verification.

Creating reports on flags and notes

You can create reports about the records to which flags or notes are attached. For more information, see Chapter 9.

Working with flags

Flags are attachments to clinical data used to label and monitor data quality problems. For example, you might attach a flag to a number that was illegible on the CRF or is out of the expected range. You can attach flags to items, records, and observations. The object to which a flag is attached determines its *level*. For a description of the levels of flags and notes, see "Levels of flags and notes" on page 98.

When you attach a flag, you must specify:

- The flag *category*, which indicates the problem with the data (for example, Illegible or Missing).
- The flag *name*, which specifies the action to be taken (for example, Accepted or To Be Queried).

You can also include an optional comment.

Flags can be modified or deleted as necessary. You can create reports showing summary and detail flag information. Certain Clintrial software tasks (for example, creating subject lists) allow you to retrieve records based on the flags attached to them.

What are automatic flags?

Automatic flags are flags that the Clintrial software attaches automatically to data in certain situations:

- Blind verification

  During blind verification, when newly entered data conflicts with existing data, the Clintrial software attaches a VERIFICATION/AUTOFLAG to the item.
- Validation

  Your designer can set up rules that automatically attach flags to records that fail validation.

- Data conversion procedures

  Your designer can set up procedures that attach flags to data during data conversion.

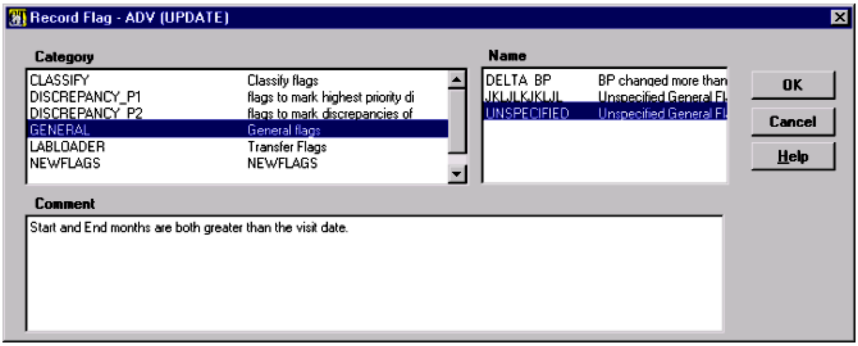
Attaching a flag

To attach a flag, place the cursor in the appropriate data-entry field. Then, from the **Flags** menu, select one of the following flag levels:

- **Item Flag**
- **Record Flag**
- **Observation Flag**

A dialog box opens in which you must specify the flag category and name. You can also include a comment up to a maximum of 2,000
characters.

The following figure shows the Record Flag dialog box:



Category

CLASSIFY	Classify flags
DISCREPANCY_P1	flags to mark highest priority di
DISCREPANCY P2	flags to mark discrepancies of
GENERAL	General flags
LABLOADER	Transfer Flags
NEWFLAGS	NEWFLAGS

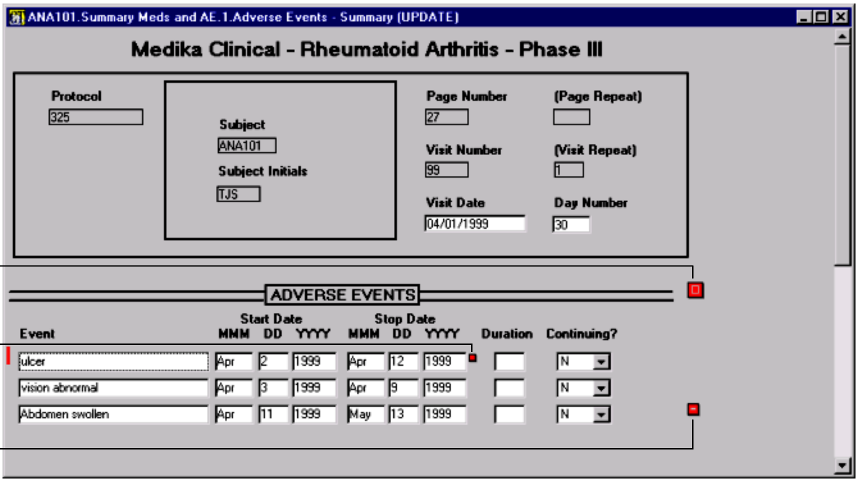
Name

DELTA BP	BP changed more than
JKLJLKJKLJL	Unspecified General Fl
UNSPECIFIED	Unspecified General Fl

Comment

Start and End months are both greater than the visit date.

The following figure shows different levels of flags on a study page:



ANA101.Summary Meds and AE.1.Adverse Events - Summary (UPDATE)

Medika Clinical - Rheumatoid Arthritis - Phase III

Protocol: 325

Subject: ANA101

Subject Initials: TJS

Page Number: 27

(Page Repeat):

Visit Number: 99

(Visit Repeat): 1

Visit Date: 04/01/1999

Day Number: 30

ADVERSE EVENTS

	Start Date			Stop Date				
Event	MMM	DD	YYYY	MMM	DD	YYYY	Duration	Continuing?
ulcer	Apr	2	1999	Apr	12	1999		N
vision abnormal	Apr	3	1999	Apr	9	1999		N
Abdomen swollen	Apr	11	1999	May	13	1999		N

Observation flags apply to all records in a page section.

Item flags apply to individual items.

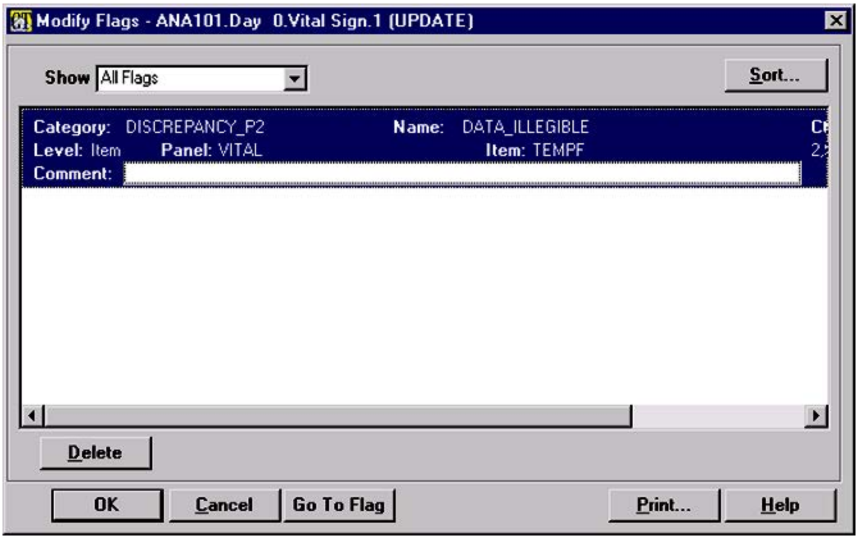
Record flags apply to individual records.

Attaching multiple flags

You can attach multiple flags to a single item, record, or observation, as long as the flag category and name combination of each flag are unique. When you attach multiple flags, only one flag icon appears next to the item, record, or observation.

Modifying or deleting a flag

You modify or delete flags using the Modify Flags dialog box. To open the Modify Flags dialog box, from the **Flags** menu, select **Modify Flags**, or double-click a flag's study page icon. The Modify Flags dialog box opens:


Modify Flags - ANA101.Day 0.Vital Sign.1 (UPDATE)

Show: All Flags

Category: DISCREPANCY_P2 Name: DATA_ILLEGIBLE Cr
Level: Item Panel: VITAL Item: TEMPF 2,
Comment:

To modify a flag:

1. Place the cursor in the flag's Comment field.
2. Add or delete text as needed.
3. Click **OK**.

Note: You can edit the flag's comment, but not the flag's category or name. If the category or name was entered incorrectly, you must delete the flag and reenter it.

To delete a flag:

1. Select the flag that you want to delete. You can select multiple flags at one time.
2. Click **Delete**.
3. Click **OK**.

In the Modify Flags dialog box you can also:

- Show only flags of a certain level by selecting a flag level from the Show drop-down list.
- Sort the list so flags are displayed in a particular order by clicking **Sort**.

- Go to the place on the study page where a particular flag was attached by selecting the flag from the list and clicking **Go To Flag** (the cursor is automatically repositioned at the item, record, or observation).
- Print the displayed flag data by clicking **Print**.

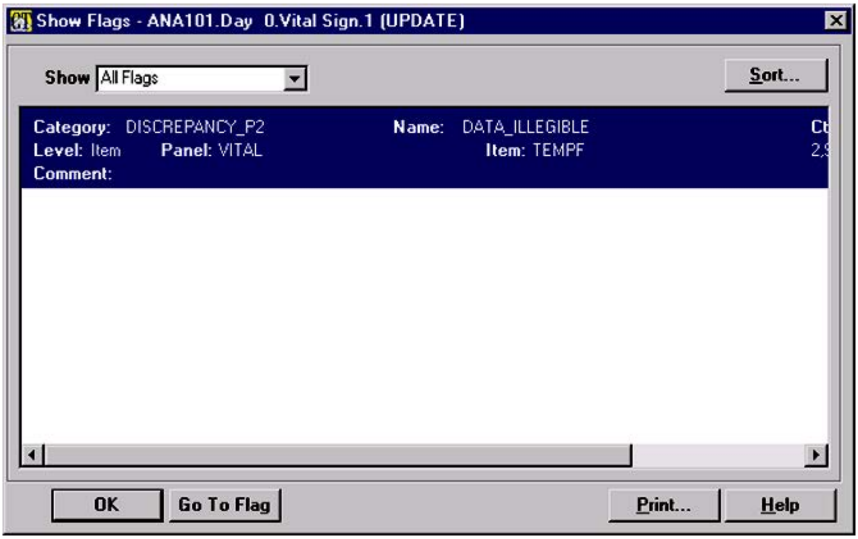
Each flag listed in the Modify Flags dialog box is shown along with its attributes. The following table describes the flag attributes:

Attribute:	**Description:**
Category	The category designation assigned to the flag.
Name	The name designation assigned to the flag.
Level	The flag's level: item, record, or observation.
Panel	The panel (page section) to which the flag was attached.
Item	The item (field) to which the flag was attached.
Comment	The text of the flag comment.
CT Rec ID	The ID of the record to which the flag was attached.

V*iewing flags*

You can view a list of all the flags in the study book using the Show Flags dialog box. The Show Flags dialog box is similar in appearance to the Modify Flags dialog box; however, it does not allow you to modify or delete flags.

To open the Show Flags dialog box, from the **Flags** menu, select **Show Flags**, or double-click a flag’s study page icon. The Show Flags dialog box opens:



Show Flags - ANA101.Day 0.Vital Sign.1 (UPDATE)

Show: All Flags

Category: DISCREPANCY_P2 Name: DATA_ILLEGIBLE Ct
Level: Item Panel: VITAL Item: TEMPF 2.
Comment:

Note: For information on using the buttons and options in the Show Flags dialog box, see page 105.

Working with notes

Notes are attachments to clinical data used to record comments made by investigators or sponsors on a CRF. For example:

- An investigator might comment:

  Subject did not take full regimen of drug.

- A sponsor might state:

  Investigator confirmed measurement not taken.

You can attach notes to items, records, and observations. The object to which a note is attached determines its *level*. For a description of the levels of flags and notes, see "Levels of flags and notes" on page 98.

When you attach a note, you must specify:

- The note *category,* which specifies the source of the note – Sponsor or Investigator.

- The note *name*, which further defines the source of the note. If the category is Sponsor, the name will be the person at your company who annotates clinical data. If the category is Investigator, the name will always be Unspecified.
- The actual text of the note.

Notes can be modified or deleted as necessary. You can create reports showing summary and detail note information. Certain Clintrial software tasks (for example, creating subject lists) allow you to retrieve records based on the notes attached to them.

What are sponsor notes?

Sponsor notes are annotations that the sponsor attaches to clinical data. You can attach multiple sponsor notes to the same level of data, as long as each sponsor note name is unique. The designer can create additional note names for sponsor notes.

What are investigator notes?

Investigator notes are annotations that investigators attach to clinical data. You can attach only one investigator note to each level of data, and that investigator note always has the UNSPECIFIED note name.

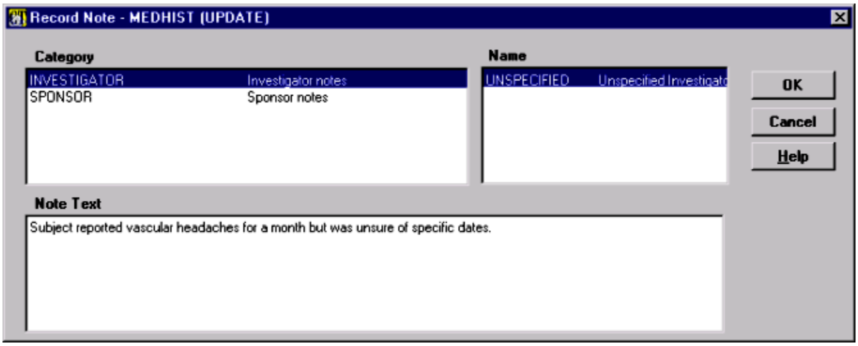
Attaching a note

To attach a note, place the cursor in the appropriate data-entry field. Then, from the **Notes** menu, select one of the following note levels:

- **Item Note**
- **Record Note**
- **Observation Note**

A dialog box opens in which you must specify the note category (SPONSOR or INVESTIGATOR), the note name, and the note text. The maximum length for note text is 2,000 characters.

The following figure shows the Record Note dialog box:



Record Note - MEDHIST (UPDATE)

Category

INVESTIGATOR	Investigator notes
SPONSOR	Sponsor notes

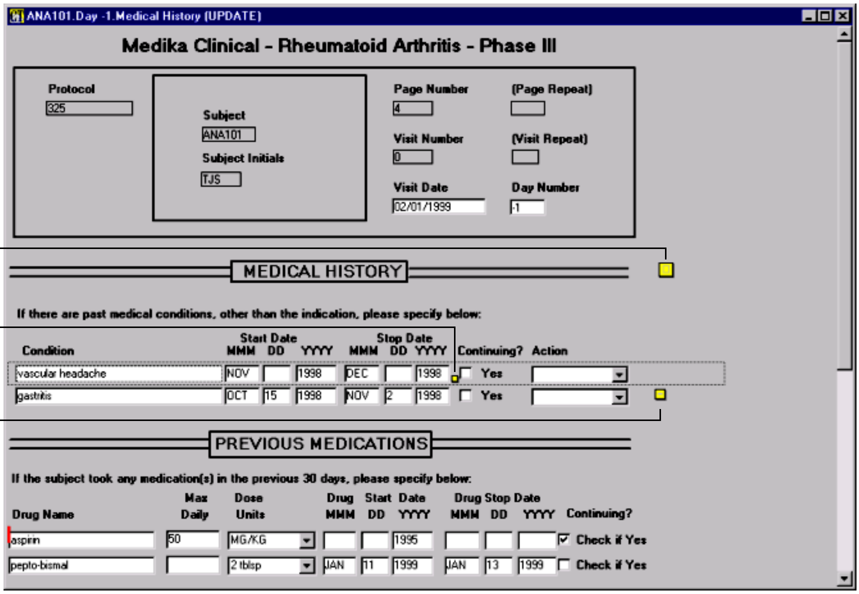
Name

UNSPECIFIED	Unspecified Investigato

Note Text

Subject reported vascular headaches for a month but was unsure of specific dates.

The following figure shows different levels of notes on a study page:



ANA101.Day -1.Medical History (UPDATE)

Medika Clinical - Rheumatoid Arthritis - Phase III

Protocol: 325
Subject: ANA101
Subject Initials: TJS
Page Number: 4
(Page Repeat):
Visit Number: 0
(Visit Repeat):
Visit Date: 02/01/1999
Day Number: -1

MEDICAL HISTORY

If there are past medical conditions, other than the indication, please specify below:

Condition	Start Date MMM	DD	YYYY	Stop Date MMM	DD	YYYY	Continuing?	Action
vascular headache	NOV		1998	DEC		1998	☐ Yes	
gastritis	OCT	15	1998	NOV	2	1998	☐ Yes	

PREVIOUS MEDICATIONS

If the subject took any medication(s) in the previous 30 days, please specify below:

Drug Name	Max Daily	Dose Units	Drug Start Date MMM	DD	YYYY	Drug Stop Date MMM	DD	YYYY	Continuing?
aspirin	50	MG/KG			1995				☑ Check if Yes
pepto-bismal		2 tblsp	JAN	11	1999	JAN	13	1999	☐ Check if Yes

Observation notes apply to all records.

Item notes apply to individual items.

Record notes apply to individual records.

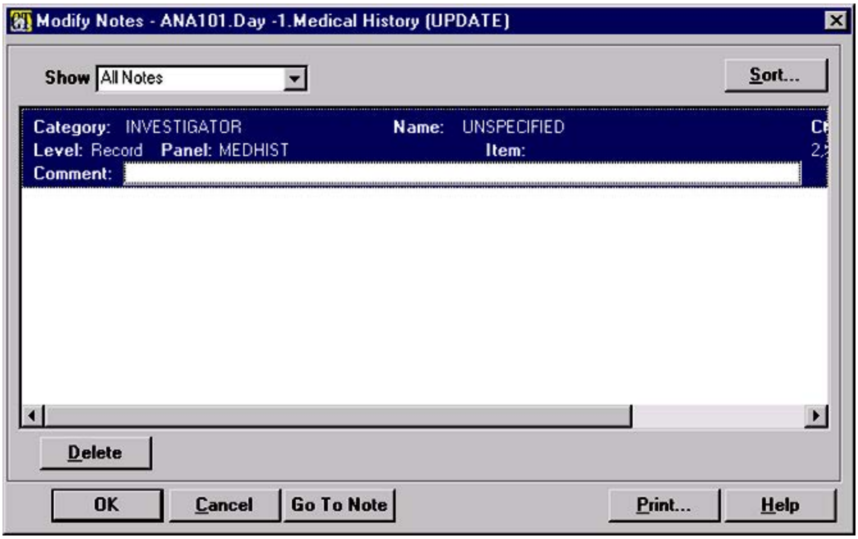
Attaching multiple notes

You can attach multiple sponsor notes to a single item, record, or observation, provided each note name used is unique. When you attach multiple notes, only one note icon appears next to the item, record, or observation.

You cannot attach multiple investigator notes.

Modifying or deleting a note

You modify or delete notes using the Modify Notes dialog box. To open the Modify Notes dialog box, from the **Notes** menu, select **Modify Notes**, or double-click on a note’s study page icon. The Modify Notes dialog box opens:


Modify Notes - ANA101.Day -1.Medical History (UPDATE)

Show: All Notes

Category: INVESTIGATOR Name: UNSPECIFIED C
Level: Record Panel: MEDHIST Item: 2,
Comment:

Note: If the designer has enabled notes auditing, the changes that you make will be audited.

To modify a note:

1. Place the cursor in the note's Comment field.
2. Add or delete text as needed.
3. Click **OK**.

Note: When modifying a note, you can edit a note's comment, but not the note’s category or name. If the category or name was entered incorrectly, you must delete the note and reenter it.

To delete a note:

1. Select the note that you want to delete. You can select multiple notes at one time.
2. Click **Delete**.
3. Click **OK**.

In the Modify Notes dialog box you can also:

- Show only notes of a certain level by selecting a note level from the Show drop-down list.
- Sort the list so notes are displayed in a particular order by clicking **Sort**.
- Go to the place on the study page where a particular note was attached by selecting the note from the list and clicking **Go To Note**. (The cursor is automatically repositioned at the item, record, or observation.)
- Print the displayed note data by clicking **Print**.

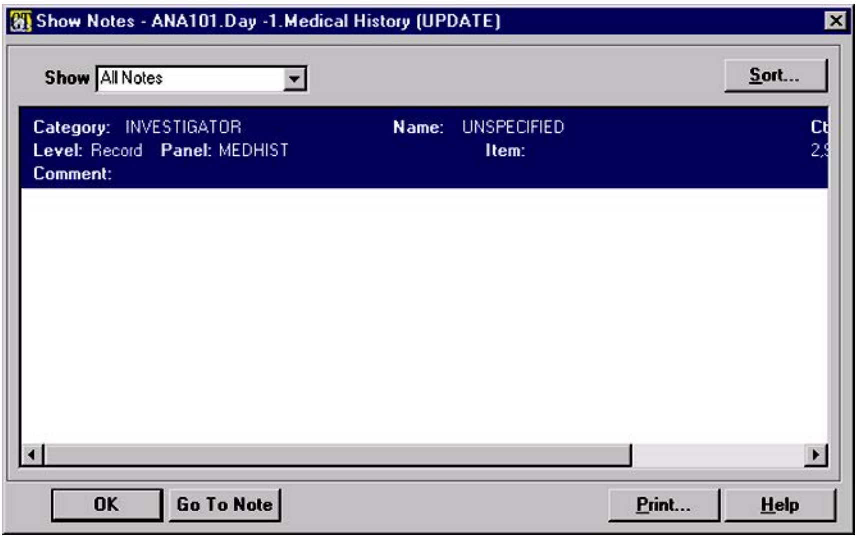
Each note listed in the Modify Notes dialog box is shown along with its attributes. The following table describes the note attributes:

Attribute:	**Description:**
Category	The category designation assigned to the note.
Name	The name designation assigned to the note.
Level	The note's level: item, record, or observation.
Panel	The panel (page section) to which the note was attached.
Item	The item (field) to which the note was attached.
Comment	The text of the note comment.
CT Rec ID	The ID of the record to which the note is attached.

Viewing notes

You can view notes in the current study book using the Show Notes dialog box. The Show Notes dialog box is similar in appearance to the Modify Notes dialog box; however, it does not allow you to modify or delete notes.

To open the Show Notes dialog box, from the **Notes** menu, select **Show Notes**, or double-click a note’s study page icon. The Show Notes dialog box opens:



Show Notes - ANA101.Day -1.Medical History (UPDATE)

Show: All Notes

Category: INVESTIGATOR Name: UNSPECIFIED Ct

Level: Record Panel: MEDHIST Item: 2,

Comment:

Note: For information on using the buttons and options in the Show Notes dialog box, see "Attaching a note" on page 108.

9 Working with Reports

Overview

This chapter describes Enter reports.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

Enter reports

The following reports are available in Enter:

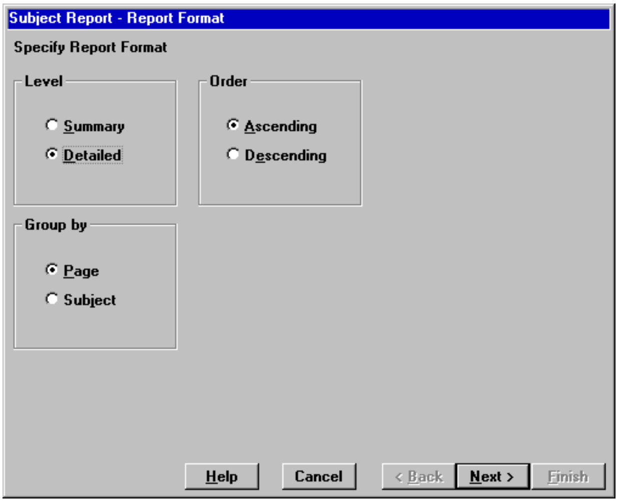
- The *Subject Report* provides information about the status of records in clinical data study books.
- The *Flag Report* provides information about records to which flags are attached.
- The *Note Report* provides information about records to which notes are attached.
- The *Verification Report* provides information about value conflicts in records that have failed blind verification.

You can only create reports for the protocol in which you are working. Once created, Enter reports can be viewed online, printed, or saved to a file.

Creating reports

You create Enter reports by using the Report Wizard; a series of dialog boxes in which you specify the report's format and content. You do not have to use all of the Wizard's dialog boxes. As soon as the **Finish** button becomes available, it indicates that the minimum amount of necessary information has been entered. At any point you can return to change an earlier entry by clicking **Back**.

The following figure shows the Report Wizard:



Subject Report - Report Format

Specify Report Format

Level

○ Summary
● Detailed

Order

● Ascending
○ Descending

Group by

● Page
○ Subject

Printing and saving reports

You can print a report by selecting **Print** from the **File** menu. To save a report to a file (for example, a Lotus spreadsheet), from the **File** menu, select **Save As**.

What are summary and detailed reports?

The Flag, Note, and Subject reports are available in two different formats: detailed and summary. The Verification Report is available in only one format.

- The *summary* report format provides a count and categorical breakdown of the records it finds. For example, the Summary Subject Report can provide the total number of records in the DEMOGRAPHY page section having the status *Verified*, *Unverified*, and so on.
- The *detailed* report format provides in-depth information about the attributes of each record it retrieves. For example, the Detailed Subject Report provides details on all subject records in the DEMOGRAPHY page section.

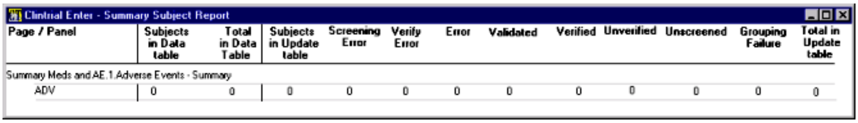
Subject Report

The *Subject Report* provides information about the status of subject records. The Subject Report can be created in either a summary or detailed format.

What is the Summary Subject Report?

The *Summary Subject Report* gives a count of the record totals for each record status. For example, you might create a Summary Subject report that shows the total number of subject records for each status type in the Inclusion/Exclusion page section.

The following figure shows a section from the Summary Subject Report:



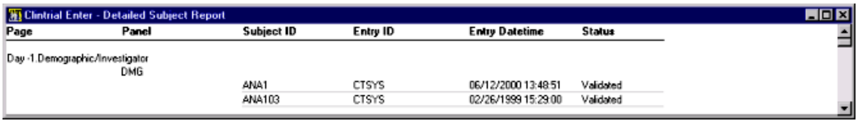
Clintrial Enter - Summary Subject Report

Page / Panel	Subjects in Data table	Total in Data Table	Subjects in Update table	Screening Error	Verify Error	Error	Validated	Verified	Unverified	Unscreened	Grouping Failure	Total in Update table
Summary Meds and AE.1.Adverse Events - Summary												
ADV	0	0	0	0	0	0	0	0	0	0	0	0

What is the Detailed Subject Report?

The *Detailed Subject Report* shows the status of individual records. For example, you might create a Detailed Subject report that lists only subject records in the Inclusion/Exclusion page section with the status *Unverified*.

The following figure shows a section from the Detailed Subject Report:



Clintrial Enter - Detailed Subject Report

Page	Panel	Subject ID	Entry ID	Entry Datetime	Status
Day -1.Demographic/Investigator					
	DMG				
		ANA1	CTSYS	06/12/2000 13:48:51	Validated
		ANA103	CTSYS	02/26/1999 15:29:00	Validated

Creating a Subject Report

To create a Subject Report, from the **Reports** menu, select **Subject**. The Report Wizard opens and guides you through the following dialog boxes in which you specify criteria for the report:

- Report Format
- Subjects and Pages
- Entry Information
- Record Status (for the Detailed Subject Report only)
- Flag/Note Details

After specifying the criteria, click **Finish** to create the report.

Subject Report criteria

The following table lists criteria options available in the Report Wizard when creating a Subject Report:

Report criteria:	Options:
Report Format	Select either a Summary or Detailed format.
	Sort records in Ascending or Descending order.
	Group records by Page or Subject, to determine how they appear in the report. ***Note*****:** The Group By Page version of the report includes totals from both the update table and the data table. The Group By Subject version includes data from the update table only.
Subjects and Pages	Choose from the Subject List drop-down list.
	Select Page(s) to include in the report.
Entry Information	Select records based on the User IDs that entered the records. You can specify one or more User IDs. "ALL" is the default.
	Select records based on the Date Range during which they were entered. If no date is specified, all available dates are used. You can leave one or both fields blank to signify Any Date. ***Note*****:** The end date (To: date) is not inclusive. You must specify the last day you want to include plus one day. For example, to include 3/1/1999 in the report, you must set this field to 3/2/1999.
Record Status (Detailed Subject Report only)	Select one or more of the following record statuses to include in the report: • Validated • Verified • Unverified • Unscreened • Error • Screen Error • Verify Error

Report criteria:	Options:
Flag/Note Details	Select records based on one or more of the following: • Category of flag or note. • Page of flag or note. • Names of flag or note. • Level to which flag or note is attached (Item, Record, Observation, or Any). • Items to which flag or note is attached. Click **Add** to activate the Selection box, which enables you to build restrictions.

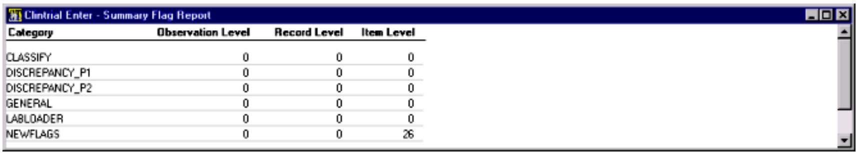
Flag Report

The *Flag Report* provides information about records to which flags are attached. It can include records in the update table, the data table, or both. The Flag Report can be created in either a summary or detailed
format.

What is the Summary Flag Report?

The *Summary Flag Report* lists the number of flags for each flag category.

The following figure shows sections from the Summary Flag Report:

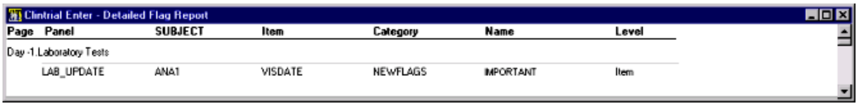


Category	Observation Level	Record Level	Item Level
CLASSIFY	0	0	0
DISCREPANCY_P1	0	0	0
DISCREPANCY_P2	0	0	0
GENERAL	0	0	0
LABLOADER	0	0	0
NEWFLAGS	0	0	26

What is the Detailed Flag Report?

The *Detailed Flag Report* lists records that have flags attached to them. If a record has multiple flags attached to it, there is a separate entry in the report for each.

The following figure shows a section from the Detailed Flag Report:



Clintrial Enter - Detailed Flag Report

Page	Panel	SUBJECT	Item	Category	Name	Level
Day -1.Laboratory Tests						
	LAB_UPDATE	ANA1	VISDATE	NEWFLAGS	IMPORTANT	Item

Creating a Flag Report

To create a Flag report, from the **Reports** menu, select **Flag**. The Report Wizard opens and guides you through the following dialog boxes in which you specify criteria for the report:

- Report Format
- Subjects and Pages
- Flag/Note Details

After specifying the criteria, click **Finish** to create the report.

Flag Report criteria

The following table lists criteria options available in the Report Wizard when creating a Flag Report:

Report criteria:	**Options:**
Report Format	Select either a Summary or Detailed format.
	Sort records in Ascending or Descending order.
	Specify records by Page or Subject, to determine how they appear in the report.
Subjects and Pages	Choose from the Subject List drop-down list.
	Select Page(s) to include in the report.
Flag/Note Details	Select records based on one or more of the following: • Level to which flag or note is attached (Item, Record, Observation, or Any). • Category of flag or note.

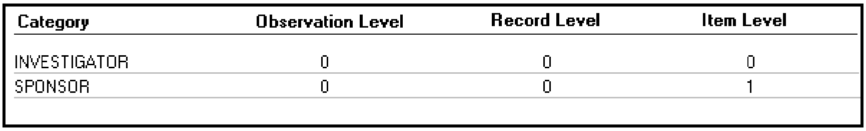
Note Report

The *Note Report* provides information about records to which notes are attached. It can include records in the update table, the data table, or both. The Note Report can be created in either a summary or detailed format.

What is the Summary Note Report?

The *Summary Note Report* lists the number of notes for each note category.

The following figure shows a section from the Summary Note Report:

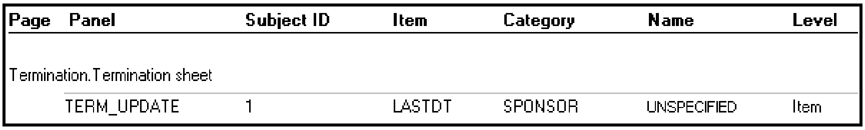


Category	Observation Level	Record Level	Item Level
INVESTIGATOR	0	0	0
SPONSOR	0	0	1

What is the Detailed Note Report?

The *Detailed Note Report* lists records that have notes attached to them. If a record has multiple notes attached to it, there is a separate entry in the report for each.

The following figure shows a section from the Detailed Note Report:



Page	Panel	Subject ID	Item	Category	Name	Level
Termination.Termination sheet						
	TERM_UPDATE	1	LASTDT	SPONSOR	UNSPECIFIED	Item

Creating a Note Report

To create a Note report, from the **Reports** menu, select **Note**. The Report Wizard opens and guides you through the following dialog boxes in which you specify criteria for the report:

- Report Format
- Subjects and Pages
- Flag/Note Details

After specifying the criteria, click **Finish** to create the report.

Note Report criteria

The following table lists criteria options available in the Report Wizard when creating a Note Report:

Report criteria:	**Options:**
Report Format	Select either a Summary or Detailed format. Sort records in Ascending or Descending order. Specify records by Page or Subject, to determine how they appear in the report.
Subjects and Pages	Choose from the Subject List drop-down list. Select Page(s) to include in the report.
Flag/Note Details	Select records based on one or more of the following: • Level to which flag or note is attached (Item, Record, Observation, or Any). • Category of note.

Verification Report

The *Verification Report* shows the total number of errors that occur during blind verification for all subjects, and conflicts between original and reentered values. You can use this report to track potential data problems.

Creating a Verification Report

To create a Verification Report, from the **Reports** menu, select **Verification**. The Report Wizard opens and guides you through the following dialog boxes in which you specify criteria for the report:

- Subjects and Pages
- Entry Information

After specifying the criteria, click **Finish** to create the report.

Verification Report criteria

The following table lists criteria options available in the Report Wizard when creating a Verification Report:

Report criteria:	**Options:**
Subjects and Pages	Choose from the Subject List drop-down list.
	Select Page(s) to include in the report.
Entry Information	Select records based on the User IDs that entered the record. You can specify one or more User IDs. The default is ALL.
	Select records based on the Date Range. The default is ALL.

10 *Verifying Clinical Data*

Overview

This chapter describes how to verify clinical data.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

What is verification?

Verification (or *double data-entry*) is the process of reentering clinical data on a study page to ensure that the original entry was correct. The Clintrial software compares the old and new values, checking them for errors or conflicts.

Your designer determines which panels (page sections) and items (fields) require verification. If verification is not required for the records on a particular study page, you cannot verify them. You can only verify records that are in the update table and that have the status *Unverified*.

The Clintrial software supports these two types of verification:

- Interactive verification
- Blind verification

Only one type of verification is used in a given protocol. The type of verification used is determined by the VERIFY_BLIND protocol parameter, set in Admin by your Clintrial software administrator or designer. If VERIFY_BLIND is set to Yes, blind verification is used. Otherwise, interactive verification is performed.

What is interactive verification?

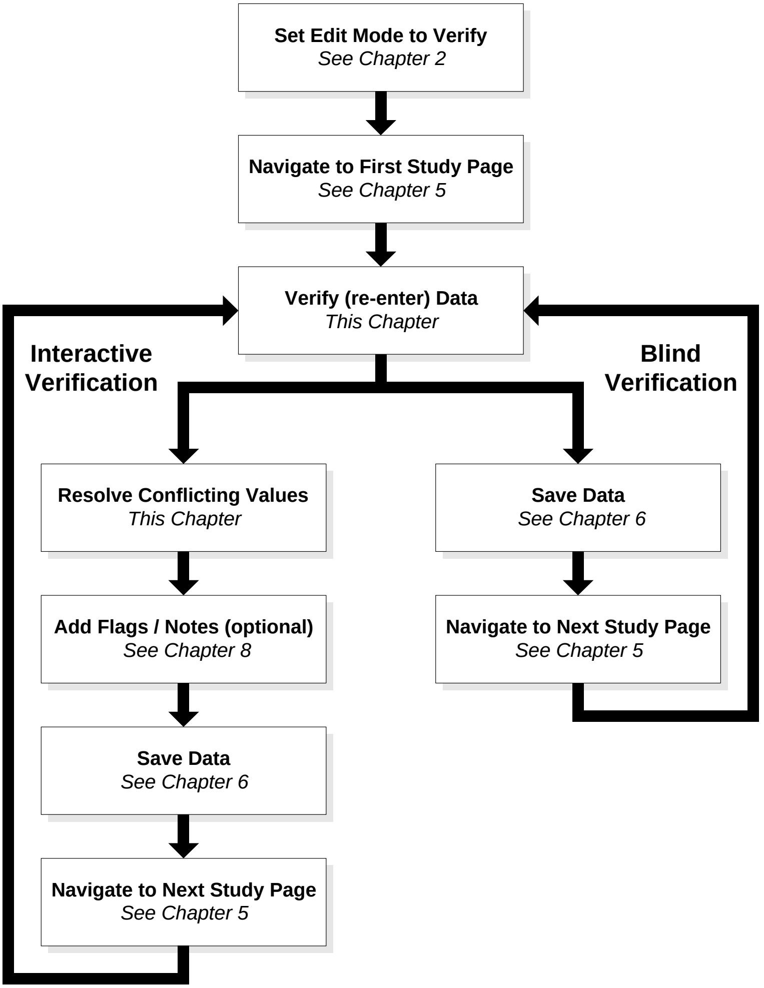
In *interactive verification*, the Clintrial software compares old and new values as you enter data in each field. If there is a conflict, the Clintrial software notifies you immediately. You can then choose to keep either the new value or the original one. All records have their status changed to *Verified* except those records that you specifically select to fail (that is, those records whose status you set to *Verification Error*).

What is blind verification?

In *blind verification*, the Clintrial software compares old and new values when you save a study page; however, the new values DO NOT overwrite the old ones, nor can you choose which value to retain. If there is a conflict, the Clintrial software attaches a VERIFICATION/AUTOFLAG flag to the item and an entry is made in the tags table. Records in which there is a value conflict fail verification and have their status changed to *Verification Error.* These records are listed in the Verification Report. Records that pass have their status changed to *Verified*.

Verification workflow

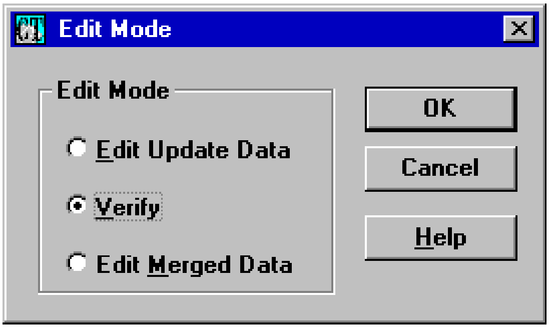
The following figure shows a typical workflow for verifying data:


Set Edit Mode to Verify
See Chapter 2
Navigate to First Study Page
See Chapter 5
Verify (re-enter) Data
This Chapter
Interactive Verification
Blind Verification
Resolve Conflicting Values
This Chapter
Add Flags / Notes (optional)
See Chapter 8
Save Data
See Chapter 6
Navigate to Next Study Page
See Chapter 5
Save Data
See Chapter 6
Navigate to Next Study Page
See Chapter 5

Setting the edit mode for verification

By default, when you start Enter, the edit mode is set to Edit Update Data, the mode for entering new data. To perform verification, you must set the edit mode to **Verify**. To set the edit mode, first close any open study pages. Then:

1. From the **File** menu, select **Edit Mode**. The Edit Mode dialog box opens:



Edit Mode

○ Edit Update Data

● Verify

○ Edit Merged Data

2. Select **Verify**.
3. Click **OK**.

Using flags and notes during verification

You can add, modify, or delete flags and notes during verification. If your Clintrial software administrator has set the VERIFY_SHOW_TAGS system parameter to:

- **Yes**, all flags and notes are available.
- **No**, only verification flags are available and visible on the study page. Notes are not available.

Note: Flags and notes themselves do not require verification.

Adding data subsequent to verification

Data added to a page after verification will receive the status of Verified (1) if it is in the UPDATE table, and Validated (0) if it is in the DATA table (and successfully passes validation).

Nonverifiable panels

During verification, you may encounter study pages that contain both panels (page sections) requiring verification and panels that do not. The fields in these nonverifiable panels remain enterable and, unlike panels that require verification, display previously entered data. If you are performing interactive verification, you can make and save changes to the data in these panels. If you are performing blind verification, changes you make to the data in these panels will not be saved.

Verification in non-subject study books

Verification processes records in non-subject study books on the observation level. When performing verification in a non-subject study book with multiple observations, it is possible to verify the records in one observation without verifying the records in others (namely, by using an SQL restriction that retrieves only the records in the observation you want to verify).

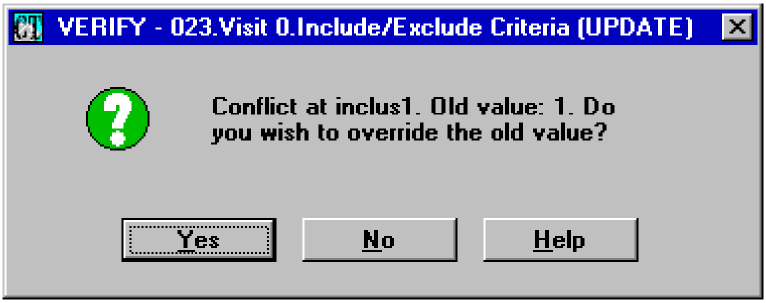
Once the records in an observation have been verified, if you add additional records to the observation, the new records receive the status *Verified*.

Performing interactive verification

After you set the edit mode to **Verify**, open the study page in which you want to verify data. During verification, you can only open study pages with records that require verification. If a particular study page contains no records that require verification, the following message displays when you try to open the page:

There are no records to verify.

Enter data in all fields that require verification. If there is a conflict between the old and new data, the Clintrial software displays a message asking whether you want to use the new value or retain the old one:



Conflict at inclus1. Old value: 1. Do you wish to override the old value?

Save the changes when you are done. Once you do this, records in the study page have their status changed to *Verified* and cannot be verified again.

Adding and deleting data

When you save a study page during verification, the Clintrial software prompts you for confirmation if you have:

- Added a new row of repeating items.
- Deleted a row of repeating items.
- Deleted data from a field that originally contained a value.

If you elect to keep the change, you may also be required to provide a Reason for Change.

Caution: If you add new repeating items during verification, when the data is saved, the new repeating items are assumed to be verified and their individual records are given a status of 1 (*Verified*).

Specifying a Reason for Change

When you modify a record during interactive verification, you may be required to provide a Reason for Change if both of these conditions are true:

- The current protocol requires a reason for change; that is, the Manage AUDIT_REASON_REQD system parameter has been set to **Yes**.
- The audit start point for the protocol has been reached. For example, if auditing is set to begin at data entry, any change to records in the update table requires a Reason for Change.

Note: For more information on specifying a Reason for Change, see Chapter 6.

Setting record status to Verification Error

When you save a study page during interactive verification, all records in that page are automatically verified. In some studies, however, your data manager may require you to set the *Verification Error* status on study pages where there is a conflict or problem. This must be done manually. When you do this, all records on that study page have their status changed to *Verification Error.* To set the V*erification Error* status, place the cursor in any enterable field on the page, and from the **Flags** menu, select **Set Verify Error**.

Performing blind verification

After you set the edit mode to **Verify**, open the study page in which you want to verify data. During verification, you can only open study pages with records that require verification. If a particular study page contains no records that require verification, the following message displays when you try to open the study page:

```
There are no records to verify.
```

Enter data in all required fields. Save the window.

The Clintrial software automatically attaches a VERIFICATION/AUTOFLAG flag if you:

- Reenter a value that is different than the original value.
- Leave blank a field that originally had a value.
- Fail to reenter a previously entered row of repeating items.
- Enter an additional row of repeating items.

The VERIFICATION/AUTOFLAG flag comment contains the original and reentered values, and the account names of the data-entry operator(s) who entered the values.

After completing blind verification, you can create a Verification Report to view information on records that failed verification. For more information, see Chapter 9.

11 *Working with Non-Subject Study Books*

Overview

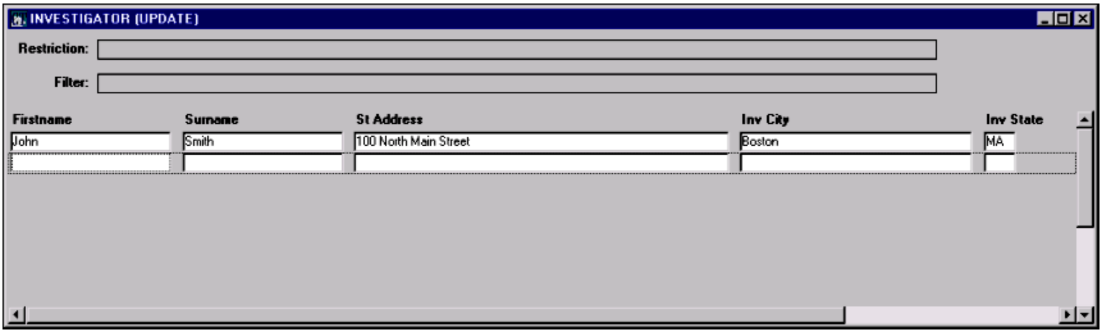
This chapter describes how to use non-subject study books.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

Working with non-subject study books

Non-subject study books are study books that you use to enter nonclinical data; that is, data not related to a particular study subject. Nonclinical data can include data in standard coding thesauruses, view codelists, or laboratory normal ranges. A clinical data protocol can contain any number of non-subject study books. Like enrollment study books, non-subject study books contain only one study page that opens automatically when you open the study book.

The following figure shows a study page from a non-subject study book:



Firstname	Surname	St Address	Inv City	Inv State
John	Smith	100 North Main Street	Boston	MA

Adding a record

To add a record in a non-subject study book:

1. From the **Edit** menu, select **Add Record**.
2. Complete the record's fields. The fields will vary according to the requirements of your study. You can use Enter's Item Help for assistance to complete individual fields.
3. Save the study page.

Deleting a record

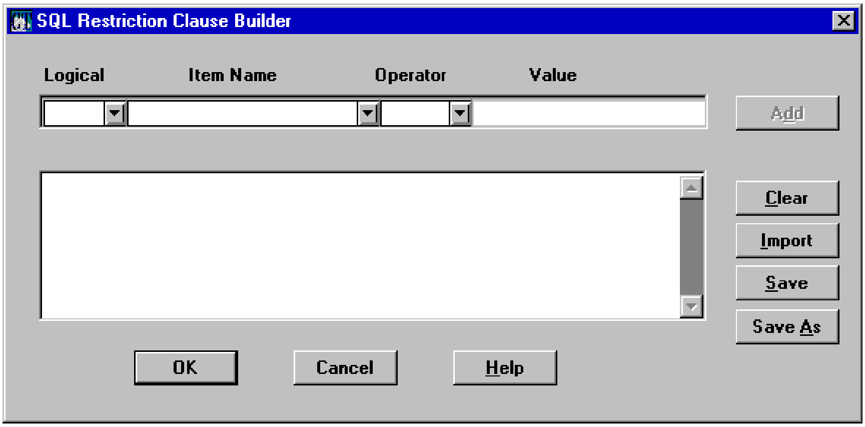
To delete a record in a non-subject study book:

1. Place the cursor in a field in the record that you want to delete.
2. From the **Edit** menu, select **Delete Record**.
3. Save the study page.

Using an SQL restriction

You can use an SQL restriction to specify which records the study book should retrieve from the database. For example, you might use an SQL restriction to specify that only records entered on a certain date should be retrieved.

To use an SQL restriction, from the **File** menu, select **Respecify**. The SQL Restriction Clause Builder opens:



Note: The SQL Restriction Clause Builder dialog box also opens automatically when you first open a non-subject study book.

Filtering and sorting

You can filter and sort records in a non-subject study page. Filtering and sorting allow you to determine which records are displayed in the study page and the order in which they appear.

- To filter records, from the **View** menu, select **Filter**. The Specify Filter dialog box opens.

- To sort records, from the **View** menu, select **Sort**. The Specify Sort Columns dialog box opens.

  Note: Filtering does not limit the retrieval of records from the database; filtering only determines which retrieved records are visible in the study page.

Saving data in a non-subject study page

You can save the data in a non-subject study page at any time by selecting **Save** from the **File** menu. If you try to close the study page without saving it, the Clintrial software prompts you to save it.

If the ENT_FORM_INSERT user preference is set to **Yes**, data is saved automatically when you close a study page. No confirmation to save is required on your part. The Clintrial software default for this preference is **No**. To close a study page without saving it, from the **File** menu, select **Close Without Saving**.

Printing a non-subject study page

To print a non-subject study page, from the **File** menu, select **Print**. All records that appear on the study page are printed. Prior to printing,

Note: Clintrial prints what can fit on a page, as determined by paper size, page orientation (landscape vs. portrait), etc.

Note: You can use the **Filter** and **Sort** commands to specify which records appear on the page and the order in which they should be listed.

Using observations in non-subject study books

Your designer can configure a non-subject study book for grouping. When this is done, you create observations in the study book which you subsequently use to group the records you enter into sets. The records belonging to a particular observation are then processed together during events such as verification and merging.

Observations in non-subject study books

Observations in non-subject study books work differently than observations in clinical data study books.

In a clinical data study book, an *observation* consists of all of the records entered in a single page section that contains repeating items.

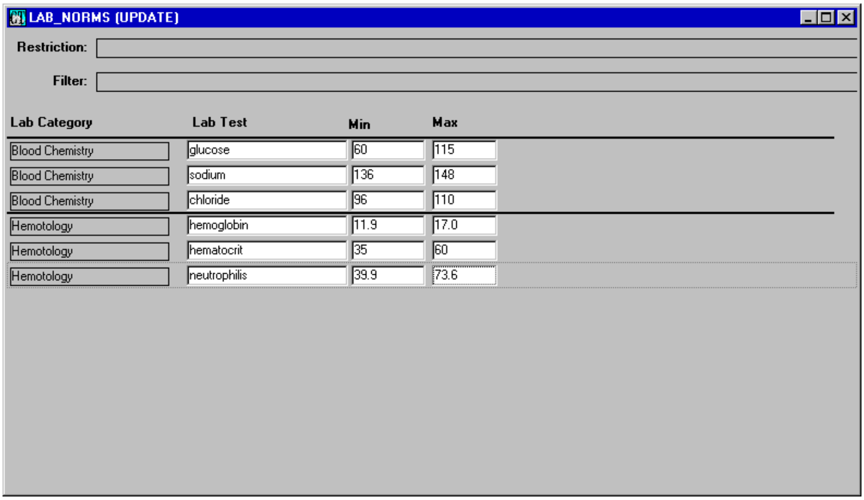
In a non-subject study book, an observation is a set of records within a page section that have identical values for the items the designer has designated as the *group keys*. Thus, in a non-subject study book, a single page section can contain multiple observations, each with its own subset of records. Each record you enter in the page section is grouped with a particular observation.

Creating an observation

You must create an observation first before you can begin adding records to it. You create an observation by defining a unique set of values for the items the designer has designated as the group keys. Each record added to the observation automatically uses these values.

For example, suppose a study page contains these items: Lab Test Type, Lab Test, Minimum Range, and Maximum Range, and Lab Test Type is the group key. If you create an observation using Blood Chemistry as the value for Lab Test Type, all of the records added to that observation use Blood Chemistry as well.

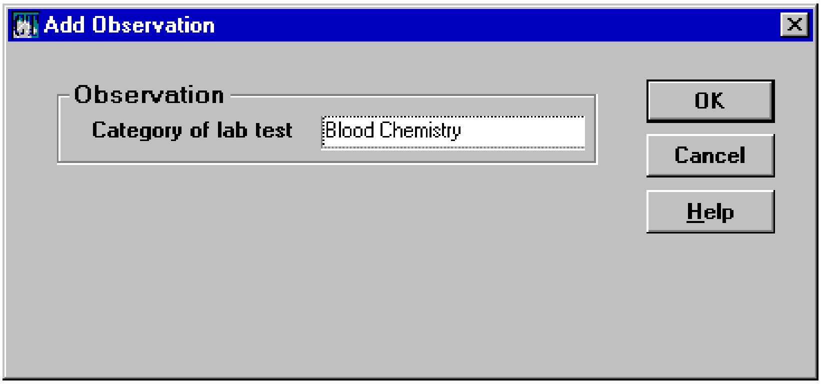
The following figure shows a non-subject study page with observations:



Lab Category	Lab Test	Min	Max
Blood Chemistry	glucose	60	115
Blood Chemistry	sodium	136	148
Blood Chemistry	chloride	96	110
Hemotology	hemoglobin	11.9	17.0
Hemotology	hematocrit	35	60
Hemotology	neutrophilis	39.9	73.6

To create an observation:

1. From the **Edit** menu, select **Add Observation**. The Add Observation dialog box opens:



Category of lab test: Blood Chemistry

2. Enter or select a value for each field in the dialog box. These values will serve as the grouping keys for the observation. The actual names and number of fields in the dialog box are determined by the designer. If the dialog box contains multiple fields, as a group, the values you enter must be unique. Key values are always of data type FIXED or VARCHAR2.
3. Click **OK**. The dialog box closes and the new observation appears at the bottom of the study page. A blank row is inserted in which you can enter the first record for the new observation.

Adding a record to an observation

To add a record to an observation:

1. Place the cursor in an existing record in the observation to which you want to add the new record.
2. From the **Edit** menu, select **Add Record**.
3. Enter the required values for the record.
4. Save the study page.

Deleting a record from an observation

To delete a record from an observation:

1. Place the cursor in a field in the record that you want to delete.
2. From the **Edit** menu, select **Delete Record**.
3. Save the study page.

Sorting records in an observation

Within an observation, records are listed according to the *sort keys*. A sort key is an item whose values are used to determine the order in which records within a particular observation appear. For example, the designer can specify that records sort alphabetically according to the values entered for an item named Drug Name. When you add a new record to an observation, it initially appears at the bottom of the order. To sort the records in an observation so that new entries appear in their correct position, from the **View** menu, select **Regroup**.

Verification in non-subject study books

Verification processes records in non-subject study books on the observation level. Verification uses panel keys, set up by the designer, to uniquely identify repeating records. When performing verification in a non-subject study book with multiple observations, it is possible to verify the records in one observation without verifying the records in other observations, namely, by using an SQL restriction that retrieves only the records in the observation you want to verify.

Once the records in an observation have been verified, if you add additional records to the observation, the new records receive the status *Verified*.

Enter

12 *Working with Discrepancies*

Overview

This chapter describes how to work with discrepancies in Enter.

For step-by-step instructions on any of the tasks described in this chapter, see the Enter Help.

Working with discrepancies

If you have the required Resolve access rights and your protocol is enabled for Resolve, you can display, create, and delete discrepancies from within Enter.

You can work with discrepancies in any kind of study book. You can also start Resolve from within Enter to view and work with discrepancies in greater detail.

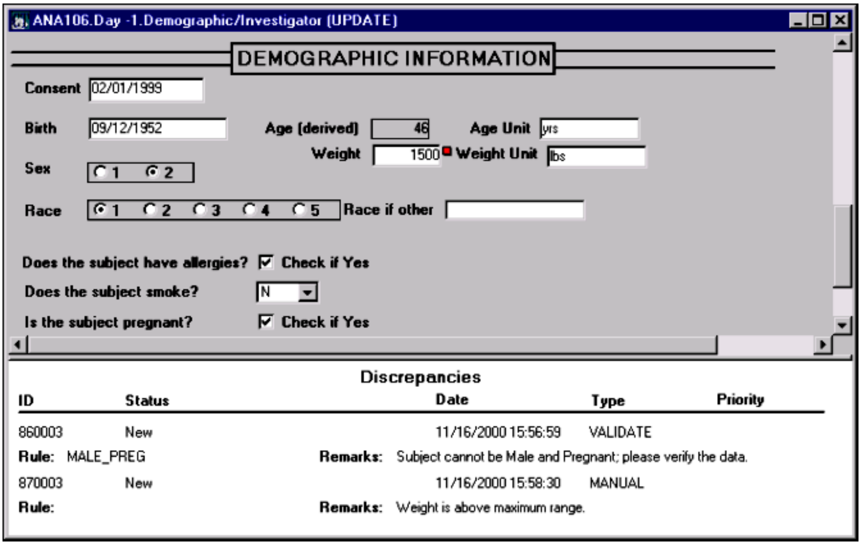
What is a discrepancy?

A *discrepancy* is a potential or actual data problem, such as inconsistent or missing data, identified either through automated checking or manual inspection. When you manually create a discrepancy, or when a discrepancy is created through automated checking, a record of that discrepancy is created in Resolve, allowing you to track the discrepancy through resolution. Deleting a discrepancy in Enter removes its record from Resolve.

Viewing discrepancies

Discrepancies must be viewed from the study page on which they were defined. All pages with discrepancies contain a ? icon.

1. From the **Discrepancies** menu, select **Show Discrepancies**, or double-click the ? icon on the study page.
2. The window is redrawn as a split screen, with the discrepancies shown in the lower half:



ANA106.Day -1.Demographic/Investigator (UPDATE)

DEMOGRAPHIC INFORMATION

Consent 02/01/1999

Birth 09/12/1952 Age (derived) 46 Age Unit yrs

Weight 1500 Weight Unit lbs

Sex ○ 1 ● 2

Race ● 1 ○ 2 ○ 3 ○ 4 ○ 5 Race if other

Does the subject have allergies? ☑ Check if Yes

Does the subject smoke? N

Is the subject pregnant? ☑ Check if Yes

Discrepancies

ID	Status	Date	Type	Priority
860003	New	11/16/2000 15:56:59	VALIDATE	
Rule: MALE_PREG		Remarks: Subject cannot be Male and Pregnant; please verify the data.		
870003	New	11/16/2000 15:58:30	MANUAL	
Rule:		Remarks: Weight is above maximum range.		

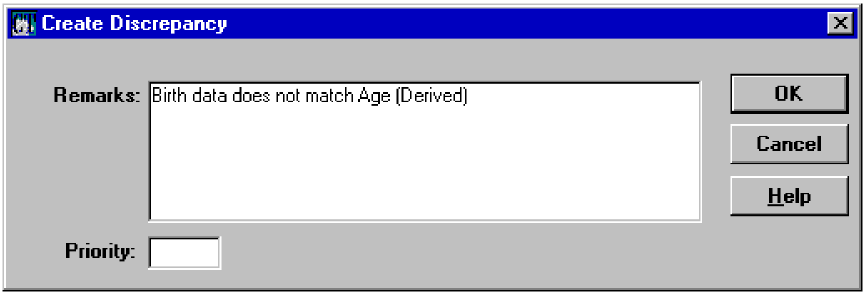
- To sort the discrepancies, from the **View** menu, select **Sort Discrepancies**.
- To filter the discrepancies, from the **View** menu, select **Filter Discrepancies**.
- To print the discrepancies, from the **File** menu, select **Print Discrepancies**.

3. To close the split screen, select **Show Discrepancies** again.

Creating a discrepancy

To create a discrepancy:

1. Place the cursor in the field for which you want to create a discrepancy.
2. From the **Discrepancies** menu, select **Create**. The Create Discrepancy dialog box opens:



Remarks: Birth data does not match Age (Derived)

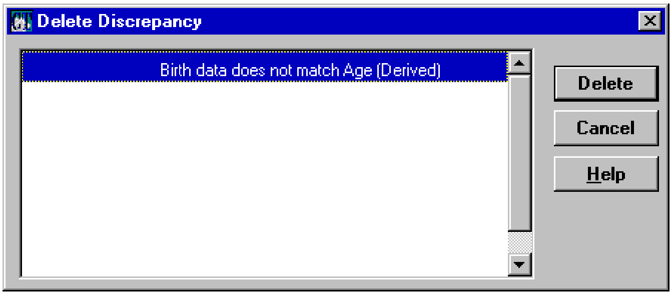
Priority:

3. In the Remarks field, enter a description of the discrepancy.
4. If required by your study, in the Priority field, you can also enter a priority ID.
5. Click **OK**. The dialog box closes.
6. Save the study page to save the discrepancy.

Deleting a discrepancy

To delete a discrepancy:

1. Open the study page on which you want to delete a discrepancy record.
2. From the **Discrepancies** menu, select **Delete**. The Delete Discrepancy dialog box opens:



Birth data does not match Age (Derived)

3. Select one or more discrepancies that you want to delete.
4. Click **Delete**. The dialog box closes.

5. Save the study page to make the deletion permanent.

Starting Resolve from Enter

You can start Resolve directly from Enter. From the **Discrepancies** menu, select **Resolve**. Resolve displays the discrepancies for the current Enter study page.

Caution: Once Resolve is started, it does not communicate with Enter. If you make changes to discrepancies in Resolve, you must refresh the study page before you can see the changes in Enter. To refresh the study page, from the **File** menu, select **Refresh**.

13 *Enter and Multisite*

Overview

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

- The Navigator
- Type 0 panels
- Attaching flags and notes in a replication environment

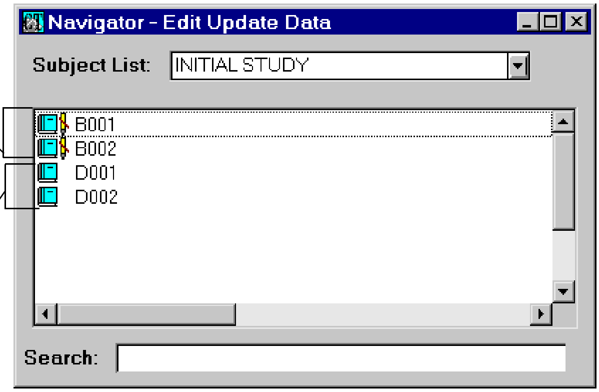
The Navigator

This section describes the following features in the Navigator that are functional only when the current protocol is in a replication environment:

- The display of subject ownership in the Navigator
- The SITE subject list
- Replication of repeating blocks and study pages

Display of subject ownership in the Navigator

The following figure shows the Navigator and indicates which subjects are owned by the current site and which subjects are owned by another site in the replication environment. Subjects that are owned by a different site have an icon to the right of the study book:



These subjects are owned by a different site.

These subjects are owned by the current site.

You can enter and edit clinical data for subjects that are owned by the current site.

You can open blocks and study pages and view clinical data for subjects that are not owned by the current site; however, you cannot modify or delete clinical data for these subjects. When you view a study page for a subject that is not owned by the current site, all menu commands used to edit or delete clinical data are unavailable.

The SITE subject list

The SITE subject list is created automatically when the current protocol is in a replication environment. When you select the SITE subject list, only subjects owned by the current site appear in the Navigator.

You cannot edit or delete the SITE subject list.

Replication of repeating blocks and study pages

The designer may set up an undetermined number of repeating blocks and repeating study pages. The data-entry operator may then create repeating blocks and repeating study pages as needed, that is, dynamically.

When the data-entry operator dynamically creates a repeating block or repeating study page, metadata to define the block or study page is created. During data entry, this metadata exists at the current site only. Then, when the clinical data in the repeating block or repeating study page replicates to other sites in the replication environment, the metadata to define the repeating block or repeating study page is created automatically at those sites.

Because it is impossible for a single site to control the number of repeating blocks and repeating study pages created when a protocol is in a replication environment, the maximum number of repeating blocks and repeating study pages set by the designer is ignored.

Type 0 panels

This section describes how Enter works with the following characteristics of Type 0 panels when the current protocol is in a replication environment:

- Ownership of records
- Ownership of observations

Ownership of records

Records in a Type 0 panel may be owned by different sites. A site originally has ownership of a record in a Type 0 panel if the record was entered at that site.

You can edit clinical data in records that are owned by the current site.

You can view clinical data in records that are not owned by the current site, but you cannot modify or delete clinical data in these records. Records that are not owned by the current site are shaded; all menu
commands used to edit or delete clinical data are unavailable.

Ownership of observations

All records that are part of an observation must be owned by the same site.

When you transfer ownership of records grouped in an observation, you must transfer ownership of all the records in an observation. For information on transferring ownership, see *Multisite*.

Attaching flags and notes in a replication environment

The following rules apply to flags and notes in a replication environment:

- You can only attach flags and notes to data owned by the current site.
- The flag or note replicates with the data.
- The flag or note cannot be modified or deleted at a site that does not own the data to which it is attached.

Enter, Resolve, and Retrieve

Resolve

Part I: Data Manager Information

Part II: Designer Information

Resolve

Resolve

14 *Introduction to Resolve*

Overview

This chapter provides an overview of Resolve's features and functions. It introduces terms and concepts used throughout this guide, and provides a usage scenario that suggests how you might incorporate this module into your work. This chapter also explains how this guide is organized to help you find the information you need.

For step-by-step instructions on any tasks described in this chapter, see the Help.

Resolve in the Clintrial software workflow

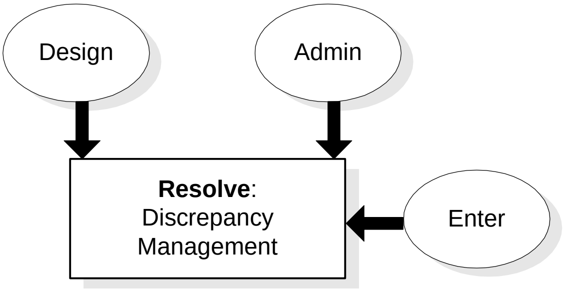
The Clintrial software Resolve module offers discrepancy management and resolution capabilities that support the work you perform in other Clintrial software modules. When this optional module is installed with the core Clintrial software modules, you can:

- Identify data discrepancies.
- Generate and track discrepancy records.
- Produce data discrepancy forms for distribution to investigators.
- Record investigation and resolution information.

In this section, figures illustrate how Resolve fits into your work process with other Clintrial software modules.

Preparing to use Resolve

This figure shows the Clintrial software modules that are involved in preparing Resolve for use and, optionally, in customizing the Resolve interface:



Preparation activities include:

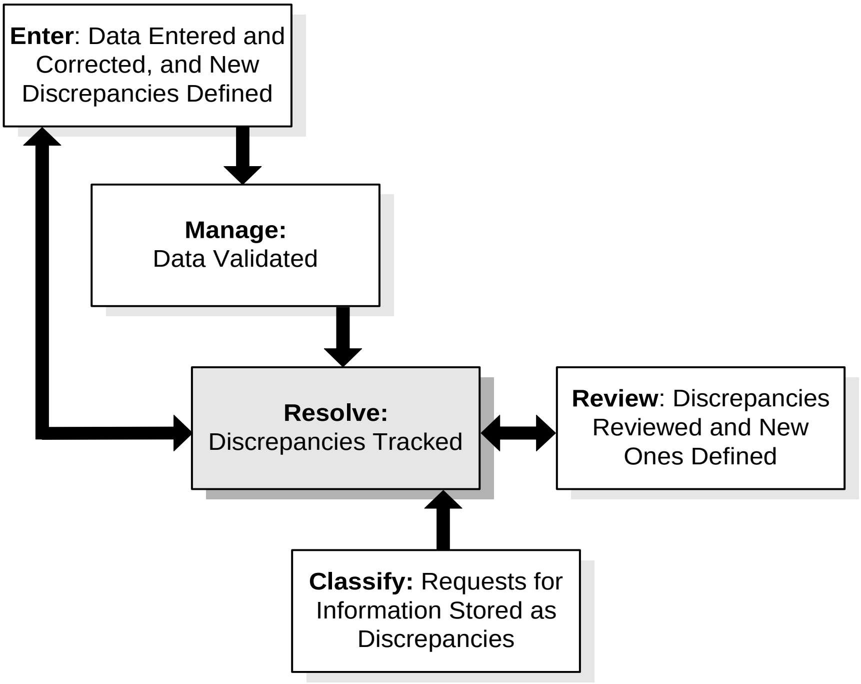
- Setting user or usergroup access rights for Resolve
- Writing rules for validation
- Creating new flag categories
- Defining protocol-level preferences

Optionally, you can customize Resolve’s features, including the set of discrepancy statuses and the format of the standard data discrepancy form. For more information, see “What are discrepancy statuses?” and “What are data discrepancy forms?” on page 157.

The responsibility for setup and customization is likely to be shared by a Designer, Clintrial software administrator, and PL/SQL programmer.

- Designers create flag categories for Resolve, define protocol parameters, and determine whether other customizations are needed, for example:
  - Displaying investigator information in discrepancies through the CTV_ONE_INVESTIGATOR protocol parameter or the CT_RESOLVE_USER package
  - Assigning reason for change values to the CTS_REASON_CODES codelist
- Clintrial software administrators use Admin to assign the Resolve access rights to users and usergroups, as well as to set default values for protocol parameters used by Resolve.
- PL/SQL programmers write rules in Design or in packages called by rules and derivations or for customizing Resolve.

This figure shows Resolve's discrepancy management role in the clinical data workflow:


Enter: Data Entered and Corrected, and New Discrepancies Defined
Manage: Data Validated
Resolve: Discrepancies Tracked
Review: Discrepancies Reviewed and New Ones Defined
Classify: Requests for Information Stored as Discrepancies

Generally, a data manager or other end user is responsible for using Resolve to track and resolve discrepancies in clinical data.

Discrepancy records

This section introduces some of the fundamental terms and concepts used in Resolve.

What are discrepancies?

As data is collected during a subject visit and entered interactively through Enter or batch loaded in Manage, discrepancies can be introduced. *Discrepancies* include both actual and potential data problems, such as missing values or variations from an expected range, and inconsistencies among data values entered over time or for different panels.

For example:

- Omissions can occur when study data is collected.
- Transcription errors can occur when data is copied to or from a case report form (CRF).
- Typographical errors can occur during data entry into the Clintrial software study database.

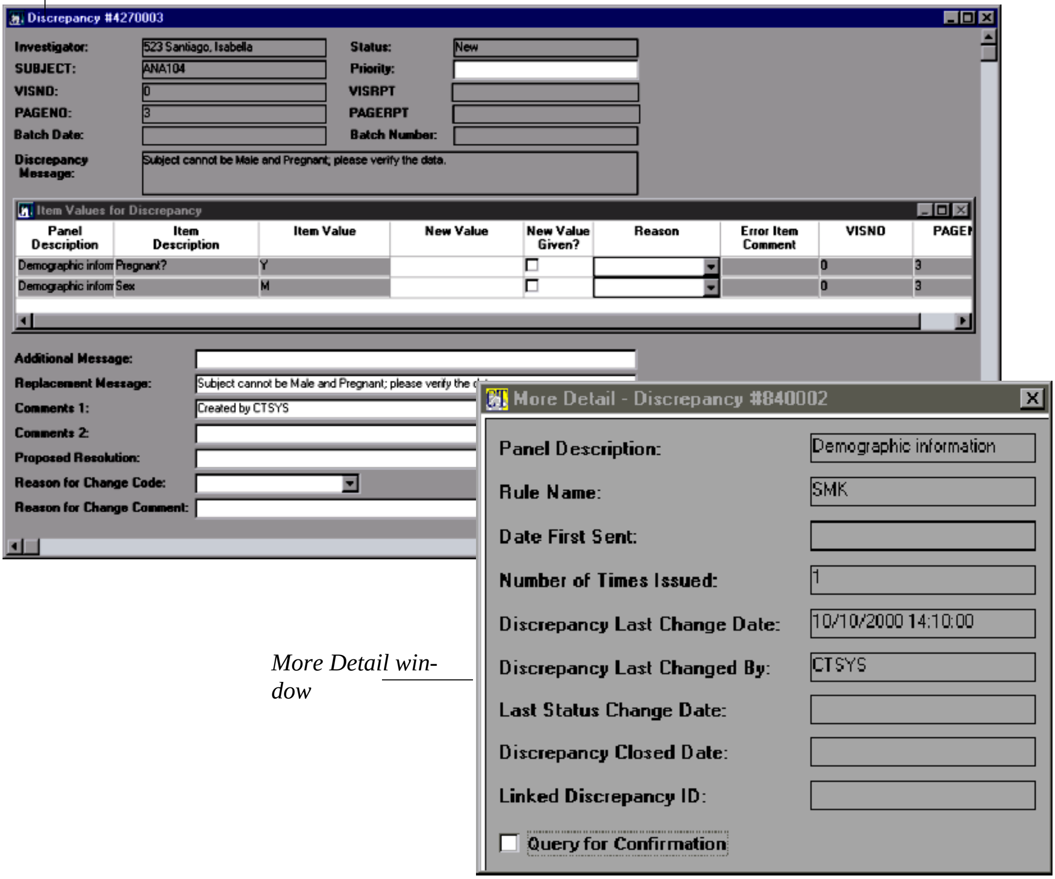
The following table lists several discrepancies noted during a single subject visit:

Discrepancy type:	**For example:**
Item value discrepancies	Blood pressure is 1222/76. Pulse is illegible.
Inconsistencies on a panel	Sex is male and pregnant is yes.
Inconsistencies between panels	Visit date (3/4/1998) in Vital Signs panel precedes birth date (9/7/1999) in Demography panel.

What are discrepancy records?

In Resolve, data discrepancies are stored in individual records. Each *discrepancy record* contains a description of an error and identifies the record that was the source of the discrepancy. A discrepancy record may also include items from any panel associated with the error. The following example shows the Resolve windows that display discrepancy record information:

Detailed Discrepancy # window



Discrepancy #4270003

Investigator:	523 Santiago, Isabella	Status:	New
SUBJECT:	ANA104	Priority:	
VISNO:	0	VISRPT	
PAGENO:	3	PAGERPT	
Batch Date:		Batch Number:	
Discrepancy Message:	Subject cannot be Male and Pregnant; please verify the data.		

Item Values for Discrepancy

Panel Description	Item Description	Item Value	New Value	New Value Given?	Reason	Error Item Comment	VISNO	PAGE[illegible]
Demographic inforn	Pregnant?	Y		□			0	3
Demographic inforn	Sex	M		□			0	3

Additional Message:	
Replacement Message:	Subject cannot be Male and Pregnant; please verify the d
Comments 1:	Created by CTSYS
Comments 2:	
Proposed Resolution:	
Reason for Change Code:	
Reason for Change Comment:	

More Detail - Discrepancy #840002

Panel Description:	Demographic information
Rule Name:	SMK
Date First Sent:	
Number of Times Issued:	1
Discrepancy Last Change Date:	10/10/2000 14:10:00
Discrepancy Last Changed By:	CTSYS
Last Status Change Date:	
Discrepancy Closed Date:	
Linked Discrepancy ID:	

□ Query for Confirmation

More Detail window

Discrepancy records store the comments, clarifications, and proposed solutions generated by research and attempts at resolution. No changes are made to the source records until all research is complete. When a satisfactory resolution is found for a discrepancy, you can update the underlying record with new values and close the corresponding discrepancy record.

How are discrepancy records created?

Resolve discrepancy records are created in the following ways:

- Manually, in Resolve and Enter.

- Automatically, when a rule returns FALSE for a record during validation in Manage
- Automatically, based on flags attached to data in Enter (rules must be present to detect these flags during validation in Manage)
- Automatically, when a record fails merging in Manage
- Automatically, when a user requires more information to solve a coding problem in Classify

For more information on how discrepancy records are created, see Chapter 15 and Chapter 21.

What are discrepancy statuses?

To help data managers track discrepancies during research, review, and resolution activities, a series of discrepancy statuses can be assigned. *Discrepancy statuses* indicate where a discrepancy record is in the resolution process. For example, the status of a newly created discrepancy record is New, and the status of a discrepancy record for which a solution has been suggested is Resolution Proposed.

Typically, a data manager uses Resolve to change the status of a discrepancy record. However, in some cases a Resolve action results in an automatic change in status.

For more information on discrepancy statuses, see "Discrepancy statuses" on page 175.

Both the discrepancy statuses supplied with Resolve and the transitions permitted between them can be customized. For more information on customizing these features, see "How to customize discrepancy statuses" on page 182.

What are data discrepancy forms?

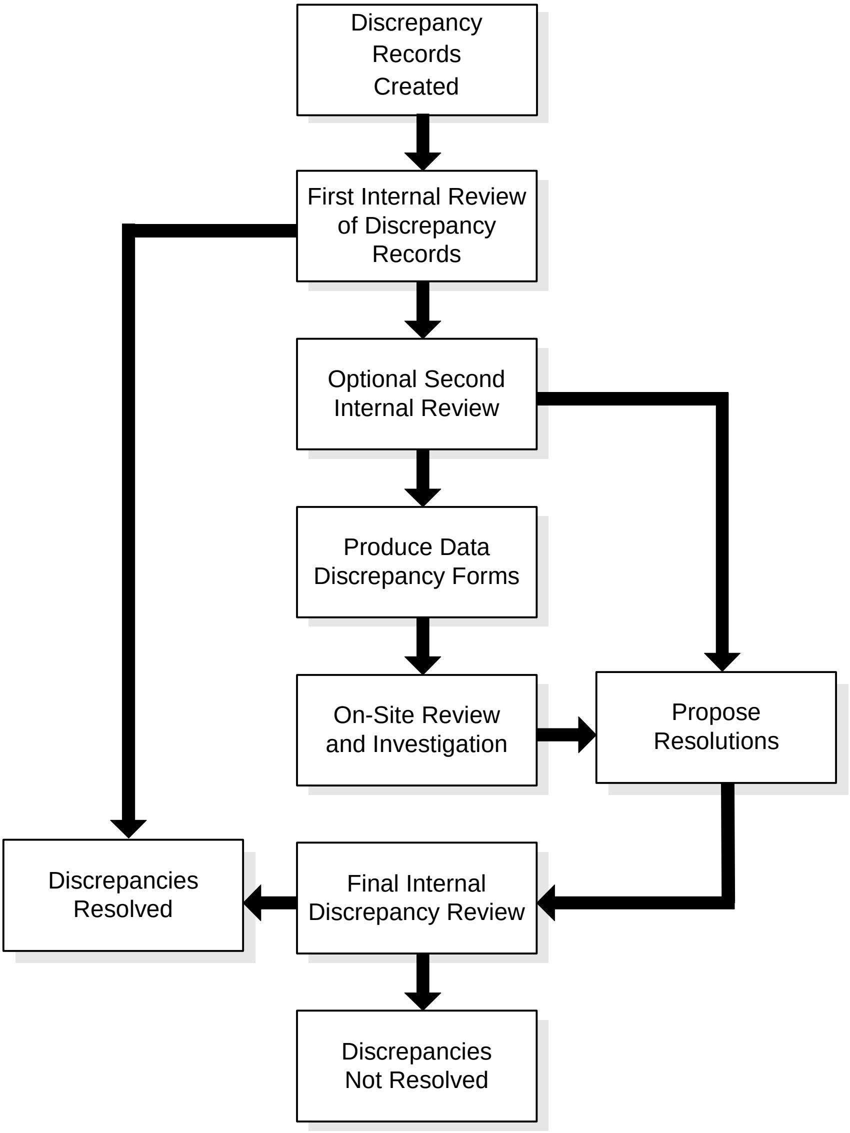
Data discrepancy forms present information about one or more discrepancy records, and provide space for proposed resolutions. These printed reports streamline communication between data managers and clinical investigators, and document the discrepancy resolution process.

For more information on data discrepancy forms, see Chapter 16.

The standard data discrepancy form supplied with Resolve can be customized. For more information on changing the format of this report, see Chapter 20.

Usage scenario

The following figure shows the workflow for discrepancy management and resolution in Resolve:


Discrepancy Records Created
First Internal Review of Discrepancy Records
Optional Second Internal Review
Produce Data Discrepancy Forms
On-Site Review and Investigation
Propose Resolutions
Discrepancies Resolved
Final Internal Discrepancy Review
Discrepancies Not Resolved

After the Designer prepares a protocol for use with Resolve, and it is set up for Resolve, discrepancy records are created for review and resolution. You can use Resolve to review all of the discrepancy records for a given protocol, or select specific subsets, such as those records relating to a particular data panel or sharing a particular discrepancy status.

During this initial review, resolutions may be discovered for some discrepancy records. For example, suppose a discrepancy record is generated for a record with a value of 1222/76 for blood pressure. By examining the case report form (CRF), you find that the blood pressure should be 122/76. To correct the discrepancy and close the discrepancy record at this point in the process, you can edit the blood pressure item value in Enter. The validation procedure runs, the rule no longer fails, and the discrepancy status changes to Autoclosed.

Other discrepancy records may need to be routed to internal experts or to the investigational site for review and resolution. You can use Resolve to change the discrepancy status of these records to indicate that another internal review is taking place (Released) or that data discrepancy forms need to be generated for distribution (Ready to Send).

As proposed resolutions are returned, the comments and new values that are supplied are added to the discrepancy records. No changes are made to the underlying records until after a final review, when the you use Resolve to apply the resolutions. This process updates the underlying records to correct the discrepancies and closes the discrepancy records by applying an appropriate discrepancy status.

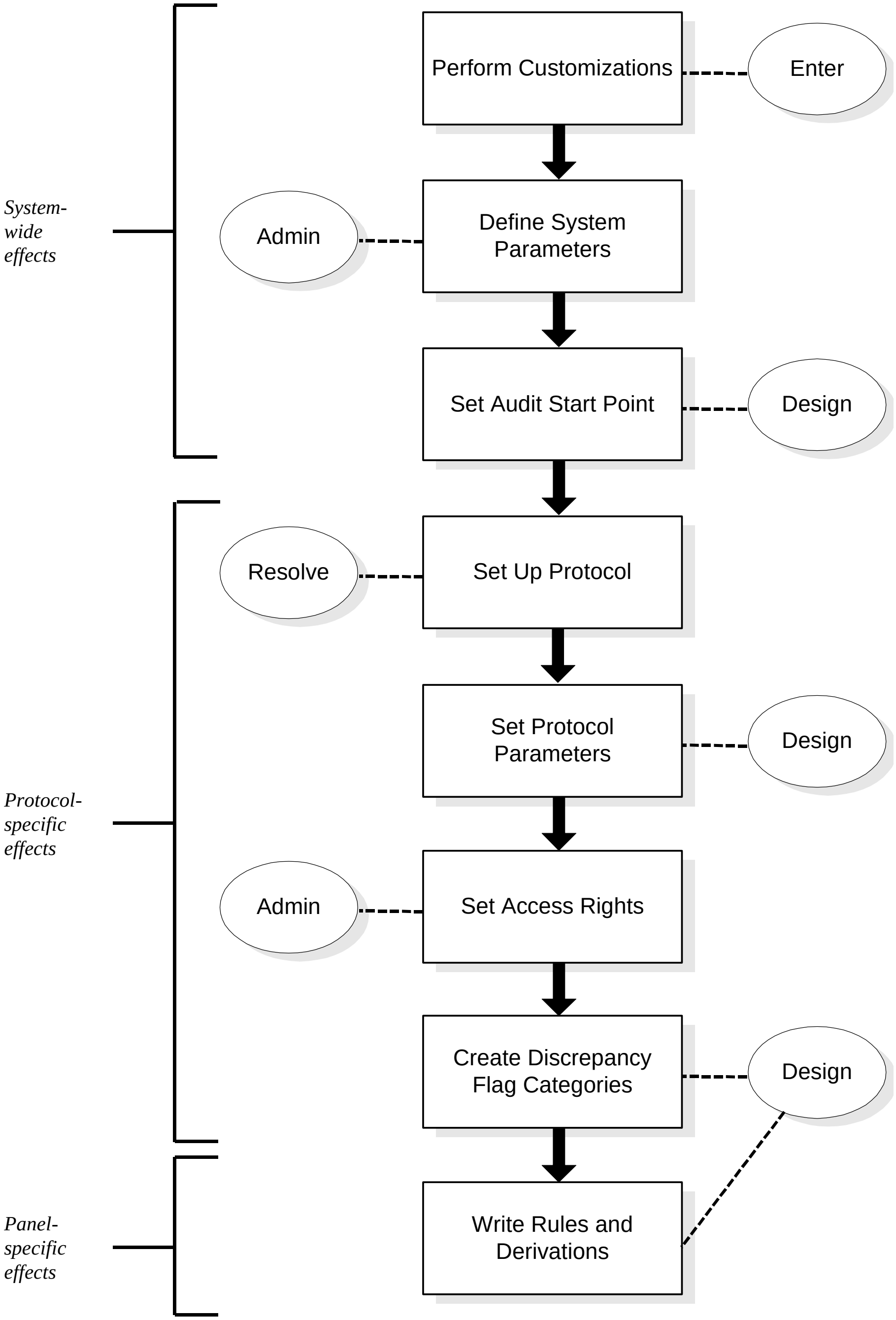
For more information on these tasks, see the chapters in Part I of *Resolve*.

Preparing and customizing Resolve

Before you can take advantage of Resolve’s discrepancy management and resolution capabilities, the Designer must perform tasks in other Clintrial software modules that prepare the Clintrial software to use Resolve. Some of these tasks are required. Optional tasks allow you to tailor Resolve to your needs and work habits.

In addition, the protocol preference value and PL/SQL procedure that retrieve an investigator for each discrepancy record can be changed.

Resolve's setup tasks can apply to the entire system, to a specific protocol, or to the panels in a protocol. The following figure shows the order in which these tasks should be performed:



Only two tasks in this flowchart are required:

- Set up protocol to use Resolve (performed in Resolve).
- Set access rights for the protocol (performed in Admin).

If you choose to customize Resolve, make your modifications before you perform protocol- and panel-specific setup activities. The following Resolve features can be customized:

- Discrepancy statuses can be changed and new statuses can be added.
- The set of transitions permitted between discrepancy statuses can be modified.
- The protocol preference value and PL/SQL procedure that retrieve an investigator for each discrepancy record can be changed.

You can also redesign the data discrepancy form at any time.

For more information on these tasks, see the chapters in Part II of this guide.

Using this guide

The remaining chapters of this guide are divided into two parts:

Part I: Data Manager Information

Part I is intended for users who are responsible for using Resolve to manage and resolve discrepancies. Before a data manager begins to use Resolve, the Design user, Clintrial software administrator, and PL/SQL programmer should complete the setup activities and customizations described in Part II. The chapters in Part I are:

Chapter 15: Managing Discrepancy Records

Chapter 16: Browsing and Printing

Chapter 17: Resolving Discrepancy Records

Chapter 18: Reporting Operational and Statistical Data

Part II: Designer Information

Part II is intended for users who perform setup activities, write rules, and customize Resolve data structures. However, before you implement any of the changes described in Part II, thorough familiarity with the Resolve functionality detailed in Part I is recommended. The chapters in Part II are:

15 Managing Discrepancy Records

Identifying discrepancies

After a protocol is set up for use with Resolve, discrepancies can be detected and added to the Resolve database of discrepancy records using any of the following methods:

- Automated checks (validation)
- Data-entry flags (attached to data in Enter, detected during validation)
- Manual entry
- Merging
- Requests for more information in Classify

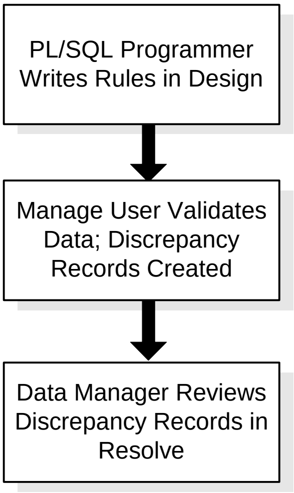
You can create discrepancy records manually in Resolve at any time. To use the other methods, which automate the creation of discrepancy records, preparatory tasks must first be completed in Manage and Design (or Classify must be installed). For example, to use automated checks to identify discrepancies and create Resolve discrepancy records, rules must be written and attached to panels in Design. Then, validation must be run to apply the rules and create new discrepancy records. Run validation directly in Manage.

The following sections describe the tasks that a data manager, a Designer or a PL/SQL programmer perform to create Resolve discrepancy records for data discrepancies.

How validating data creates discrepancy records

Designers and PL/SQL programmers can design and write rules that find missing or inconsistent values in data records after data entry is complete. 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 in Manage that validates clinical data before it is moved to the data table (merged). During validation, rules evaluate to TRUE or FALSE.

The following figure shows the steps in this process:


PL/SQL Programmer Writes Rules in Design
Manage User Validates Data; Discrepancy Records Created
Data Manager Reviews Discrepancy Records in Resolve

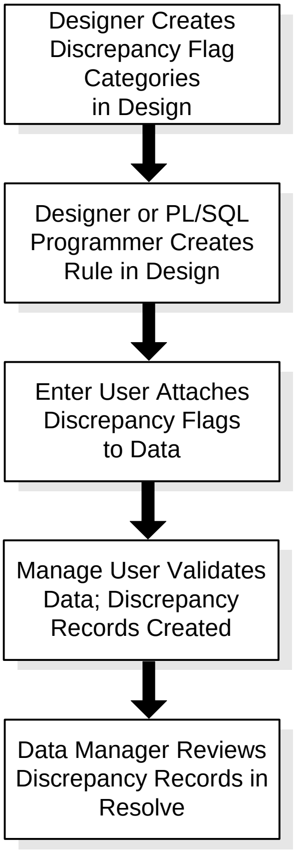
For example, a rule may be defined to compare the visit date in the Vital Signs panel to the subject's date of birth in the Demography panel. If the subject's date of birth is recorded as occurring after the visit date, a discrepancy record is generated automatically when validation runs.

For more information on writing rules for Resolve, see Chapter 21.

How to flag erroneous data in Enter

For a discrepancy noticed during data entry, a flag can be attached to data in Enter to describe the error and, if available, its possible resolution. A *flag* is an attachment to clinical data used to label and monitor data quality problems. A flag can be attached to an item, a record, or an observation. To automate the creation of discrepancy records in Resolve from flags attached in Enter, a Designer creates special discrepancy flag categories and a Designer or PL/SQL programmer creates a rule attached to the context panel to detect flags in those categories during validation.

The following figure shows the steps required to create Resolve discrepancy records from flags:

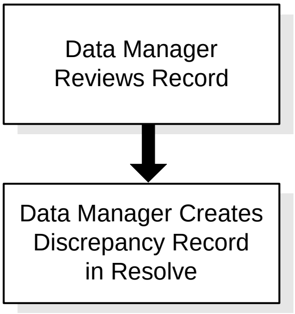

Designer Creates Discrepancy Flag Categories in Design
Designer or PL/SQL Programmer Creates Rule in Design
Enter User Attaches Discrepancy Flags to Data
Manage User Validates Data; Discrepancy Records Created
Data Manager Reviews Discrepancy Records in Resolve

For example, during data entry, a user might notice that a particular item requires a TRUE or FALSE value, but **No** appears on the case report form (CRF). As a result, the data-entry operator leaves the item blank, but attaches a flag to it with a comment describing the situation and suggesting that **No** may indicate FALSE. If the data-entry operator selects a discrepancy flag category when attaching the flag, a rule will detect the flag and create a discrepancy record in Resolve the next time validation runs.

For more information on writing rules and attaching flags, see Chapter 21.

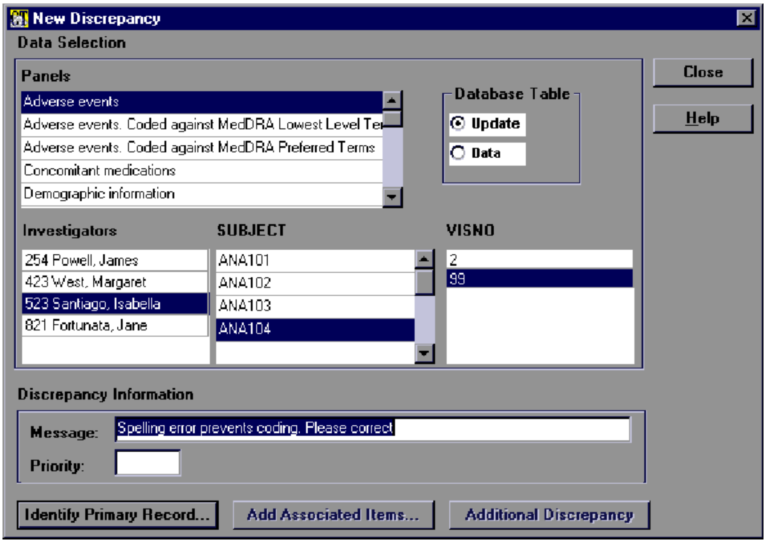
How to create discrepancy records manually in Resolve

For discrepancies observed during a review, a data manager can create new discrepancy records in Resolve. For example, you might observe that the vital signs data collected for a subject over several visits is unusually consistent, with none of the normal, expected variations. Before you investigate further, you create a new discrepancy record to document your research. The following figure shows the steps in the manual creation of discrepancies:


Data Manager Reviews Record
Data Manager Creates Discrepancy Record in Resolve

To avoid creating a duplicate discrepancy record, you can first use Resolve's viewing options to review the existing discrepancy records for similar discrepancies.

To create a new discrepancy record, you must have the Create access right in Resolve. From the **Discrepancy** menu, select **New**. The New Discrepancy dialog box opens:



Panels:
- Adverse events
- Adverse events. Coded against MedDRA Lowest Level Te
- Adverse events. Coded against MedDRA Preferred Terms
- Concomitant medications
- Demographic information

Database Table: ● Update ○ Data

Investigators	SUBJECT	VISNO
254 Powell, James	ANA101	2
423 West, Margaret	ANA102	99
523 Santiago, Isabella	ANA103	
821 Fortunata, Jane	ANA104	

Message: Spelling error prevents coding. Please correct

Priority:

In this dialog box, you select data from the available lists to help identify the source of the discrepancy. The following information about the discrepancy record's source record (primary record) is required:

- Panel
- Database Table
- Subject context item

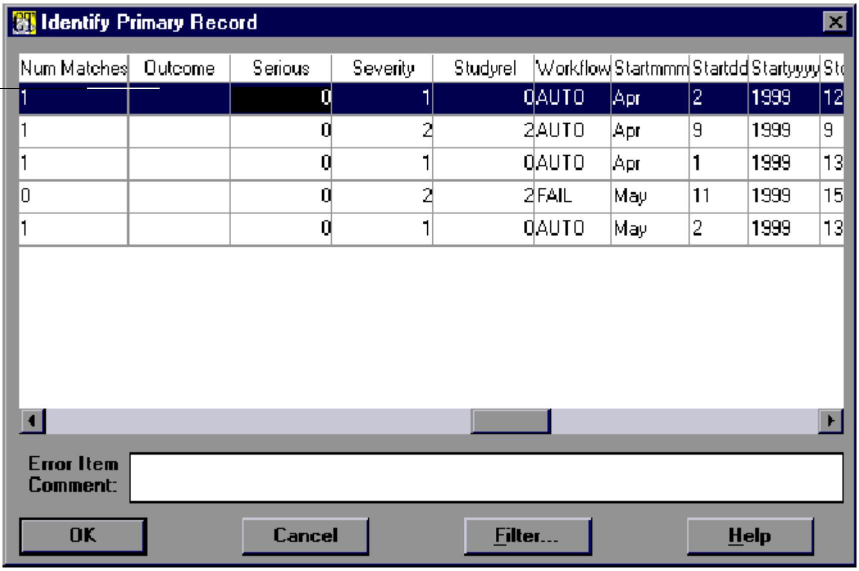
Optionally, you can select a block context item.

Note: If the panel is a Type 0 panel (non-patient data), you cannot select a subject context item or a block context item.

You must also describe the discrepancy, and optionally give the discrepancy record a numeric priority assignment.

Note: The Clintrial software administrator or Designer can also set up Resolve to display information about the investigator in the discrepancy record, various discrepancy reports, and in the Select By dialog box. For more information on displaying investigator information, see "Modifying investigator retrieval instructions" on page 256.

When these selections and entries are complete, click **Identify Primary Record**. The Identify Primary Record dialog box opens:



Primary item

Identify Primary Record

Num Matches	Outcome	Serious	Severity	Studyrel	Workflow	Startmmm	Startdd	Startyyyy	St
1		0	1	0	AUTO	Apr	2	1999	12
1		0	2	2	AUTO	Apr	9	1999	9
1		0	1	0	AUTO	Apr	1	1999	13
0		0	2	2	FAIL	May	11	1999	15
1		0	1	0	AUTO	May	2	1999	13

Error Item Comment:

This dialog box contains a list of all records that meet your specifications for Panel, Database Table, Investigator, and the subject and block context items, as selected in the New Discrepancy dialog box. After you find the record and the specific, or primary item that is the source of the discrepancy, click that item, and optionally add text to Error Item Comment field, then click **OK**. The new discrepancy record is created with information included about the specific item selected. Then, you may do one or both of the following as many times as appropriate:

- To identify other items that are relevant to your discrepancy record, click **Add Associated Items**. The Add Associated Items dialog box opens. Complete it the same way you did the Identify Primary Record dialog box.

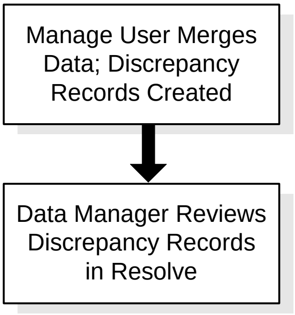
  Note: You can select associated items from either the same panel or other panels.

- To create another new discrepancy record, click **Additional Discrepancy**. The New Discrepancy dialog box opens. Define this discrepancy the same way as you did the preceding one.

Note: You can also create discrepancy records manually in Enter. For information on how to create discrepancy records in Enter, see "Working with discrepancies" on page 140.

How merging data creates discrepancy records

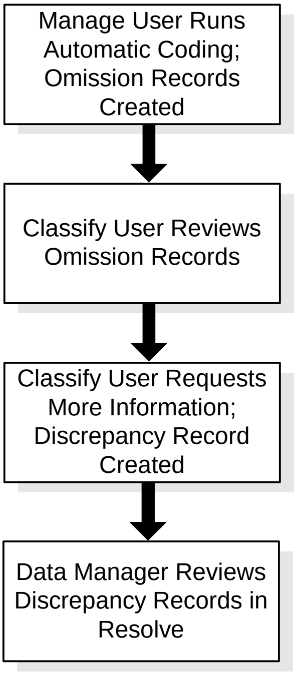
Discrepancy records are created automatically for records that fail merging in Manage. The following figure shows the steps in this process:


Manage User Merges Data; Discrepancy Records Created
Data Manager Reviews Discrepancy Records in Resolve

For example, suppose that when merging runs, two records with the same subject, block ID, and page are found. This duplication results in a merge error, and a discrepancy record is generated automatically.

How Classify creates discrepancy records

If Classify is installed, discrepancy records are created when more information is requested about a term that failed automatic coding in Manage. The following figure shows the steps in this process:


Manage User Runs Automatic Coding; Omission Records Created
Classify User Reviews Omission Records
Classify User Requests More Information; Discrepancy Record Created
Data Manager Reviews Discrepancy Records in Resolve

For example, suppose that the adverse event CHILLS W/ FEVER is recorded for a given subject. When automatic coding runs, no match is found in the coding thesaurus. An omission record is therefore created in Classify. To solve it, a coding specialist may want input from the on-site investigator. These requests are stored as discrepancy records, so that the necessary information can be solicited using data discrepancy forms.

For more information on how you can use Classify to create discrepancy records, see *Manage, Classify, and Lab Loader.*

Tracking discrepancies

The process of managing and resolving discrepancies begins after discrepancies are identified and their corresponding discrepancy records are created. Resolve helps you to keep track of discrepancy records and the discrepancies they represent as you perform the following tasks:

- Review discrepancy records and route them for resolution.
- Produce data discrepancy forms to send to investigators.
- Collect and enter information to resolve the discrepancies.

How reviewing discrepancies works

First, the data manager or person responsible for a study performs an expert review of each discrepancy record and its corresponding clinical data. This step may involve reconciling data stored in the Clintrial software with the data that appears on the original CRFs. As a result of this review, some discrepancy records may be resolved or closed, while others may require additional research.

If more research is necessary, an optional second review by clinical research associates (CRAs) or other personnel can then occur. This review provides an opportunity for additional resolutions to be proposed before requests for more information are sent to the investigational site(s).

You can add comments to a discrepancy record after each review to maintain a complete online record of suggestions, proposed resolutions, or other messages.

How producing data discrepancy forms works

Data discrepancy forms solicit research and resolution information for discrepancy records from on-site investigators. In Resolve, you group discrepancy records into batches by investigator or investigational site, then print and distribute the corresponding data discrepancy forms.

Investigators analyze the discrepancies, then enter their proposed resolutions on the data discrepancy forms and return them to the data management department.

For more information on data discrepancy forms, see Chapter 16.

How entering and evaluating resolutions works

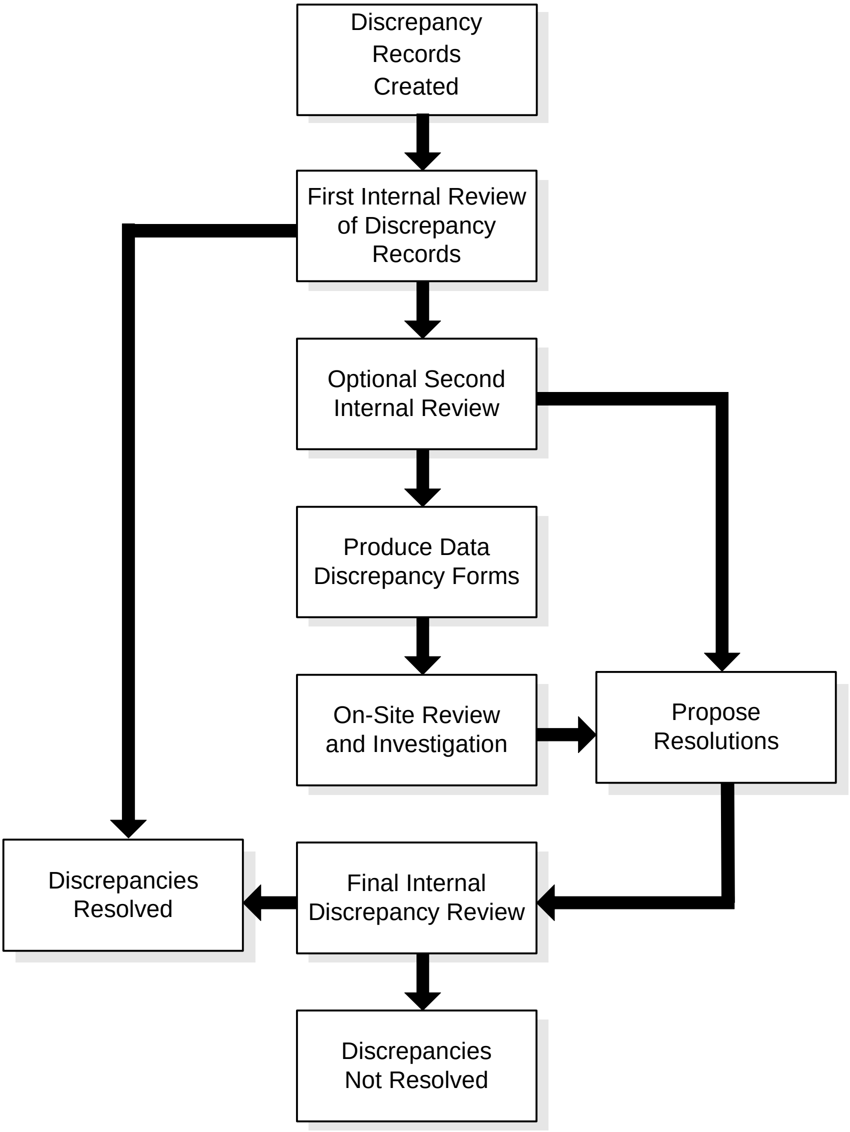
When a CRA review or an on-site investigation results in potential resolutions, descriptions of the proposed resolutions are added to the discrepancy records. You can perform a final, online review in Resolve before you apply changes to the database.

During this final review, you can do one of the following:

- Accept and implement the proposed resolutions.
- Close the discrepancy records without making the proposed changes.
- Seek further information from the CRA or investigator.

For more information on resolving discrepancy records, see Chapter 17.

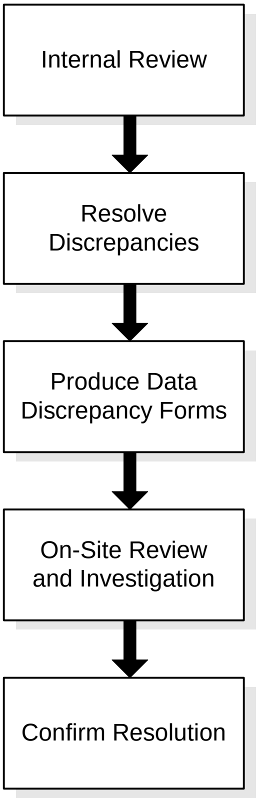
The following figure shows the workflow for discrepancy management and resolution in Resolve:


Discrepancy Records Created
First Internal Review of Discrepancy Records
Optional Second Internal Review
Produce Data Discrepancy Forms
On-Site Review and Investigation
Propose Resolutions
Discrepancies Resolved
Final Internal Discrepancy Review
Discrepancies Not Resolved

How to use Query for Confirmation

After new discrepancy records are created in Resolve, you may be able to make an educated guess about the appropriate resolution for some of them. To resolve these discrepancies immediately, you can edit the data records at once, then send out data discrepancy forms (DDFs) to investigational sites to confirm that your resolutions are correct.

The following figure shows the steps for managing and resolving discrepancies in this way:

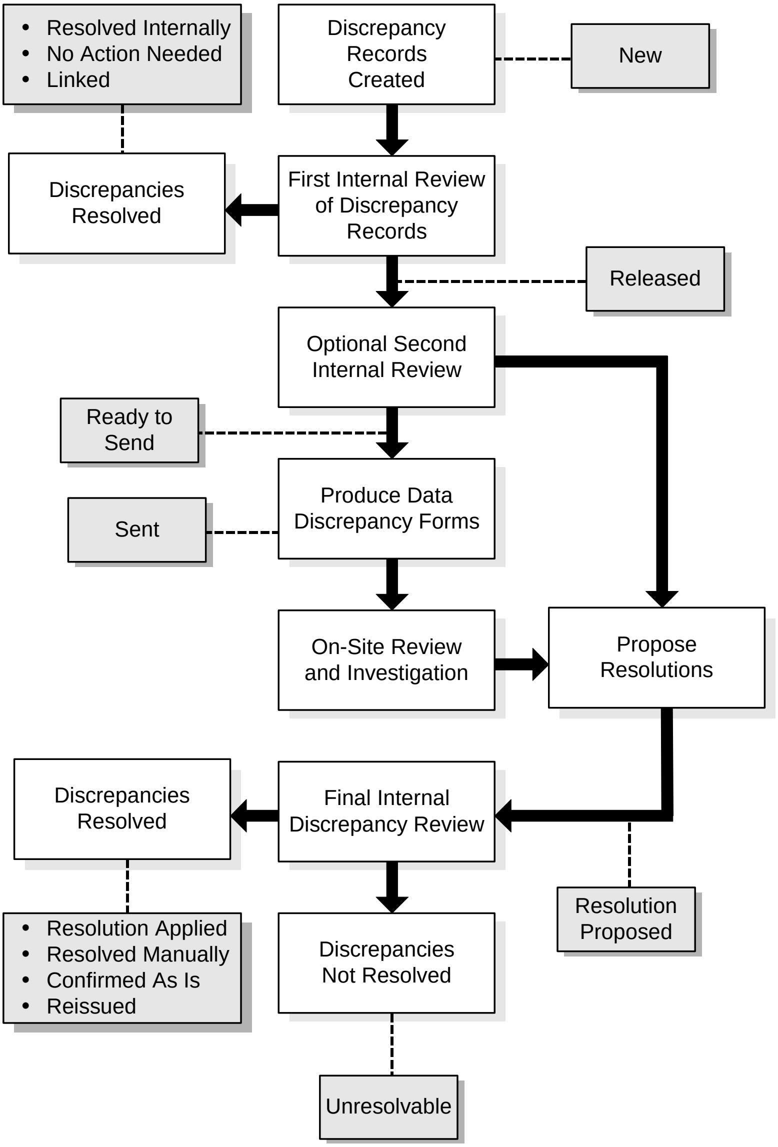

Internal Review
Resolve Discrepancies
Produce Data Discrepancy Forms
On-Site Review and Investigation
Confirm Resolution

If you use this approach, check the Query for Confirmation check box in the corresponding discrepancy records to indicate that changes have already been made to the related data records. When you print data discrepancy forms for these records, the forms include both the original and current item values, as well as a request for confirmation of your changes.

For more information on resolving discrepancy records, see Chapter 17.

Discrepancy statuses

To help you manage individual discrepancy records and work with sets of them efficiently, Resolve offers a variety of discrepancy statuses. In the following figure, the shaded elements indicate the discrepancy statuses that correspond to different points in Resolve's discrepancy management and resolution workflow:



While you are performing these Resolve tasks, users working in other Clintrial software modules can update records, rules, and panels. As a result, additional discrepancy statuses (Autoclosed and Obsolete) may be assigned automatically to discrepancy records.

Types of discrepancy statuses

The following table describes each discrepancy status:

Discrepancy status:	**Code:**	**Description:**
Autoclosed	ACD	Indicates one of the following: • The source record associated with the discrepancy record was corrected with the **Edit Source Data** command or through Enter. • The discrepancy flag was deleted. • Validation was run again, and the record failed the same rule, but with a different erroritem. In this case, the first discrepancy is autoclosed and the query for confirmation is cleared. A new discrepancy is then opened. This status, which can be assigned only by validation, closes a discrepancy record.
Confirmed As Is	CN	Indicates that, upon review, the data value is accurate as entered and the proposed resolution (if any) will not be applied. Assigning this status closes a discrepancy record.
Linked	LK	Because discrepancy records can be created by both automated and manual processes, multiple discrepancy records may refer to an identical problem. You can link one discrepancy record to another to eliminate duplication and indicate that the same action will resolve both. Assigning this status closes a discrepancy record.
New	N	In general, all discrepancy records start out with this discrepancy status. Indicates that the discrepancy record is at the beginning of the discrepancy management process.

Discrepancy status:	**Code:**	**Description:**
No Action Needed	NO	Some discrepancies, particularly those found by automated checks, may not require further action after review. Assigning this status closes a discrepancy record.
Obsolete	ACM	Indicates one of the following: • The source record associated with the discrepancy record was deleted. • The panel’s clinical data tables were deinstalled. • A rule was changed or deleted so that the record no longer failed validation. This status, which is only assigned automatically by the Clintrial software, closes a discrepancy record. ***Note:*** Complete deinstallation of the panel (metadata as well as clinical data tables), results in the deletion of any discrepancy for data in that panel.
Ready to Send	RTS	Indicates that internal reviews are complete. The next step for a discrepancy with this status is for a data discrepancy form to be printed and distributed.
Reissued	REI	If a proposed resolution does not adequately resolve a discrepancy, you can create another discrepancy record to pursue a different resolution. A reference to the new discrepancy record is stored with the original discrepancy record. The discrepancy status of the original record changes to Reissued, which closes the record. You can choose the discrepancy status for the new record: New, Released, or Ready to Send.
Released	REL	Allows for a second internal review before a data discrepancy form is generated and sent to investigators. For example, a review by a clinical research associate (CRA) might be necessary for some discrepancy records.

Discrepancy status:	Code:	Description:
Resolution Applied	RA	If a proposed resolution includes specific item value changes that can be stored in the Item Values for Discrepancy section of the discrepancy record's detailed Discrepancy # window, you can apply the changes to the referenced data records. This status is assigned automatically by selecting the **Discrepancy** menu's **Apply Proposed Values** command; it closes a discrepancy record.
Resolution Proposed	RP	Indicates that a proposal was provided by the investigator or site to which you sent the data discrepancy form, or by an associate who evaluated the discrepancy during the second internal review. This status is required if you intend to update the source records by using the **Apply Proposed Values** command. It is also required if you intend to resolve a discrepancy whose record was previously released for a second internal review.
Resolved Internally	RI	Applies to discrepancy records resolved during the first internal review, such as transcription errors in which data in the Clintrial software is different from data on the CRF. You should correct the error in the clinical data before you apply this status, which closes a discrepancy record.
Resolved Manually	RM	Indicates that the proposed resolution was accepted and the source record(s) manually resolved. Assigning this status closes a discrepancy record.
Sent	S	Indicates that a data discrepancy form was sent to the investigator or site. The discrepancy status of a discrepancy record is changed to Sent automatically when you print the record's data discrepancy form.
Unresolvable	UN	Indicates that the discrepancy record cannot be resolved. For example, a discrepancy associated with a subject who has dropped out of the study may be impossible to reconcile. Assigning this status closes a discrepancy record.

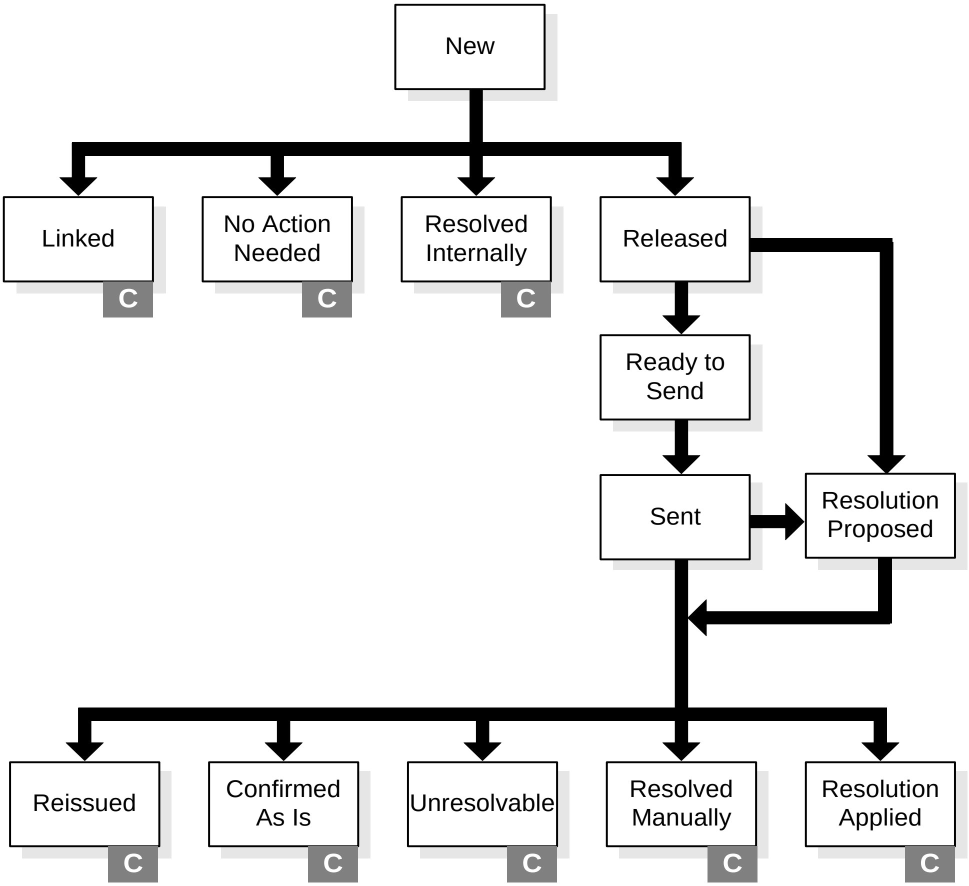
How to set the initial discrepancy status

The Designer sets the initial discrepancy status by selecting a status from the drop-down list in the rule's Discrepancy Initial Status field. If this field is left blank, the initial discrepancy status is New.

How to assign discrepancy statuses

You control the assignment of discrepancy statuses to discrepancy records. To change a discrepancy status, from the **Discrepancy** menu, select **Change Status**. To close a discrepancy record, you assign it a terminal discrepancy status, such as Resolved Internally or Confirmed As Is.

Discrepancy status assignments must adhere to a prescribed processing flow. The following figure shows the processing flow that is supplied with Resolve for discrepancy statuses. The **C** notation indicates the discrepancy statuses that close discrepancy records when they are assigned.



This processing flow is implemented and enforced by discrepancy transitions that specify which status changes are supported. For example, you can change the discrepancy status of a record from New to Resolved Internally, but you cannot change it to Resolved Manually without assigning each of the intervening statuses to it in sequence.

Note: This figure simplifies the standard processing flow. Additional discrepancy statuses and discrepancy transitions are available. A Clintrial software administrator can modify the possible transition states between statuses. For complete listings of the discrepancy statuses and discrepancy transitions supplied with Resolve, see the *Reference Guide*.

Automatically assigned discrepancy statuses

In addition, some discrepancy statuses are assigned automatically to discrepancy records as a result of an action performed in Resolve or another Clintrial software module.

The following discrepancy statuses are assigned as a result of operations performed in Resolve and cannot be set manually:

Action:	**Discrepancy status assigned:**
Manually create a discrepancy record	New
Print a batch of data discrepancy forms	Sent
Apply proposed resolution values	Resolution Applied

For example, when you print data discrepancy forms for a batch of discrepancy records, the discrepancy status of each of those records is updated automatically to Sent.

Additional discrepancy statuses are assigned as a result of actions performed using other Clintrial software modules and also cannot be set manually:

Action:	**Discrepancy status assigned:**
Source record corrected in Enter or by a Global Change in Manage Flag removed from source record in Enter Validation was run again, and the record failed the same rule, but with a different erroritem	Autoclosed
Source record deleted in Enter or by Global Delete in Manage Clinical data tables for a panel for the source record deinstalled in Design Rule changed or deleted from the panel in Design ***Note:*** Complete deinstallation of the panel (metadata as well as clinical data tables), results in the deletion of all discrepancies for data in that panel.	Obsolete

For example, when validation runs, a discrepancy record is created for a record that fails a rule. The record can be corrected manually in Enter, causing validation to run again. If the record no longer fails the rule, its corresponding discrepancy record is set to Obsolete.

Note: For some automatically closed discrepancy records, you may want to receive confirmation that the changes are indeed correct. For discrepancy records that are set to Autoclosed, you can use the **Discrepancy** menu's **Query for Confirmation** command to set this flag and change the discrepancy record's status to Ready to Send. The discrepancy record can then be included in a batch of data discrepancy forms.

How to customize discrepancy statuses

Designers can customize the discrepancy statuses and transitions that are supplied with Resolve. New discrepancy statuses and transitions can be added to those supplied with Resolve, and the existing statuses and transitions can be modified as needed.

For more information on customizing discrepancy statuses, see Chapter 20.

For more information on the discrepancy statuses and transitions supplied with Resolve, see the *Reference Guide*.

16 Browsing and Printing

Browsing discrepancy records

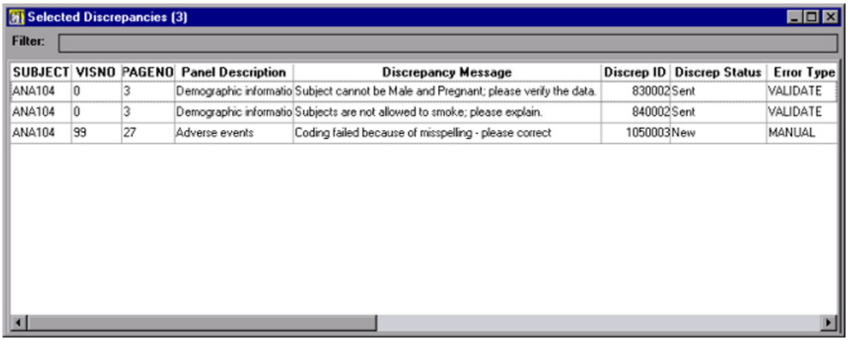
You can review discrepancy record data at two levels of detail:

- Summary (Selected Discrepancies window) — Lists groups of discrepancy records as line items in a summary window display
- Detailed (Discrepancy # window) — Shows data for a single discrepancy record

To review summary information, you choose a *view*, which selects and orders discrepancy records for display. From the summary Selected Discrepancies window, you can access detailed information about a specific discrepancy record for review and editing.

What is the summary Selected Discrepancies window?

The following example shows the summary Selected Discrepancies window with the discrepancies for subject ANA104:



SUBJECT	VISNO	PAGENO	Panel Description	Discrepancy Message	Discrep ID	Discrep Status	Error Type
ANA104	0	3	Demographic informatio	Subject cannot be Male and Pregnant; please verify the data.	830002	Sent	VALIDATE
ANA104	0	3	Demographic informatio	Subjects are not allowed to smoke; please explain.	840002	Sent	VALIDATE
ANA104	99	27	Adverse events	Coding failed because of misspelling - please correct	1050003	New	MANUAL

The following table describes the information provided in the summary Selected Discrepancies window:

Column:	Description:
subject-item-name	SUBJECT context item of a discrepancy record; for example, Subject. This column name reflects the value in the current protocol.
block-item-name	Block context item for the discrepancy record; for example, VISNO. This column name reflects the value in the current protocol.

Column:	Description:
page-item-name	Page context item of a discrepancy record; for example, PAGENO. This column name reflects the value in the current protocol.
Panel Description	Description of the panel for which inconsistent or missing data was found.
Discrepancy Message	Brief description of the discrepancy.
Discrep ID	Unique identifying number for the discrepancy record.
Discrep Status	Current discrepancy status of the discrepancy record; for example, New or Ready to Send.
Error Type	The activity taking place when the discrepancy was detected: VALIDATE, MERGE, MANUAL, FLAG, or CLASSIFY.
Date Created	The date and time the discrepancy record was created.
Priority	Numeric code used to prioritize the resolution of discrepancy records.
Rule Name	The name of the rule for records created by rules.
Batch Number	If a data discrepancy form has been printed for the discrepancy record, its batch number displays.
Batch Date	If a data discrepancy form has been printed for this discrepancy record, its batch date displays. No edits permitted.

Column:	Description:
Investigator	Information identifying the subject's investigator, if set by either of the following methods: • By default, the CTV_ONE_INVESTIGATOR protocol parameter uses a PL/SQL fragment call to the Clintrial software-supplied CT_RESOLVE_USER.INV_ID function to retrieve the subject item value for this field. To display investigator information from a different context item, in Admin or Design, edit the CTV_ONE_INVESTIGATOR protocol parameter to display investigator information from a context item. Using this method, you can edit the value so that it does not call the function, which may speed processing time. • In the CT_RESOLVE_USER package, directly edit the CT_RESOLVE_USER.INV_ID function and run it in the account specified in the Admin PROC_SITE_ACCOUNT system parameter (typically CTSITEPROC) to retrieve investigator information for display. Using this method, you can retrieve investigator information from items that are not context items. ***Note***: In Resolve, some actions require that you select an investigator from a list. In those cases, by default the CTV_SELECT_INVESTIGA protocol parameter uses a PL/SQL fragment call to the Clintrial software-supplied CT_RESOLVE_USER.INV_ID function to retrieve the subject item value. In Admin or Design, you can edit the CTV_SELECT_INVESTIGA protocol parameter to customize retrieval of investigator information for display wherever the user selects an investigator from a list. Or, if you edit the CT_RESOLVE_USER.INV_ID function and run it in the CTSITEPROC account, INV_ID supplies the value for investigator selection, as well as for the Investigator field described here. For examples, see the Field Help for that protocol parameter or "Modifying investigator retrieval instructions" on page 256.
Link Group ID	Identifies the linked record for any record that is linked. For an unlinked record, its record number (Discrep ID) displays.

Column:	Description:
block-repeat-item-name	Block repeat context item, if one exists, for the discrepancy record; for example, VISRPT. This column name reflects the value in the current protocol.
page-repeat-item-name	Page repeat context item, if one exists, for the discrepancy record; for example, PAGERPT. This column name reflects the value in the current protocol.
Source Ct_Recid	The unique ID number of the primary data record associated with the discrepancy record.

How to select views

You can display all of a protocol’s discrepancy records, or limit the summary Selected Discrepancies window to records that share certain characteristics. For example, you can review discrepancy records with a single discrepancy status, such as Resolution Proposed, or those associated with a particular investigator.

To help you specify which discrepancy records you want to examine, the **View** menu offers the following commands:

- **All** displays all discrepancy records on file for the protocol.
- **New Discrepancies** displays only discrepancy records with a discrepancy status of New.

  Note: This command does not display a new discrepancy with a status other than New; for example new discrepancies generated by a rule whose Discrepancy Initial Status is set to Sent. To display new discrepancies with a status other than New, from the **View** menu, select **Recently Changed**, or from the **View** menu, select **All**
- **Recently Changed** displays only discrepancy records that you have edited or added within the last two calendar days.
- **By Batch** prompts you for a single data discrepancy form batch number, and then displays the discrepancy records in that batch.
- **Select By** prompts you for one or more selection parameters, and then displays the discrepancy records that satisfy your criteria. You can select by page or by panel, depending on the setting in Manage of the Select_By_Page user preference. The default is by panel.

After you choose the initial set of records for the summary Selected Discrepancies window, you can:

- Focus on a subset of the available records using filters.

- Rearrange the records by modifying the sort specifications.
- Change the order and width of the summary Selected Discrepancies window's columns.

Note: If you choose Link Group ID as the basis for sorting, linked discrepancies are listed contiguously, to allow for the examination of sets of related records.

For more information on views, filtering, sorting, and the layout revision options, see the Help.

How to set a startup view

The summary Selected Discrepancies window's column layout, sort criteria, and filter string are automatically saved and reused the next time you start Resolve, but the view that displays in the Summary Selected Discrepancies window reflects the setting of the CTV_STARTUP_VIEW user preference. By default, this preference is set to ALL.

To set a different startup view, from the **File** menu, select **Preferences**. Then select the Resolve tab, and set CTV_STARTUP_VIEW to one of the other supported options. Your change will take effect the next time you start up Resolve.

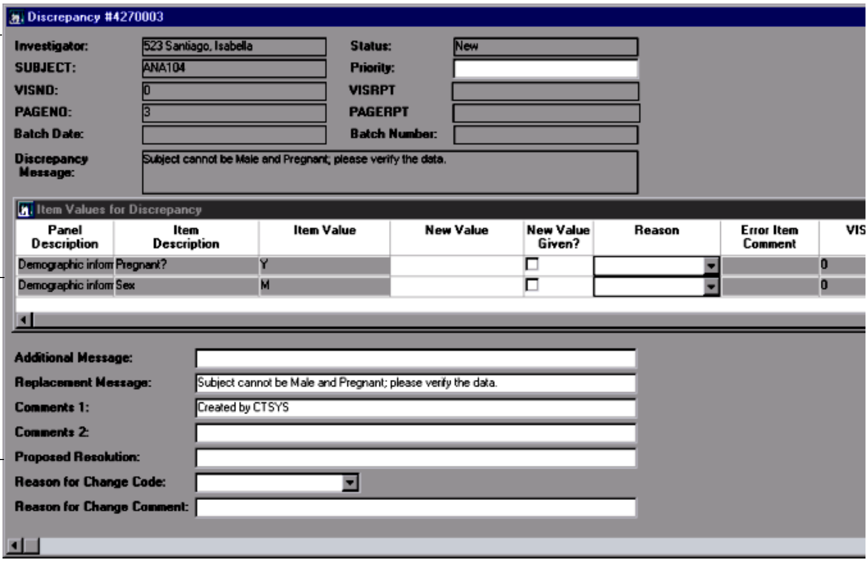
Note: If you select either **By Batch** or **Select By**, you will be prompted for a batch number or selection criteria before the summary Selected Discrepancies window displays.

What is the detailed Discrepancy # window?

The detailed Discrepancy # window provides additional information about a single discrepancy record. Open it using any of these methods:

- In the summary Selected Discrepancies window, click a row. From the **Discrepancy** menu, select **Open**.
- In the summary Selected Discrepancies window, double-click a row.
- From the **Discrepancy** menu, select **Open By Number**. When prompted, enter one discrepancy record's unique identifier (Discrep ID).

The following example shows a detailed Discrepancy # window:



Discrepancy information

Item Values for Discrepancy section

Discrepancy information

Discrepancy #4270003

Investigator:	523 Santiago, Isabella	Status:	New
SUBJECT:	ANA104	Priority:	
VISNO:	0	VISRPT	
PAGENO:	3	PAGERPT	
Batch Date:		Batch Number:	

Discrepancy Message: Subject cannot be Male and Pregnant; please verify the data.

Item Values for Discrepancy

Panel Description	Item Description	Item Value	New Value	New Value Given?	Reason	Error Item Comment	VIS
Demographic inform	Pregnant?	Y		□			0
Demographic inform	Sex	M		□			0

Additional Message:

Replacement Message: Subject cannot be Male and Pregnant; please verify the data.

Comments 1: Created by CTSYS

Comments 2:

Proposed Resolution:

Reason for Change Code:

Reason for Change Comment:

When this window is active, you can review the discrepancy record's existing information, and store proposed values for future application to data records. The following table describes the information provided in the detailed Discrepancy # window:

Field:	**Description:**
Investigator	Information identifying the subject's investigator, if set by either of the following methods: • By default, the CTV_ONE_INVESTIGATOR protocol parameter uses a PL/SQL fragment call to the Clintrial software-supplied CT_RESOLVE_USER.INV_ID function to retrieve the subject item value for this field. To display investigator information from a different context item, in Admin or Design, edit the CTV_ONE_INVESTIGATOR protocol parameter to display investigator information from a context item. Using this method, you can edit the value so that it does not call the function, which may speed processing time. • In the CT_RESOLVE_USER package, directly edit the CT_RESOLVE_USER.INV_ID function and run it in the account specified in the Admin PROC_SITE_ACCOUNT system parameter (typically CTSITEPROC) to retrieve investigator information for display. Using this method, you can retrieve investigator information from items that are not context items. ***Note***: In Resolve, some actions require that you select an investigator from a list. In those cases, by default the CTV_SELECT_INVESTIGA protocol parameter uses a PL/SQL fragment call to the Clintrial software-supplied CT_RESOLVE_USER.INV_ID function to retrieve the subject item value. In Admin or Design, you can edit the CTV_SELECT_INVESTIGA protocol parameter to customize retrieval of investigator information for display wherever the user selects an investigator from a list. Or, if you edit the CT_RESOLVE_USER.INV_ID function and run it in the CTSITEPROC account, INV_ID supplies the value for investigator selection, as well as for the Investigator field described here. For examples, see the Field Help for that protocol parameter or "Modifying investigator retrieval instructions" on page 256.
subject-item-name	SUBJECT context item of a discrepancy record; for example, Subject. This column name reflects the value in the current protocol.

Field:	**Description:**
block-item-name	Block context item for the discrepancy record; for example, VISNO. This column name reflects the value in the current protocol.
page-item-name	Page context item of a discrepancy record; for example, PAGENO. This column name reflects the value in the current protocol.
Status	Current discrepancy status of this record. To change the discrepancy status, from the **Discrepancy** menu, select **Change Status**.
Priority	Number used to prioritize the resolution of this discrepancy record. No edits permitted.
block-repeat-item-name	Block repeat context item, if one exists, for the discrepancy record; for example, VISRPT. This column name reflects the value in the current protocol.
page-repeat-item-name	Page repeat context item, if one exists, for the discrepancy record; for example, PAGERPT. This column name reflects the value in the current protocol.
Batch Date	If a data discrepancy form has been printed for this discrepancy record, its batch date displays. No edits permitted.
Batch Number	If a data discrepancy form has been printed for this discrepancy record, its batch number displays. No edits permitted.
Discrepancy Message	Description of the discrepancy supplied when this record was created. No edits permitted.
Additional Message	Appended to the Discrepancy Message or Replacement Message on data discrepancy forms.
Replacement Message	Replaces the Discrepancy Message on the data discrepancy form. When a discrepancy is created, the discrepancy message is written both to this field and to the Discrepancy Message field. This field is editable.

Field:	**Description:**
Comments 1	Internal comments from the first discrepancy review. Does not appear on data discrepancy forms. By default, the user name of the person who created the record. Only users with Resolve’s Manage access right can edit this item.
Comments 2	Internal comments from the second discrepancy review. Does not appear on data discrepancy forms. Only users with Resolve’s Propose access right can edit this item.
Proposed Resolution	Description of a proposed resolution.
Reason for Change Code	Text describing the resolution or a code from the CTS_REASON_CODES codelist.
Reason for Change Comment	Text explaining unusual resolutions.

This window’s Item Values for Discrepancy section displays specific items from clinical data records that contain data relevant to the discrepancy. For example, if a discrepancy record was created because of missing data for an item, that particular item could be included in this section. Items display in this section of the detailed Discrepancy # window if a rule created in Design has a corresponding derivation that retrieves item values from related records, or if this information was added to a record that was created manually.

The following table describes the columns in the Item Values for Discrepancy section in the detailed Discrepancy # window:

Column:	**Description:**
Panel Description	Description of a panel that stores data relevant to this discrepancy record. No edits permitted.
Item Description	Description of a specific item in the panel. No edits permitted.
Item Value	Value stored in the original data record for this item at the time the discrepancy record was created. No edits permitted.

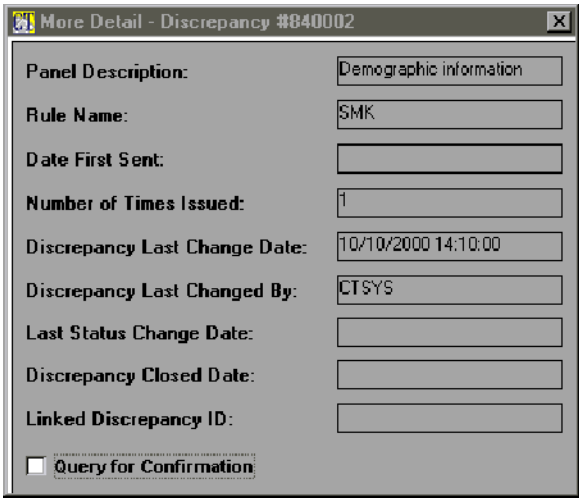
Column:	**Description:**
New Value	Proposed value for this item. Entering a new value does not change clinical data. To save the new value to this field, you must also check the New Value Given? field. For more information on applying proposed values to records, see Chapter 17.
New Value Given?	A check indicates that a new value has been entered and initiates edit checks on the new value. You must check this field to save the new or changed value, including a null value. Then you can use Apply Proposed Value to change the source data.
Reason	Descriptive text describing the reason for this change or a code from the CTS_REASON_CODES codelist. Required if New Value Given is checked.
Error Item Comment	Identifying contextual information for this item value; supplied by a rule or during manual entry. No edits permitted.
block-item-name	Value stored for the block context item of the record; for example, VISNO. No edits permitted.
page-item-name	Value stored for the page context item of the record; for example PAGENO. No edits permitted.

Note: If your access right is Create, you can add associated items to or delete them from manually created records using the **Discrepancy** menu's **Error Item** >> **Add** or **Delete** command. **Delete** is enabled only if your access level is Full and there are preexisting items. **Add** is available if your access level is either Full or No Delete.

For more information on creating rules and derivations to display information in the Item Values for Discrepancy section, see Chapter 21.

What is the More Detail window?

You can view a second window with additional discrepancy record information. First, open the detailed Discrepancy # window for a discrepancy record. Then, from the **Discrepancy** menu, select **More Detail**. The following example shows the More Detail window:



More Detail - Discrepancy #840002

Panel Description: Demographic information

Rule Name: SMK

Date First Sent:

Number of Times Issued: 1

Discrepancy Last Change Date: 10/10/2000 14:10:00

Discrepancy Last Changed By: CTSYS

Last Status Change Date:

Discrepancy Closed Date:

Linked Discrepancy ID:

□ Query for Confirmation

The following table describes the information provided in the More Detail window:

Field:	**Description:**
Panel Description	Description of the panel for which inconsistent or missing data was found. No edits permitted. ***Note:*** Displays as null for panels that do not have a panel description.
Rule Name	Stores the name of the rule (for discrepancy records created by a rule). No edits permitted.
Date First Sent	If a data discrepancy form has been printed for this discrepancy record, or for a previous discrepancy record that was reissued to create this discrepancy, displays its batch date. No edits permitted.

Field:	**Description:**
Number of Times Issued	For discrepancy records created by the reissue of another discrepancy record, shows the number of times the discrepancy has cycled through the Resolve workflow. No edits permitted.
Discrepancy Last Change Date	Date and time this discrepancy record was last modified. No edits permitted.
Discrepancy Last Changed By	User name of the person who last modified this discrepancy record. No edits permitted.
Last Status Change Date	Date and time that the discrepancy status of this discrepancy record was last modified. No edits permitted.
Discrepancy Closed Date	Date and time a terminal discrepancy status was applied to this discrepancy record. No edits permitted.
Linked Discrepancy ID	Applies to discrepancy records with the Linked or Reissued discrepancy status. Stores the discrepancy ID of the related discrepancy record. No edits permitted.
Query for Confirmation	If checked, indicates that modifications were already made to an associated record in Enter to resolve the discrepancy. Only a confirmation of the change is needed from the investigator. Edits permitted only for users with Resolve's Manage access right, and only for discrepancy records that are not closed. For more information on using this approach to resolving discrepancies, see Chapter 15. ***Note:*** The **Discrepancy** menu's **Query for Confirmation** command checks this check box and also changes the discrepancy status to Ready to Send.

How to print discrepancy information

To aid your research or for future reference, you can print the data that appears in the current summary Selected Discrepancies or detailed Discrepancy # window.

To send information in the active summary Selected Discrepancies window or detailed Discrepancy # window to a printer, from the **File** menu, select **Print**, as follows:

- From a summary Selected Discrepancies window, the **Print** command produces a list of the discrepancy records selected by your view specifications.
- From a detailed Discrepancy # window, the **Print** command produces a detailed report of the information on file for that discrepancy, including the data in the More Detail window.

To send information in the active summary Selected Discrepancies window, including the data in the More Detailed window, to a printer, from the **File** menu, select **Print Detail**.

Printing data discrepancy forms

A *data discrepancy form* is a concise, formatted report that describes a data error. The purpose of this form is to solicit information from experts, such as on-site investigators or clinical research associates (CRAs), to help you correct erroneous data. Data discrepancy forms provide space for the investigator or CRA to propose a resolution for each discrepancy record.

Note: Clintrial prints what can fit on a page, as determined by paper size, page orientation (landscape vs. portrait), etc.

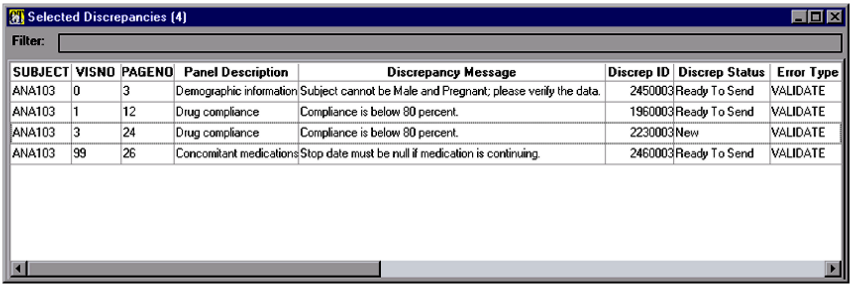
Note: Data discrepancy forms may be printed on A4 paper by using your printer's Scaling or Custom Page Size settings.

How to select records for a batch

A *batch* can consist of one or more discrepancy records. After accessing the summary Selected Discrepancies window, you can generate data discrepancy forms for batches of discrepancy records that:

- Display in the active window.
- Share the Ready to Send discrepancy status.

In the following example, only those discrepancy records with the Ready to Send status are included in a batch:



SUBJECT	VISNO	PAGENO	Panel Description	Discrepancy Message	Discrep ID	Discrep Status	Error Type
ANA103	0	3	Demographic information	Subject cannot be Male and Pregnant; please verify the data.	2450003	Ready To Send	VALIDATE
ANA103	1	12	Drug compliance	Compliance is below 80 percent.	1960003	Ready To Send	VALIDATE
ANA103	3	24	Drug compliance	Compliance is below 80 percent.	2230003	New	VALIDATE
ANA103	99	26	Concomitant medications	Stop date must be null if medication is continuing.	2460003	Ready To Send	VALIDATE

To select records for a batch, you use specific **View** menu commands to limit the discrepancy records that display in the active summary Selected Discrepancies window. For example, you can use the **Select By** command to select only records associated with a particular investigator, or records for a specific subject. All of the discrepancy records that satisfy your view criteria display in a summary Selected Discrepancies window.

When the summary Selected Discrepancies window opens, review the resulting set of discrepancy records to ensure that your selection criteria are adequate. To specify additional criteria that further limits the number of discrepancy records in the batch, from the **View** menu, select **Filter**.

How to print a batch

After you define the batch of discrepancy records, you can print data discrepancy forms. From the **Report** menu, select **Produce Form** >> **Print Batch**. The results are as follows:

- Data discrepancy forms are printed for all records in the summary Selected Discrepancies window whose status is Ready to Send.
- An identifying batch number is assigned.
- The discrepancy status of every record in the batch is changed to Sent.

What is the standard data discrepancy form?

A standard format for data discrepancy forms is provided with Resolve. This format displays:

- Identifying information for the discrepancy record’s primary data record
- The text of the discrepancy message

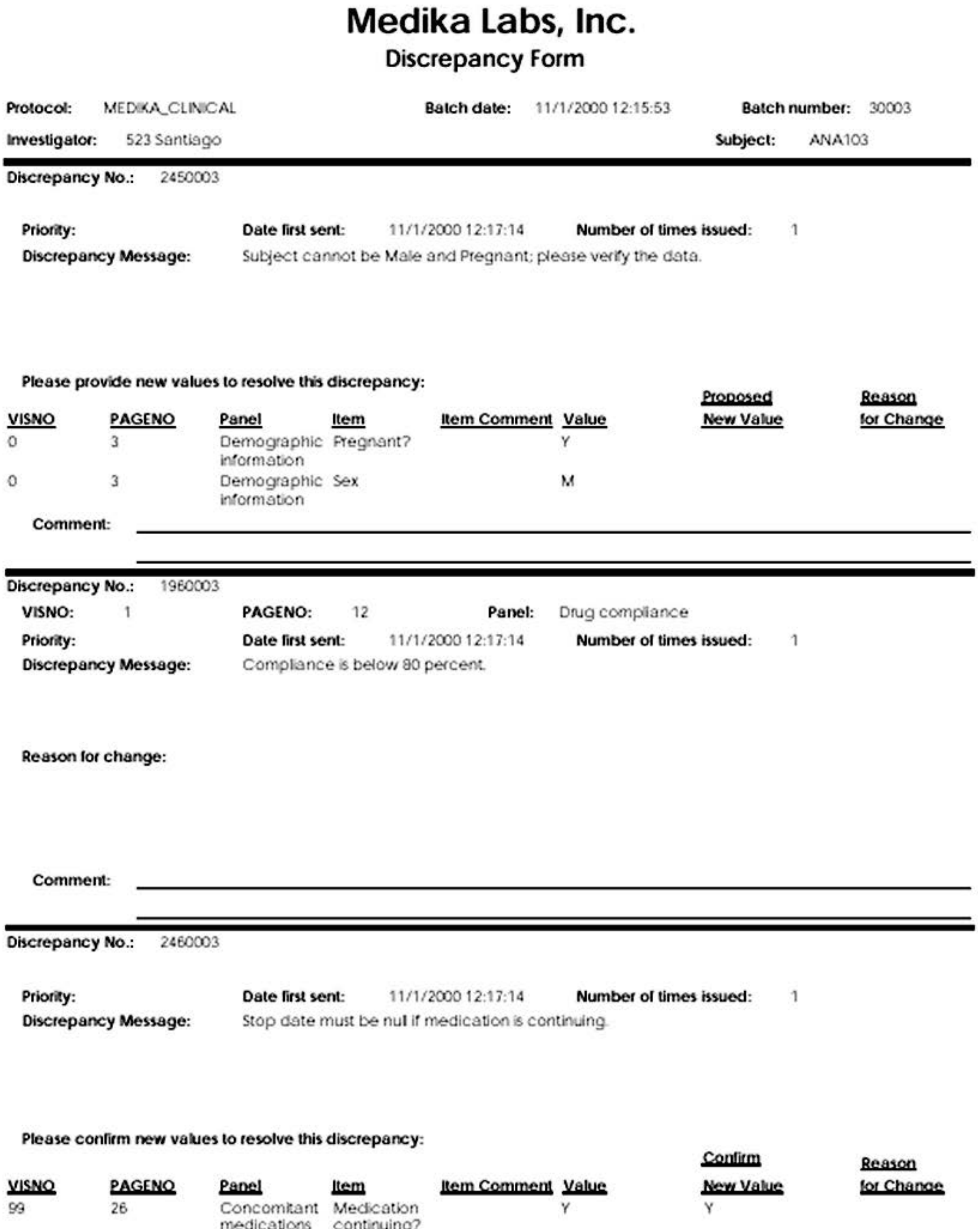
Note: If the detailed Discrepancy # window's Replacement Message field contains text that differs from the Discrepancy message text, the replacement text appears on the data discrepancy form in place of the original discrepancy text.

- Data for the items associated with the discrepancy record
- Proposed new values (if present) for the items associated with the discrepancy record

By default, this data discrepancy form displays values instead of codes for codelist items, and panel and item descriptions instead of names. To display codes rather than values, change the value of the user parameter CTV_ENTER_CODES to Yes. If more than one discrepancy record pertaining to a single subject is included in a batch, the standard form groups them onto one study page.

Note: The value in the Query for Confirmation check box affects the display of data on a data discrepancy form. If Query for Confirmation is checked, then both the original data value and the modified, current data value in the source data are included on the data discrepancy form with a request for confirmation. In this case, any values entered into the New Value field of the Item Values for Discrepancy section of the detailed Discrepancy # window are not displayed on the data discrepancy form.

The following is an example of a data discrepancy form. The Query for Confirmation check box was checked only for the last of the three discrepancy records on this form, resulting in the request for new values:



Medika Labs, Inc.

Discrepancy Form

Protocol: MEDIKA_CLINICAL **Batch date:** 11/1/2000 12:15:53 **Batch number:** 30003

Investigator: 523 Santiago **Subject:** ANA103

Discrepancy No.: 2450003

Priority: **Date first sent:** 11/1/2000 12:17:14 **Number of times issued:** 1

Discrepancy Message: Subject cannot be Male and Pregnant; please verify the data.

Please provide new values to resolve this discrepancy:

VISNO	PAGENO	Panel	Item	Item Comment	Value	Proposed New Value	Reason for Change
0	3	Demographic information	Pregnant?		Y		
0	3	Demographic information	Sex		M		

Comment: ____________________

Discrepancy No.: 1960003

VISNO: 1 **PAGENO:** 12 **Panel:** Drug compliance

Priority: **Date first sent:** 11/1/2000 12:17:14 **Number of times issued:** 1

Discrepancy Message: Compliance is below 80 percent.

Reason for change:

Comment: ____________________

Discrepancy No.: 2460003

Priority: **Date first sent:** 11/1/2000 12:17:14 **Number of times issued:** 1

Discrepancy Message: Stop date must be null if medication is continuing.

Please confirm new values to resolve this discrepancy:

VISNO	PAGENO	Panel	Item	Item Comment	Value	Confirm New Value	Reason for Change
99	26	Concomitant medications	Medication continuing?		Y	Y	

Note: The data discrepancy forms created by the submenu command **Produce Form >> Print Batch** print only records whose status is Ready to Send. To print similar reports for discrepancy records in any status and at any point in the discrepancy management and resolution workflow, from the **File** menu, select **Print Detail**. This command reports discrepancy record information in a format that is similar to that of the data discrepancy form. Printing this report for a discrepancy record does not change its discrepancy status.

How to cancel a batch

After data discrepancy forms print, you can cancel a batch. From the **Report** menu, select **Produce Form >> Cancel Batch**. When you cancel a batch, the discrepancy status of each record with Sent status reverts to Ready to Send. These discrepancy records can be included the next time you are ready to produce data discrepancy forms.

For more information on how to cancel a batch, see the Help.

How to reprint forms

You can reprint the data discrepancy forms for a previously printed batch. From the **Report** menu, select **Produce Form >> Reprint Batch**. Data discrepancy forms print for all the discrepancy records in the selected batch. When you reprint data discrepancy forms:

- The same batch number is used (no new batch number is assigned).
- No change is made to the discrepancy status of any discrepancy record in the batch.

You can select **Produce Form >> Reprint Batch** to regenerate a set of data discrepancy forms in the event of a printing problem, or if an incomplete batch was delivered to the investigator.

For more information on how to reprint data discrepancy forms, see the Help.

How to customize forms

You can customize the appearance of data discrepancy forms. All users can set user preferences to include or exclude header and requester data. In addition, the designer can set protocol preferences for Resolve, or reformat the data discrepancy form to meet your company’s needs. For example, the layout of the standard data discrepancy form might be altered so that only one discrepancy ever displays on a study page.

For more information on customizing Resolve, see Chapter 20.

17 *Resolving Discrepancy Records*

Discrepancy resolution

This chapter describes the activities you perform in Resolve after:

- Resolutions are found during a first internal review.
- Data discrepancy forms are returned from on-site investigators or other expert reviewers.

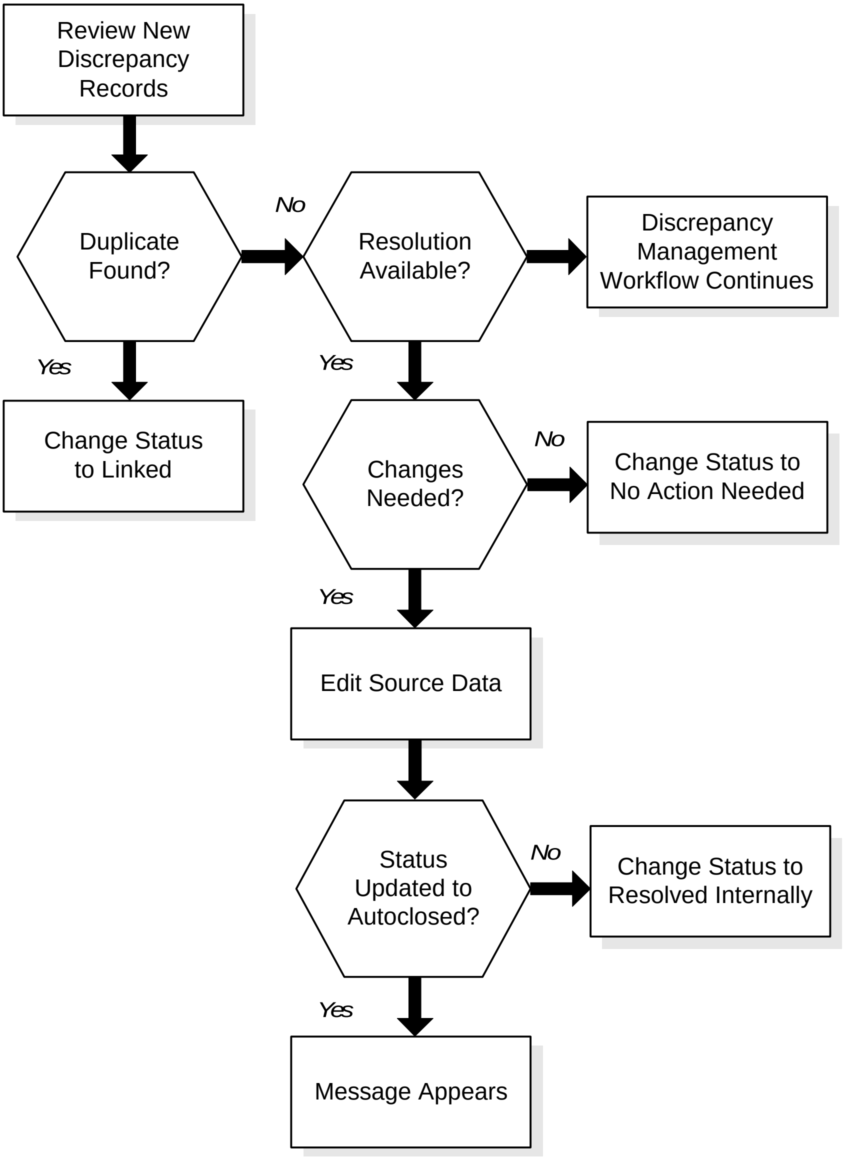
In addition, automated discrepancy status updates for obsolete and autoclosed discrepancies are described.

For step-by-step instructions on tasks described in this chapter, see the Help.

How resolution after a first review works

After discrepancy records are created, an initial review takes place, during which reviewers may discover resolutions for some discrepancy records. For example, reviewers may find transcription and typographical errors, discover that suspected data problems are satisfactory after all, or determine that two or more discrepancy records identify the same problem.

The following figure shows your choices for managing discrepancy records after a first review:


Review New Discrepancy Records
Duplicate Found?
No
Resolution Available?
Discrepancy Management Workflow Continues
Yes
Yes
Change Status to Linked
Changes Needed?
No
Change Status to No Action Needed
Yes
Edit Source Data
Status Updated to Autoclosed?
No
Change Status to Resolved Internally
Yes
Message Appears

To resolve a discrepancy record at this point in the workflow, you describe your decision in the Comments 1 field in Resolve, optionally make changes to records in Enter, and then close the discrepancy record by assigning one of these discrepancy statuses to it in Resolve:

- Resolved Internally indicates that the error reported by a discrepancy record has a straightforward solution, and that you have already made the correction to the source record.
- No Action Needed indicates that a suspected problem in the source record is actually accurate, and does not require any changes.
- Released indicates that a second internal review is called for.
- Linked indicates that two discrepancy records have an identical source record and refer to the same problem; one discrepancy record is closed while the other record remains open.

To assign a discrepancy record one of these three statuses, from the **Discrepancy** menu, select **Change Status**. When the Change Status dialog box opens, select the appropriate status, and then click **OK**. If you select **Linked**, you are prompted for the ID number of the discrepancy record that will remain open. The discrepancy record whose status you set to Linked is closed.

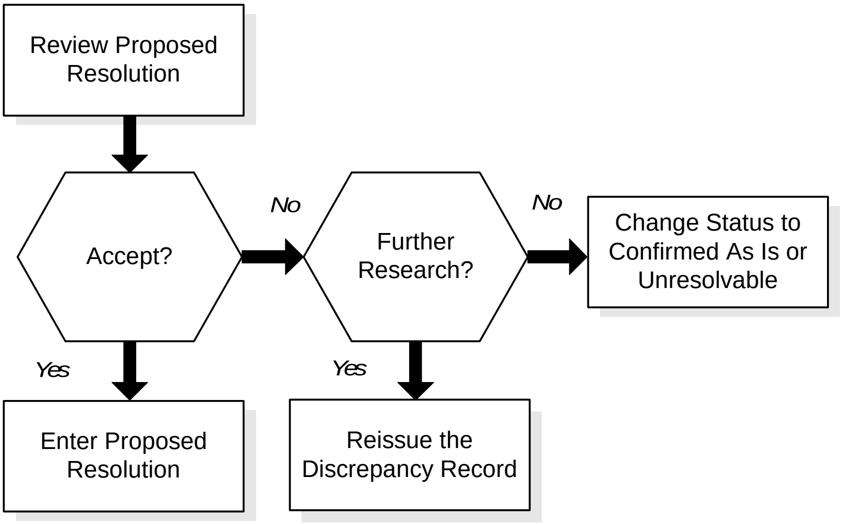
Note: If a change made elsewhere in the Clintrial software prompts the system to determine that a discrepancy record is outdated, the record closes without your having to change its status. For more information, see "Automated discrepancy resolution" on page 215.

How resolution after an expert review works

After investigators or other expert reviewers return the data discrepancy forms, you enter their proposed resolution information in each discrepancy record. Then, during a final review, you can:

- Accept the proposed resolution and update the data record.
- Seek further information from the clinical research associate (CRA) or investigator by reissuing the record.
- Conclude research on the record without accepting the proposed resolution.

The following figure shows your choices for managing discrepancy records after an expert review:


Review Proposed Resolution
Accept?
No
Further Research?
No
Change Status to Confirmed As Is or Unresolvable
Yes
Yes
Enter Proposed Resolution
Reissue the Discrepancy Record

The remaining sections in this chapter provide more detail on resolving discrepancy records after an expert review is complete.

Entering proposed resolutions

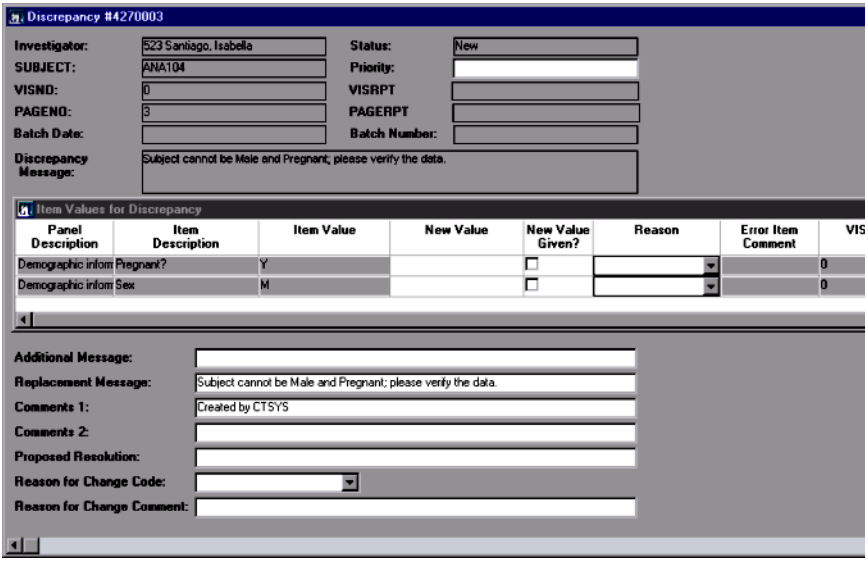
When a CRA review or on-site investigation results in a proposed resolution, descriptive text is written on the data discrepancy form, and the form is returned for review and action. To facilitate online distribution and review of proposed resolutions, you must open the detailed Discrepancy # window for the discrepancy record, record specific replacement values for incorrect items, and include descriptive information as appropriate. Then change the discrepancy record's status to Resolution Proposed.

How to store resolution information

The detailed Discrepancy # window for discrepancy records stores proposed resolution information in the following fields:

Field:	**Description:**
New Value	A specific, proposed new value for an item in a data record relating to this discrepancy record (or null).
New Value Given	A check indicates that an entry has been made in the New Value field, and that it should be used to update the data record when you select the **Discrepancy** menu's **Apply Proposed Values** command. When you check this check box, edit checks are applied to the new value to ensure its validity.
Reason	Required if New Value Given is checked. Enter comments to describe the type of resolution, such as Original Value Confirmed or Corrected Value Supplied, or select a code from the drop-down list. The designer can customize the CTS_REASON_CODES codelist as needed.
Proposed Resolution	Comments describing a proposed resolution when it is not a simple change to an associated item.
Reason for Change Code	Accepts text or a code from the CTS_REASON-_CODES codelist to describe the type of resolution.
Reason for Change Comment	Text explaining unusual resolutions.

The following figure shows the detailed Discrepancy # window of a discrepancy record:



Item Values for Discrepancy section

Discrepancy #4270003

Investigator:	523 Santiago, Isabella	Status:	New
SUBJECT:	ANA104	Priority:	
VISNO:	0	VISRPT	
PAGENO:	3	PAGERPT	
Batch Date:		Batch Number:	

Discrepancy Message: Subject cannot be Male and Pregnant; please verify the data.

Item Values for Discrepancy

Panel Description	Item Description	Item Value	New Value	New Value Given?	Reason	Error Item Comment	VIS
Demographic infom	Pregnant?	Y		□			0
Demographic infom	Sex	M		□			0

Additional Message:	
Replacement Message:	Subject cannot be Male and Pregnant; please verify the data.
Comments 1:	Created by CTSYS
Comments 2:	
Proposed Resolution:	
Reason for Change Code:	
Reason for Change Comment:	

The fields related to resolution activity at the bottom of the detailed Discrepancy # window (Proposed Resolution, Reason for Change Code, and Reason for Change Comment) are present for every discrepancy record. In contrast, a given discrepancy record can list one, several, or no items in the Item Values for Discrepancy section of the detailed Discrepancy # window (where the columns for New Value and Reason are displayed). If present, items in the Item Values for Discrepancy section display information from one or more related data records.

- Depending on how the PL/SQL programmer has written rules and derivations, items may be included in this section automatically when validation creates discrepancy records.
- When you add a new discrepancy record with the **Discrepancy** menu's **New** command, you manually select the items that display in the Item Values for Discrepancy section. Later, you can add or delete items as needed using the **Discrepancy** menu's **Error Item** command.

How to record confirmations

For a discrepancy record whose Query for Confirmation check box is checked, the data discrepancy form returned by the investigator contains either a confirmation of your changes or a different proposed resolution.

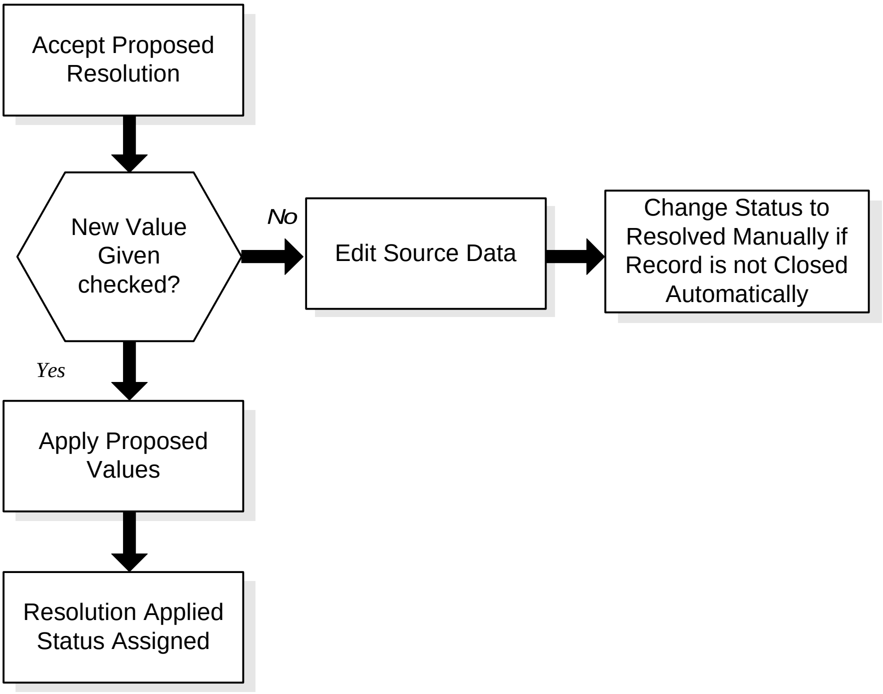
- For confirmed changes, you enter comments (if any) and close the discrepancy record by changing its discrepancy status to Resolved Manually.
- If a different resolution was proposed, you store the resolution information as explained in the previous section.

Accepting proposed resolutions

After you review and store proposed resolutions, you make the appropriate changes to the original data records by doing one of the following:

- Apply data stored in New Values fields in the discrepancy record to the referenced item(s) using Resolve.
- Access and edit the appropriate data record(s) using either Resolve or Enter.

The following figure shows the possible workflows after you accept a proposed resolution:


Accept Proposed Resolution
New Value Given checked?
No
Edit Source Data
Change Status to Resolved Manually if Record is not Closed Automatically
Yes
Apply Proposed Values
Resolution Applied Status Assigned

How applying new values works

For discrepancy records with new value(s) supplied in the detailed Discrepancy # window, you can use the **Discrepancy** menu's **Apply Proposed Values** command to update the data record.

Note: To update a data record automatically with a new value, the New Value Given check box must be checked, and the discrepancy record's status must be set to Resolution Proposed.

After you select one or more discrepancy records with new values, the **Apply Proposed Values** command performs the following activities:

- Finds each referenced data record. Both the data and update tables are checked, if necessary.
- Compares each Item Value stored in the discrepancy record to the value that is now present for that item in the referenced data record. If any of the values no longer match because of an edit made after the creation of the discrepancy, you can either cancel processing or proceed.
- Copies the discrepancy record's new value(s) to the corresponding items in the related data record(s). Only new values with check marks in the corresponding New Value Given? check boxes are copied.
- Changes the status of the discrepancy record to Resolution Applied. Discrepancy records with this status are closed and require no further action.

Apply Proposed Values processing

Apply Proposed Values processing is the most convenient way to correct discrepancies that can be resolved by replacing one value with another value; however, you cannot use this technique to correct errors associated with CONTEXT items.

In addition, remember that:

- Value changed procedures are not run when you use the **Apply Proposed Values** command to modify data records. Therefore, you should avoid using it to change the value in any field used to derive the contents of another field.
- Because **Apply Proposed Values** runs validation and cannot save a record in the data table unless the entire record passes, proposed values will not be applied to records in the data table unless those new values successfully pass validation.
- The **Apply Proposed Values** command used for one discrepancy record only updates one source record for each error item that has New Value Given? checked. For study pages that have both repeating and non-repeating sections, if you need to update values in the non-repeating section (which

affects multiple records), you should use Edit Source Data or the Enter module to edit the data, not the **Apply Proposed Values** command.

How editing source records works

Other discrepancy records with acceptable proposed resolutions may not have the appropriate items in the Item Values for Discrepancy section of the detailed Discrepancy # window. Instead of using Apply Proposed Value processing, you resolve these discrepancy records by editing the primary data record, other associated data records, or both through Resolve.

To begin editing source records, from Resolve's **Discrepancy** menu, select **Edit Source Data** or **Edit Item Source Data**. These two commands differ as follows:

- **Edit Source Data** accesses the primary data record for a selected discrepancy record.
- **Edit Item Source Data** accesses the data record for an item selected in the Item Values for Discrepancy section of the detailed Discrepancy # window.

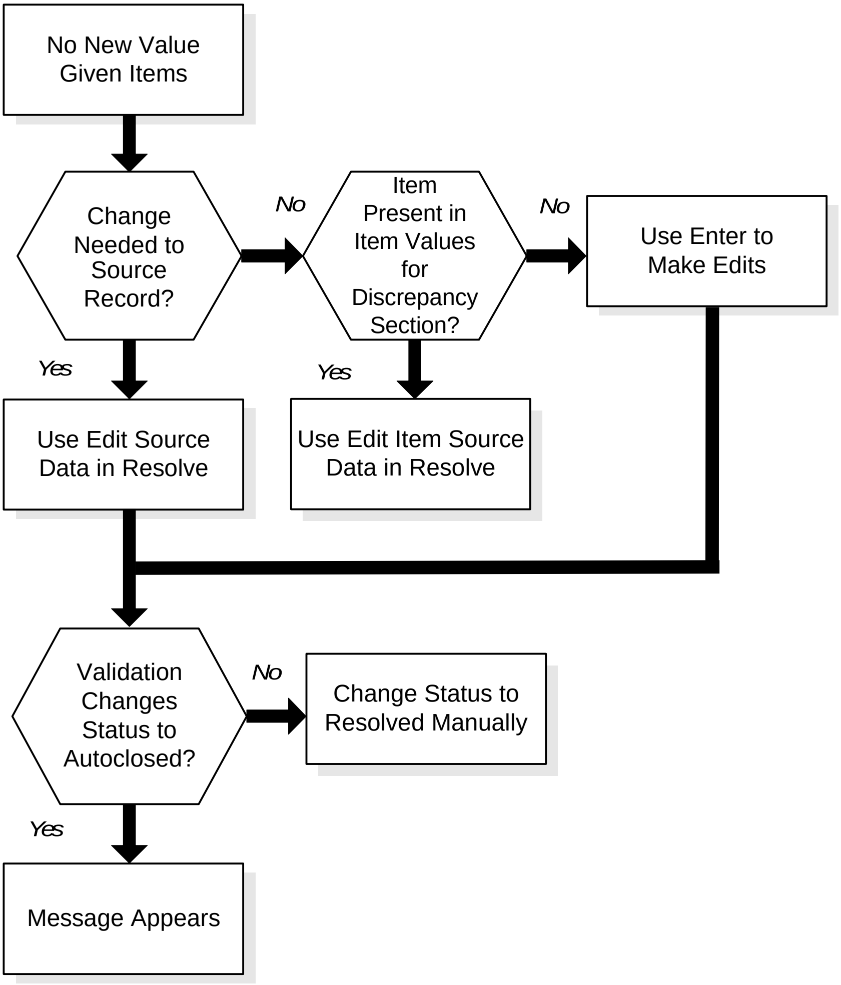
Note: You must have appropriate user access rights in both Enter and Resolve to select these commands.

Both commands open an Enter study page. From this study page, you can edit erroneous values in the data record and then save your changes. Alternatively, you can use Enter instead of Resolve to access study pages for the appropriate data records. In both Resolve and Enter, you must select a study book before you can interactively edit data. To select a study book in either module, from the **File** menu, select **Set Study Book**.

After you edit values on the study page and save your changes, validation runs and may close the discrepancy record automatically. If this occurs, a message displays when you return to Resolve. Otherwise, from the **Discrepancy** menu, select **Change Status** to change the discrepancy status of the discrepancy record to Resolved Manually. (The Change Status dialog box displays automatically for MANUAL discrepancy records.)

Resolve

The following figure shows the supported techniques for editing data records:


No New Value Given Items
Change Needed to Source Record?
No
Item Present in Item Values for Discrepancy Section?
No
Use Enter to Make Edits
Yes
Yes
Use Edit Source Data in Resolve
Use Edit Item Source Data in Resolve
Validation Changes Status to Autoclosed?
No
Change Status to Resolved Manually
Yes
Message Appears

Note: Discrepancy records whose error type is CLASSIFY identify automatic coding failures for which additional information is required. For records with this error type, you first change their discrepancy status to Ready to Send, and then generate data discrepancy forms. Once proposed resolutions are provided, you correct the coding errors, change the status of the discrepancy records to Resolved Manually, and rerun automatic coding. If the terms code successfully, the status of each omission record changes to Solved, which is considered closed in Classify.

For more information on how discrepancy records are closed automatically, see "Automated discrepancy resolution" on page 215.

Reissuing discrepancy records

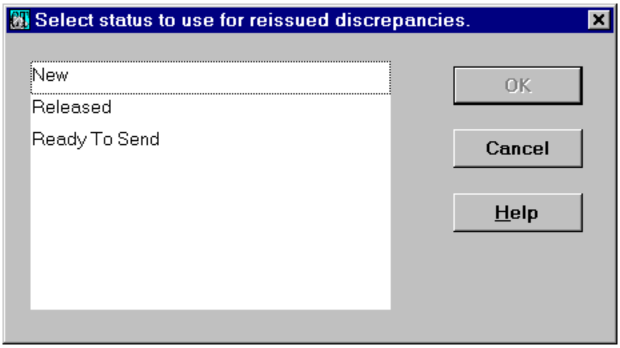
If you determine that the resolution proposed by an investigator does not resolve the discrepancy, you can indicate that you want to continue your research by reissuing the discrepancy record. This action creates a new discrepancy record as follows:

- The information from the original discrepancy record, including all associated Item Value items and their proposed new values, if any, is copied.
- The user name, date, Reissue of discrepancy, and discrepancy ID number from the original discrepancy record is added to its Comments 1 field.

After creating the new discrepancy record, the original discrepancy record is updated as follows:

- The discrepancy ID number of the new discrepancy record is inserted into its Linked Discrepancy ID field.
- Its discrepancy status is changed to Reissued, which indicates that it requires no further action.

To reissue a discrepancy record, from the **Discrepancy** menu, select **Change Status**. From the list of discrepancy statuses, select **Reissue**. The following dialog box opens:



You can assign any of these three discrepancy statuses (New, Released, Ready to Send) to the new record, which is then routed and reviewed like any other discrepancy record. When the record's discrepancy status changes to Ready to Send, you can include it with other records when you generate data discrepancy forms.

For more information on reissuing discrepancy records, see the Help.

Concluding discrepancy research

In addition to Resolution Applied and Reissued, the following discrepancy statuses are available at this point in the processing flow:

- Confirmed As Is indicates that the original data values are accurate and the proposed resolution (if any) is not applicable.
- Unresolvable indicates that the discrepancy record cannot be resolved. This discrepancy status is useful for any discrepancy that involves a subject who has dropped out of the study.

Both statuses indicate that research on a discrepancy is complete, and no further action is needed.

For more information on concluding discrepancy research, see the Help.

Automated discrepancy resolution

Finally, discrepancy records are closed automatically if they are found to be out of date. Either of the following statuses can be assigned to close discrepancy records:

- Autoclosed indicates that the discrepancy no longer exists because of changes that result in the record passing validation. This status is assigned as follows:
  - The source record associated with the discrepancy record was corrected using the **Edit Source Data** command or in Enter.
  - The discrepancy flag was deleted.
  - Validation was run again on a failed record, and the record failed the same rule, but with a different erroritem.
- Obsolete indicates that the discrepancy no longer exists because of one of the following:
  - The source data record associated with the discrepancy record was deleted
  - The panel’s clinical data tables have been deinstalled.
  - The rule was changed or deleted so that the record no longer failed validation.

  Note: Complete deinstallation of the panel (metadata tables as well as clinical data tables) results in the deletion of any discrepancy for data in that panel.

You do not assign these statuses to a discrepancy record. Instead, they are assigned automatically after one of the following system activities occurs:

- Validation (assigns the Autoclosed discrepancy status)
- Record deletion (assigns the Obsolete discrepancy status)
- Implementation of a panel revision (assigns the Obsolete discrepancy status)
- Rule deletion (assigns the Obsolete discrepancy status)

Before one of these discrepancy statuses can be assigned to a discrepancy record, its current status is checked. A discrepancy transition that allows a change from the discrepancy record's current status to the Autoclosed or Obsolete status must be on file. If there is no such discrepancy transition, the discrepancy record's status remains unchanged.

Note: Updates to records referenced in the Item Values for Discrepancy section of the detailed Discrepancy # window do not always result in automated resolution. In cases where those items are from a different panel than the discrepancy record's primary source record, change the discrepancy status to Resolved Manually when your edits are complete.

18 Reporting Operational and Statistical Data

Using Resolve’s reports

In Resolve, you can produce reports that describe the status of sets of discrepancy records and the state of the overall data-cleaning process. You can use this information to:

- Collect and monitor performance metrics on the process.
- Improve the data collection process.

All reports are generated for an entire protocol from the detailed information on file for its discrepancy records.

To display a predefined report, from Resolve’s **Report** menu, select one of the following commands:

- Discrepancy Counts by Investigator
- Discrepancy Counts by Subject/Panel
- Discrepancy Counts by Subject/Page
- Discrepancy Counts by Panel
- Discrepancy Counts by Page
- Open Discrepancy Counts by Subject and Status
- Priorities of Open Discrepancies

For step-by-step instructions on printing reports, see the Help.

Discrepancy Counts by Investigator report

This report groups a protocol’s discrepancy records by investigator, if Resolve has been set up to display investigator information by editing one of the following:

- CTV_ONE_INVESTIGATOR protocol parameter
- CT_RESOLVE_USER.INV_ID procedure

Otherwise, the subject item value displays in investigator information area of the report. For more information on setting up Resolve for display of investigator information, see the description of the Investigator field in "What is the summary Selected Discrepancies window?" on page 184.

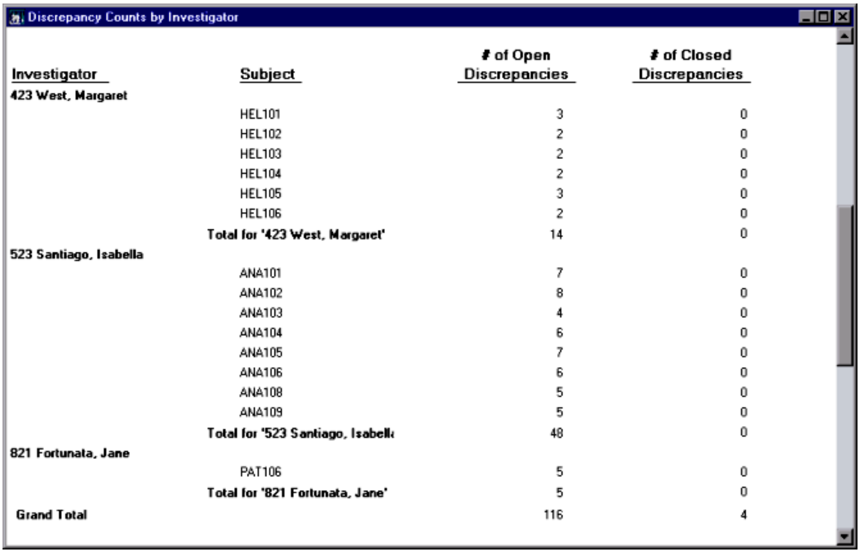
For each investigator, the report lists the specific subjects referred to by Resolve discrepancy records. Separate columns list the numbers of discrepancy records that are currently open and closed for each subject and investigator.

While many factors contribute to the presence of discrepancies in clinical data, this report may help you discover investigators or sites responsible for:

- Comparatively high numbers of discrepancy records overall
- Comparatively low numbers of closed discrepancies

With this information, you can audit procedures at specific sites to ensure, for example, that sufficient training takes place when a new case report form (CRF) is introduced, or to uncover difficulties with the
distribution or collection of data discrepancy forms. You can also discover subjects for whom data collection has been inaccurate.

The following example shows a part of a sample of this report:



Discrepancy Counts by Investigator

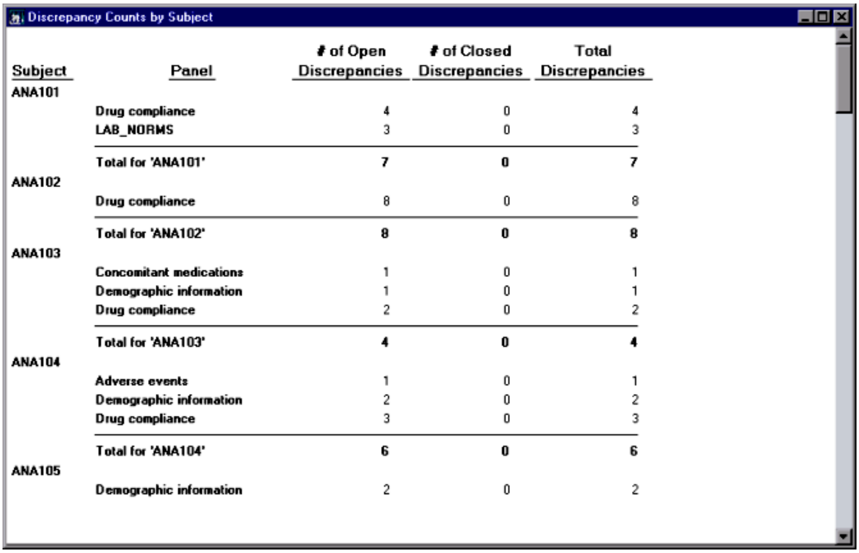
Investigator	Subject	# of Open Discrepancies	# of Closed Discrepancies
423 West, Margaret			
	HEL101	3	0
	HEL102	2	0
	HEL103	2	0
	HEL104	2	0
	HEL105	3	0
	HEL106	2	0
	Total for '423 West, Margaret'	14	0
523 Santiago, Isabella			
	ANA101	7	0
	ANA102	8	0
	ANA103	4	0
	ANA104	6	0
	ANA105	7	0
	ANA106	6	0
	ANA108	5	0
	ANA109	5	0
	Total for '523 Santiago, Isabell	48	0
821 Fortunata, Jane			
	PAT106	5	0
	Total for '821 Fortunata, Jane'	5	0
Grand Total		116	4

Discrepancy Counts by Subject/Panel report

This report groups a protocol’s discrepancy records by subject. For each subject, discrepancies are listed by panel. As in the Discrepancy Counts by Investigator report, separate columns list the number of discrepancy records currently open and closed for each subject and panel.

Using this report, you can isolate subjects for whom data is inaccurate. The data collection and entry process used for these subjects can be examined to help determine the origin of the discrepancies.

The following example shows a part of a sample of this report:



Subject	Panel	# of Open Discrepancies	# of Closed Discrepancies	Total Discrepancies
ANA101				
	Drug compliance	4	0	4
	LAB_NORMS	3	0	3
	Total for 'ANA101'	7	0	7
ANA102				
	Drug compliance	8	0	8
	Total for 'ANA102'	8	0	8
ANA103				
	Concomitant medications	1	0	1
	Demographic information	1	0	1
	Drug compliance	2	0	2
	Total for 'ANA103'	4	0	4
ANA104				
	Adverse events	1	0	1
	Demographic information	2	0	2
	Drug compliance	3	0	3
	Total for 'ANA104'	6	0	6
ANA105				
	Demographic information	2	0	2

Discrepancy Counts by Subject/Page report

This report groups a protocol’s discrepancy records by subject. For each subject, discrepancies are listed by both block and page context items. (The first two column names reflect the current protocol’s block key item and page key item values.) As in the Discrepancy Counts by Investigator report, separate columns list the number of discrepancy records that are currently open and closed for each subject, block, and page grouping.

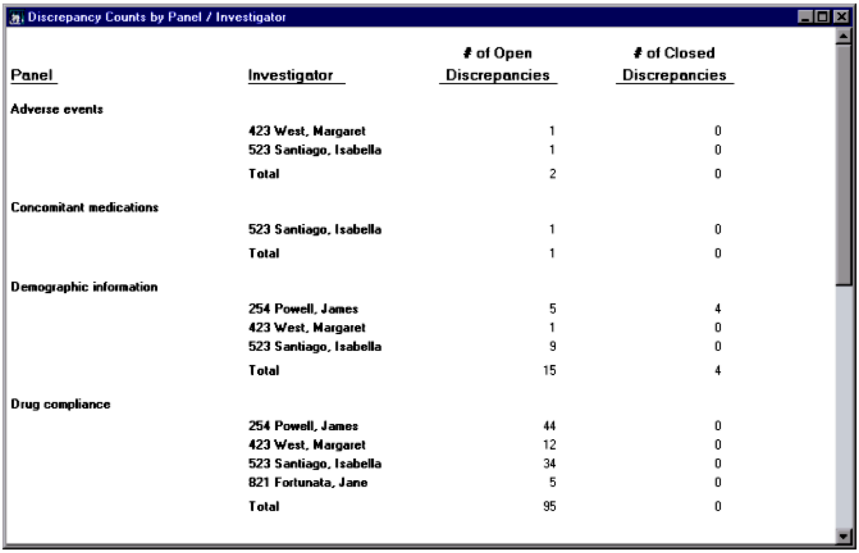
With this report, you may be able to identify subjects for whom data is particularly prone to errors. For a sample of this report, see the Discrepancy Counts by Subject/Panel report, which is identical except that it lists panels instead of blocks and pages.

Discrepancy Counts by Panel report

This report groups a protocol’s discrepancy records by panel and investigator, if Resolve has been set up to display investigator information. For each panel that has one or more discrepancy records in Resolve, separate columns list the number of discrepancy records currently open and closed for each investigator.

Using this report, you can determine whether certain panels in the protocol result in more errors than other panels. If there are panels with significantly more discrepancy records, additional training for data-entry operators or redesigning the corresponding study book page might be indicated.

The following example shows a part of a sample of this report:



Discrepancy Counts by Panel / Investigator

Panel	Investigator	# of Open Discrepancies	# of Closed Discrepancies
Adverse events			
	423 West, Margaret	1	0
	523 Santiago, Isabella	1	0
	Total	2	0
Concomitant medications			
	523 Santiago, Isabella	1	0
	Total	1	0
Demographic information			
	254 Powell, James	5	4
	423 West, Margaret	1	0
	523 Santiago, Isabella	9	0
	Total	15	4
Drug compliance			
	254 Powell, James	44	0
	423 West, Margaret	12	0
	523 Santiago, Isabella	34	0
	821 Fortunata, Jane	5	0
	Total	95	0

Discrepancy Counts by Page report

This report groups a protocol’s discrepancy records by both block and page context items, and then by investigator, if Resolve has been set up to display investigator information. (The first two column names reflect the current protocol’s block key item and page key item values.) For each block and page with one or more discrepancy records, separate columns list the number of records that are currently open and closed for each investigator.

Using this report, you can determine whether certain pages have significantly more errors than other pages. For a sample of this report, see the Discrepancy Counts by Panel report, which is identical except that it lists panels instead of blocks and pages.

Open Discrepancies by Subject and Status report

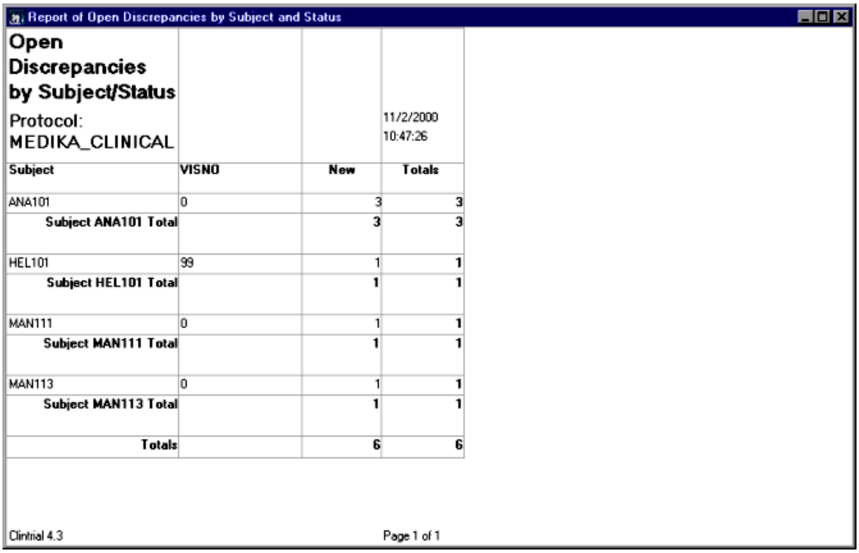
This report includes only discrepancy records that are still in progress. For each subject, the total number of open discrepancy records displays by visit. In addition, columns report the total number of discrepancy records with the following statuses:

- New
- Released
- Ready to Send
- Sent
- Resolution Proposed

This report helps you to analyze the efficiency of your Resolve workflow. If an unusually large number of open discrepancy records have the same status, a problem may exist at that point in the workflow.

Note: If the designer has customized your discrepancy statuses, only those with a value of 0 (zero) for the End Final item in the DISCREP_STATE panel are included on this report.

The following example shows a sample of this report:



Report of Open Discrepancies by Subject and Status

Open Discrepancies by Subject/Status

Protocol: MEDIKA_CLINICAL

11/2/2000
10:47:26

Subject	VISNO	New	Totals
ANA101	0	3	3
Subject ANA101 Total		**3**	**3**
HEL101	99	1	1
Subject HEL101 Total		**1**	**1**
MAN111	0	1	1
Subject MAN111 Total		**1**	**1**
MAN113	0	1	1
Subject MAN113 Total		**1**	**1**
Totals		**6**	**6**

Clintrial 4.3

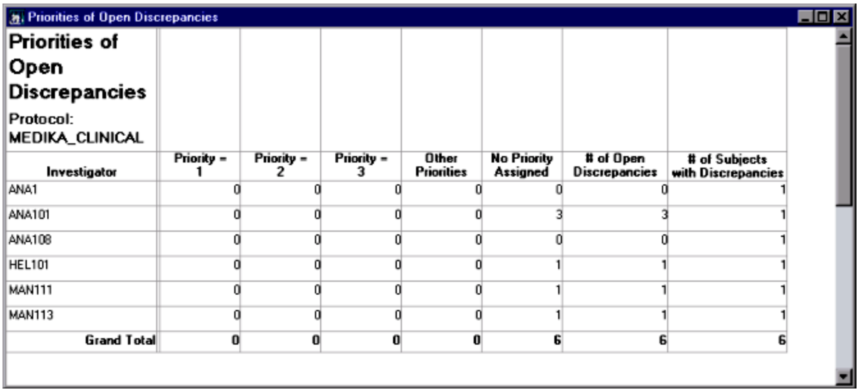
Page 1 of 1

Priorities of Open Discrepancies report

Like the Open Discrepancies by Subject and Status report, this report includes only open discrepancy records. On this report, the total number of open discrepancy records is listed for each investigator, if Resolve has been set up to display investigator information, with columns for the number of discrepancy records in different priority levels. An additional column lists the number of subjects assigned to each investigator.

Using this report, you can review the outstanding workload for each investigator who is working on a protocol. If an investigator has a large number of high priority discrepancy records, reminders to resolve those records may be needed. In addition, by listing the number of subjects assigned to each investigator as well as the number of open discrepancy records, you can determine when the workload is not equally distributed among the investigators.

The following example shows a sample of this report:



Priorities of Open Discrepancies

Protocol: MEDIKA_CLINICAL

Investigator	Priority = 1	Priority = 2	Priority = 3	Other Priorities	No Priority Assigned	# of Open Discrepancies	# of Subjects with Discrepancies
ANA1	0	0	0	0	0	0	1
ANA101	0	0	0	0	3	3	1
ANA108	0	0	0	0	0	0	1
HEL101	0	0	0	0	1	1	1
MAN111	0	0	0	0	1	1	1
MAN113	0	0	0	0	1	1	1
Grand Total	**0**	**0**	**0**	**0**	**6**	**6**	**6**

19 Resolve Setup

Overview

After Resolve is installed, certain preparatory tasks must be performed before data managers can begin to use Resolve to manage discrepancies. This chapter describes the two required setup activities:

- Setting up a protocol for use with Resolve
- Assigning Resolve access rights to users

This chapter also describes the following that you can perform to customize Resolve:

- Define system parameters.
- Set user preferences.
- Set protocol preferences.
- Turn auditing off or on.

For more information

The next two chapters describe additional tasks that you may perform before using Resolve:

- Chapter 20 explains how to modify key aspects of Resolve, such as its discrepancy statuses and data discrepancy forms.
- Chapter 21 explains how to ensure that discrepancy records are generated in response to validation failures, and that they contain all of the necessary information.

Setting up a protocol for Resolve

You can use Resolve to manage discrepancies for any number of protocols. Before you set up a protocol to use Resolve, ensure that each of its panels has a Panel Description value, which helps users identify the source of each discrepancy record. (If there is no panel description, the corresponding item displays as blank in the Resolve summary Selected Discrepancies window and detailed Discrepancy # window.)

How to set up a protocol

Before you initialize a protocol for Resolve, you must:

- Be assigned Admin’s System access right, which is a non-protocol access right.
- Have imported the CTRESOLVEREF protocol to the database instance.

To set up a protocol in Resolve, from the **File** menu, select **Set Protocol**. The Set Protocol dialog box opens. Select a protocol. Then, from the **Setup** menu, select **Setup For Resolve** to initiate the setup process. A message displays after the Resolve Setup process completes.

For more information on importing the CTRESOLVEREF protocol to the database instance, see the *Server Installation* section of *Getting Started.*

Types of changes resulting from protocol setup

The protocol setup process:

- Copies required panels from the CTRESOLVEREF protocol to the selected protocol.
- Copies records from the *protocol-name*.ERRORLOG table to the VCT_ERRORSTATUS panel.
- Creates Oracle sequences.
- Creates Resolve-specific access rights for the protocol

Note: Protocol setup bypasses any steps that were already performed. For example, if you are importing a protocol that was set up at the exporting site, setup does not recreate existing panels; it proceeds directly to the remaining tasks.

What are the required panels?

When you set up a protocol for Resolve, the following two panels in the CTRESOLVEREF protocol are copied into the selected protocol:

- VCT_ERRORSTATUS stores one record for each discrepancy found in the protocol.
- VCT_ERRORITEM stores one record for each item that you associate with a VCT_ERRORSTATUS record. Each record contains columns where proposed new values can be entered for the item.

Typically, one record from the VCT_ERRORSTATUS panel and one or more records from the VCT_ERRORITEM panel compose a discrepancy record. While a discrepancy record can consist solely of a VCT_ERRORSTATUS record, the VCT_ERRORITEM records help users review and resolve discrepancies. VCT_ERRORITEM records can be included in discrepancy records that are created during validation using rules and derivations. (For details about how to define such rules and derivations, see Chapter 21.) Rules and derivations can also be appended to discrepancy records that were created manually.

Resolve stores all data for these panels in the update table so that audit capabilities can be used. For more information, see "Auditing discrepancy data" on page 245.

How copying records from the ERRORLOG works

When you set up a protocol for use with Resolve, all of the records in the *protocol-name*.ERRORLOG table whose error type is VALIDATE or MERGE are copied to the VCT_ERRORSTATUS panel, provided that no data is already stored in that panel. These discrepancy records are assigned a status of New.

Note: ERRORLOG table records exist only if validation or merge was run before the protocol was set up for Resolve.

During this initial setup process, only the VCT_ERRORSTATUS record is included in new discrepancy records. VCT_ERRORITEM records are not included.

What are the Oracle sequences?

Two Oracle sequence objects are created during setup for use in the protocol. These sequences generate unique identifying numbers for:

- Discrepancy ID numbers
- Data discrepancy form batch ID numbers

These ID numbers are unique within a given protocol.

Assigning access rights in Admin

A complete set of module-specific protocol access rights is installed with Resolve. All Resolve users with an access level of Read, Full, or No Delete in any of these access rights can:

- Review summary Selected Discrepancies windows of discrepancy records using **View** menu commands.

- Open a detailed Discrepancy # window for selected discrepancy records.
- View the More Detail window for the selected records.
- Define a user preference for a startup view using the **File** menu's **Preferences** command.

Access rights and access levels are assigned to users and usergroups in Admin.

Types of Resolve access rights

After you set up a protocol for use with Resolve, you (or your Clintrial software administrator) assign each user or usergroup one or more of the Resolve access rights and select the appropriate access level for each user or usergroup. By selecting an access level of either Full or No Delete, you enable users or usergroups to perform certain additional tasks in Resolve.

The protocol access rights specific to Resolve are:

Access right:	Access level:	Users can:
Create	Full or No Delete	Perform both of these actions: • Create new discrepancy records using the **Discrepancy** menu's **New** command. • Add associated items to or delete them from manually created records using the **Discrepancy** menu's **Error Item** command. The **Delete** command is available only if your access level is Full. Users with Create access can also modify the Additional Message, Replacement Message, and Proposed Resolution values in any records that they created manually, provided their discrepancy status is New.
Produce	Full	Print data discrepancy forms using the **Report** menu's **Produce Form** command.

Access right:	**Access level:**	**Users can:**
Propose	Full or No Delete	Perform both of these actions: • Modify all of a discrepancy record's values except Comments 1 and Query for Confirmation, which can be modified only by Manage users. • Change the status of a discrepancy record using the **Discrepancy** menu's **Change Status** command. ***Note**:* Users with Propose access cannot add associated items to or delete them from manually created records. The **Discrepancy** menu's **Error Item** command is available only to users with Create access.
Manage	No Delete	Perform these four actions: • Run predefined reports from the **Report** menu. (You only need Read access to run predefined reports.) • Apply new data values to data records using the **Discrepancy** menu's **Apply Proposed Values** command. • Modify all of a discrepancy record's values except Comments 2, which can be modified only by users with Propose access. • Change the status of a discrepancy record using the **Discrepancy** menu's **Change Status** command.
	Full	Perform all actions available to users with Manage No Delete access rights, and can delete discrepancy records using the **Discrepancy** menu's **Delete** command. ***Note**:* These users cannot add associated items to or delete them from manually created records. The **Discrepancy** menu's **Error Item** command is available only to users with Create access.

To prevent access to Resolve's functions, you can assign users an access level of null or "-" for all of the Resolve access rights. Null is the system default.

After Resolve is installed, you can assign Resolve access rights to the *pseudo* protocol DEFAULT. The access rights you specify are then copied from DEFAULT to the protocol when you run the setup routine.

How access rights work with a discrepancy status

Users with an access right of Propose or Manage can modify values for discrepancy records in Resolve. However, the ability to modify items depends not only on the user's access right, but also on the discrepancy record's current discrepancy status.

A panel in the CTRESOLVEREF protocol, DISCREP_STATE, stores a record for each Resolve discrepancy status. For each discrepancy status, you can define values for a number of items, including:

- PROPOSE_CAN_MODIFY — Indicates that a user with the Propose access right can modify values for a discrepancy record that currently has this status.
- MANAGE_CAN_MODIFY — Indicates that a user with the Manage access right can modify values for a discrepancy record that currently has this status.

Each of these items stores a value of 0 = No (cannot modify) or 1 = Yes (can modify).

Note: The DISCREP_STATE panel also includes an END_FINAL value for each discrepancy status. If a discrepancy record is in a final (closed) status as identified by this item, it cannot be modified regardless of the values in the two CAN_MODIFY items.

Whenever a user attempts to edit the values for a discrepancy record in Resolve, the following tests are applied to determine whether the particular user can modify values for the particular discrepancy record:

1. Can the user modify the discrepancy record?

   Only if the user's access right is Propose or Manage.

   There is an exception for users with Create access. These users can modify values in the Additional Message, Replacement Message, and Proposed Resolution fields in their own manually created discrepancy records as long as the records remain in New status.

2. Can the user modify this particular discrepancy record?

   The current discrepancy status of the discrepancy record is determined. If the discrepancy record is in a closed discrepancy status, it cannot be modified. If its status is not a terminal status, the value in the CAN_MODIFY field corresponding to the user's access right is checked in the DISCREP_STATE panel.

3. Can the user modify this discrepancy record item?

   With the following exceptions, users with either the Propose or Manage access right can modify all items:

Item:	Can modify with access right:
Comments 1	Manage only
Comments 2	Propose only
Query for Confirmation	Manage only

Note: Each record installed in DISCREP_STATE is supplied with a default PROPOSE_CAN_MODIFY and MANAGE_CAN_MODIFY value.

How access rights work with transitions

In addition to being able to modify individual item values for discrepancy records, users with the Propose or Manage access right can change the discrepancy statuses of discrepancy records. A user’s ability to change discrepancy statuses is subject to values in both the DISCREP-_STATE and DISCREP_TRANSITION panels in the CTRESOLVEREF protocol.

In the DISCREP_STATE panel, there is one row for each discrepancy status. The following items apply to changing discrepancy statuses:

- PROPOSE_CAN_CHANGETO indicates that a user with Propose access rights can change a discrepancy record to this status.
- MANAGE_CAN_CHANGETO indicates that a user with Manage access rights can change a discrepancy record to this status.

These items store a value of 0 = No (cannot modify) or 1 = Yes (can not modify).

The DISCREP_TRANSITION panel in the CTRESOLVEREF protocol stores records that define the valid transitions between discrepancy statuses. There is one record for each possible transition.

The following tests determine whether a particular user can change the discrepancy status of a particular discrepancy record:

1. Can the user change the discrepancy status?

   Only if the user’s access right is Propose or Manage.

2. To what other discrepancy status can the user change this discrepancy record?

   When a user selects the **Discrepancy** menu's **Change Status** command, a list of possible discrepancy statuses displays. To determine the items that display on this list, both the DISCREP_TRANSITION and the DISCREP_STATE panels are checked:

   a. If there are any records in DISCREP_TRANSITION that allow a transition from the discrepancy record's current discrepancy status to any other discrepancy status, each corresponding DISCREP-_STATE record is checked. Proceed to b.

   b. In the DISCREP_STATE panel, the value in the CAN_CHANGETO item corresponding to the user's access right is checked for each of the discrepancy statuses previously determined. Only discrepancy statuses with a 1 (Yes) in this field display for the user to assign to the discrepancy record.

Each record installed in DISCREP_STATE is supplied with default values for PROPOSE_CAN_CHANGETO and MANAGE_CAN_CHANGETO.

Note: Certain discrepancy statuses, including Sent and Resolution Applied, are not assigned by using the **Change Status** command. However, the same tests are applied when a user attempts to print a batch of data discrepancy forms (which changes discrepancy status to Sent) or uses apply proposed value processing (which changes discrepancy status to Resolution Applied).

For more information on changing the defaults for items in the DISCREP_STATE panel, see Chapter 20.

For more information on the DISCREP_STATE panel, see the *Reference Guide*.

How access rights work for editing source data

After users receive proposed resolutions for discrepancy records, they can resolve discrepancies by selecting the **Discrepancy** menu's **Edit Source Data** and **Edit Item Source Data** commands. These commands open an Enter study page, where a record relevant to a discrepancy can be reviewed or edited. For these commands to be available, the user must have the following access rights in both Resolve and Enter:

- In Resolve, users must have the Read access level (or above) for any Resolve access right.
- In Enter, users must have the Read access level (or above) for the appropriate Enter access rights.

When the **Discrepancy** menu commands are selected, users who are assigned only the Read access level for access rights in Enter can review records. Users with other access levels for Enter access rights can modify values in the study page and save their changes as defined by their specific combination of access levels and access rights.

Note: Edit source data processing is not the same as apply proposed value processing. Only users who are assigned the Full or No Delete access level for Resolve's Manage access right can select the **Discrepancy** menu's **Apply Proposed Values** command.

For more information on resolving discrepancy records, see Chapter 17.

For more information on Enter access rights and access levels, see Chapter 14.

Defining system parameters in Admin

You set Resolve system parameters in Admin. The Resolve tab of the System Parameters dialog box lists system parameters, protocol parameters, and user preferences, and indicates which of these three types each item represents.

Types of system parameters

The following table describes the system parameters:

System parameter:	**Description:**
CTV_APPL_RES_CODE	In Resolve, the **Discrepancy** menu's **Apply Proposed Values** command updates data records and closes discrepancy records. It also supplies this system parameter value as the default Reason for Change code in the discrepancy record. To define a value for this parameter, you select from the CTS_REASON_CODES codelist, which can be modified in Design. ***Note****:* The choices for this parameter display as the codelist codes from CTS_REASON_CODES; however, they display as the codelist values in the detailed Discrepancy # window's Reason for Change Code field.

System parameter:	Description:
CTV_AUTOCLOSED_RES	Validation detects changes to source data records that correct discrepancies and remove discrepancy flags, and assigns the Autoclosed status to close any corresponding discrepancy records. It also supplies this system parameter value as the default Reason for Change code in the discrepancy record. To define a value for this parameter, you select from the CTS_REASON_CODES codelist, which can be modified in Design. ***Note**:* The choices for this parameter display as the codelist codes from CTS_REASON_CODES; however, they display as the codelist values in the detailed Discrepancy # window's Reason for Change Code field
CTV_OBSOLETE_RES	Changes to metadata, including panel deinstallation or deletion of data records or rules, may make discrepancy records obsolete. The Obsolete discrepancy status is assigned as necessary, and this system parameter value is supplied as the default Reason for Change code in the discrepancy record. To define a value for this parameter, you select from the CTS_REASON_CODES codelist, which can be modified in Design. ***Note**:* The choices for this parameter display as the codelist codes from CTS_REASON_CODES; however, they display as the codelist values in the detailed Discrepancy # window's Reason for Change Code field
CTV_USE_ERRORLOG Default: Yes	Determines whether a record is added to the Error Log when a discrepancy record is added in Resolve for each record that fails a rule during validation. If Yes, both an Error Log record and discrepancy record are added; if No, only a discrepancy record is created.

Options for overriding parameters

Protocol parameters override user preferences, which override system parameters. The designer can override system parameters of type Protocol by specifying protocol parameters, and individual users can override system parameters of type User by specifying user preferences. The following sections explain how to set Resolve's user preferences and protocol parameters.

Setting user preferences in Resolve

You can modify user preferences to customize both your working environment and the contents of data discrepancy forms. The settings that you assign to user preferences affect only your user or usergroup account. The following table describes the Resolve user preferences:

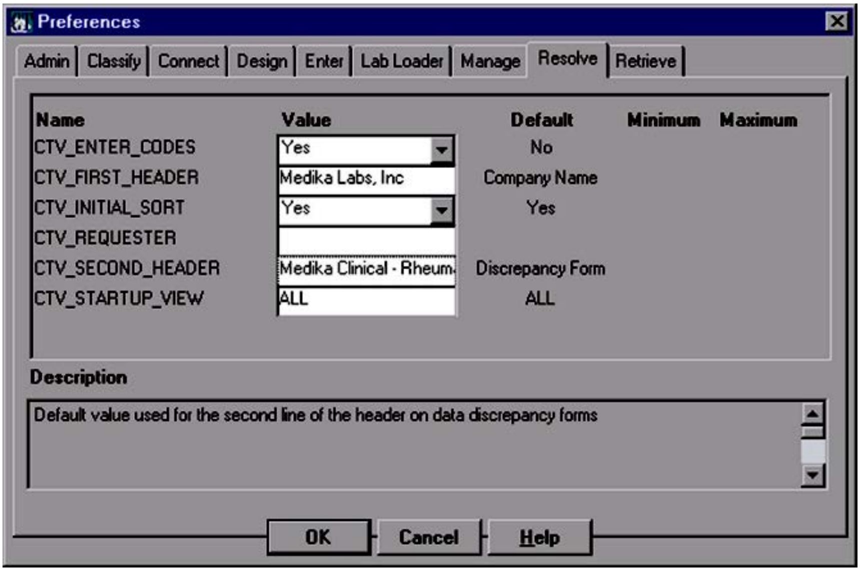
User preference:	**Description:**
CTV_ENTER_CODES Default: No	Determines whether you enter codelist codes or codelist values display in the discrepancy detailed Discrepancy # window's Item Values for Discrepancy section, in the New Value field: • If Yes, enter codelist codes. • If No, enter codelist values. ***Note***: The New Value field always displays the codelist value, whether you entered it as a code or a value.
CTV_FIRST_HEADER Default: Company Name	First line to print on data discrepancy forms, for example, the company name.
CTV_INITIAL_SORT Default: Yes	Determines whether discrepancy records in the summary Selected Discrepancies window display in order based on the subject item value or display as retrieved: • If Yes, the records sort by subject item. • If No, the records display as retrieved. ***Note:*** For large numbers of discrepancy records, a No value results in a quicker display of the records. ***Note:*** For maximum efficiency, it is recommended that when CTV_RETRIEVE_ALL (see below) is set to No, that CTV_INITIAL_SORT also be set to No.

User preference:	**Description:**
CTV_REQUESTER	Name of the person (such as a data manager or CRA) who requests discrepancy resolutions from investigational sites. This name is included on all of the data discrepancy forms you print. ***Note***: This name does not need to be a Clintrial software user name.
CTV_RETRIEVE_ALL Default: No	Determines whether all records that could be retrieved in the summary window are retrieved from the server before any display. If this parameter is set to No, then only enough records are retrieved to put up the display and more records are retrieved when you scroll down. When CTV_RETRIEVE_ALL is No, the title of the summary window will say "retrieved as needed". When CTV_RETRIEVE_ALL is Yes, the title of the summary window will include the number of records retrieved. ***Note:*** The summary window will display more quickly if CTV_RETRIEVE_ALL is No (default). ***Note:*** For maximum efficiency, it is recommended that when CTV_RETRIEVE_ALL is set to No, that CTV_INITIAL_SORT also be set to No.
CTV_SECOND_HEADER Default: Discrepancy Form	Second line to print on data discrepancy forms, for example, the study name or a department name.
CTV_STARTUP_VIEW Default: All	Determines which data subset displays in the summary Selected Discrepancies window each time you start Resolve. Options include: 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

User preference:	Description:
SELECT_BY_PAGE Default: No	Determines the contents and layout of the New Discrepancy and Select Criteria for Discrepancy Browsing dialog boxes: • If Yes, page values replace panels or assume a more prominent position. • If No, panels display. ***Note***: Set this user preference in the Manage tab.

User preferences take their default values from the corresponding system parameters. To set user preferences:

1. From the **File** menu, select **Preferences**. The Preferences dialog box opens with the Resolve tab preselected:



Name	Value	Default	Minimum	Maximum
CTV_ENTER_CODES	Yes	No		
CTV_FIRST_HEADER	Medika Labs, Inc	Company Name		
CTV_INITIAL_SORT	Yes	Yes		
CTV_REQUESTER				
CTV_SECOND_HEADER	Medika Clinical - Rheum	Discrepancy Form		
CTV_STARTUP_VIEW	ALL	ALL		

Description

Default value used for the second line of the header on data discrepancy forms

2. Modify the CTV_ user preferences as appropriate. To modify the SELECT_BY_PAGE preference, you must first select the Manage tab.
3. When you complete your changes, click **OK**.

Note: User preferences override system parameters, and protocol parameters override user preferences.

Setting protocol parameters in Design

Each time you set up a protocol for Resolve, the protocol is initialized to include the default values for the protocol parameters. If those values are not appropriate for a specific protocol, you can modify them in Design. To set protocol parameters, from the **Objects** menu, select **Protocol**, and then from the **Protocol** menu, select **Parameters**. The Modify Protocol Parameters dialog box opens. Scroll to the parameter names that start with CTV_, and make your changes.

If the default values are not appropriate for your company's users, you can modify system parameters, as explained in "Defining system parameters in Admin" on page 234. If you modify values at the system level, your revised values are assigned to all protocols for which no protocol parameters are set within Design.

Note: An alternative method of modifying a protocol parameter's value is to copy the value to the Clipboard, and then to a text editor, such as Notepad. You then modify the value, and copy the revised text into the appropriate field.

Types of protocol parameters

The following table lists the protocol parameters:

Protocol parameter:	**Description:**
CTV_FIRST_HEADER_PR Default: Null	Text for the first line to print on all of the protocol's data discrepancy forms. If set, it overrides the user preference CTV_FIRST_HEADER.

Protocol parameter:	Description:
CTV_ONE_INVESTIGATOR	SQL fragment to use within a selection statement to get the investigator information from a field in the discrepancies table (VCT_ERRORSTATUS_UPDATE). The default is CT_RESOLVE_USER.INV_ID(VCT_ERRORSTATUS_UPDATE.CTV_CONTEXT1, '<PROTOCOL>') If there is a simple algorithm for computing the investigator information during the 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 "Modifying investigator retrieval instructions" on page 256.
CTV_REQUESTER_PR Default: Null	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 user preference CTV_REQUESTER.
CTV_SECOND_HEADER_PR Default: Null	Text for the second line to print on all of the protocol’s data discrepancy forms. If set, it overrides the user preference CTV_SECOND_HEADER.

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 you edit the CT_RESOLVE_USER.INV_ID function, this parameter retrieves investigator information based on that customization. ***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

Protocol parameter:	**Description:**
CTV_SELECT_SUBJECTS	SQL statement for retrieving a list of subjects as text. This statement is used by the **View** menu’s **Select By** dialog when the User chooses from a list of investigators, and causes a list of subjects to appear. It is also used by 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 modified, this statement must contain the ‘SUBSTITUTE*INVESTIGATOR’ clause. ***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'
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'.

Protocol parameter:	Description:
CTV_STARTUP_VIEW_PR Default: Null	Determines which data subset displays in the summary Selected Discrepancies window each time you start Resolve. Options include: 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 If set, it overrides the user preference CTV_STARTUP_VIEW.

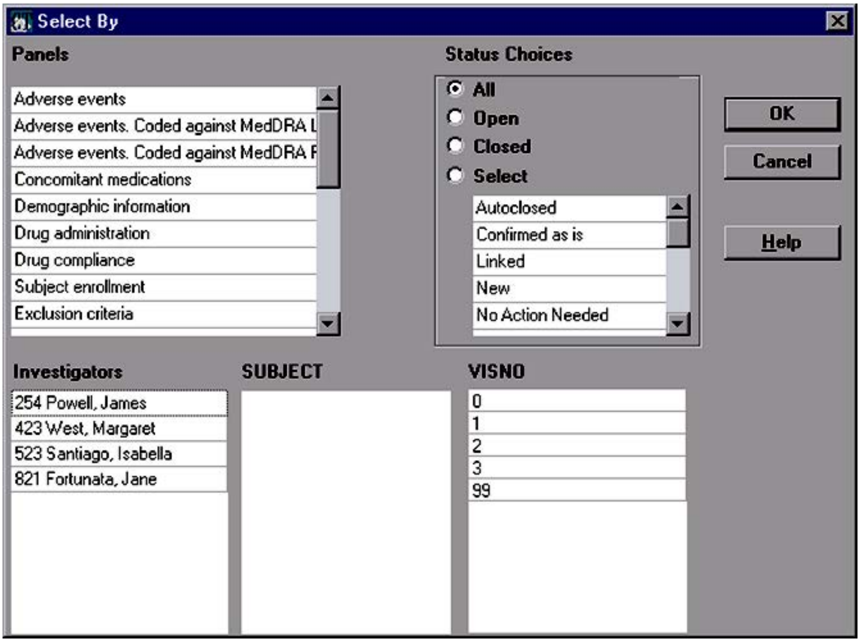
The First Header, Second Header, Requester, and Startup View can also be defined as user preferences by individual users of Resolve. However, if you set these protocol parameters, the settings override all individual user preferences.

How view filters work

The remaining protocol parameters define the view filters used by Resolve. *View filters* help users retrieve and organize discrepancy records. They also help users locate source records, so that a new discrepancy record can be added.

Note: In most cases, you do not need to change any of these filters if you customize information about investigator retrieval, as explained in Chapter 20.

In Resolve, the **View** menu's **Select By** command and the **Discrepancy** menu's **New** command use view filters to determine what items to display on selection lists. For example, when a user selects the **Select By** command, the Select By dialog box opens:



Note: The layout of this dialog box varies depending on the setting of the SELECT_BY_PAGE user preference. This example reflects the default setting (No).

The items in the Investigators list are determined by the value of CTV_SELECT_INVESTIGA. When you choose an investigator, the *subject_item* list (PATIENT in this example) changes accordingly. Your choice from that list in turn determines the contents of the *block_item* list (BLOCKVALUE in this example).

The SQL statements in the protocol parameters that define view filters specify the database location of the appropriate data elements. Two types of variables are included in the defaults for these parameters:

- Variables shown in italics, such as *block_item*, are replaced with protocol-specific values when the protocol is set up.
- Variables preceded by SUBSTITUTE* are provided at run time using information specified by the user.

The format for each of these items is text. If you modify the SQL statement to reference an item that is not a VARCHAR2 type, you should include TO_CHAR in the SQL statement; that is, begin the statement with “SELECT UNIQUE TO_CHAR”.

Modifications to these SQL statements can be made either at the protocol or system level. If you modify values at the system level, your revised values are assigned to all protocols for which no protocol parameters are set within Design.

Auditing discrepancy data

Auditing is the tracking of changes made to discrepancy records. When Resolve is installed, protocol-specific auditing is enabled by default. The start point is set to ENTRY for the CTRESOLVEREF protocol and for the panels that are copied into other protocols during setup (VCT_ERRORSTATUS and VCT_ERRORITEM).

If you do not want one of these panels to be audited, you can set its audit start point to MERGE in the CTRESOLVEREF protocol. Data in Resolve panels is not merged; therefore, this action disables auditing for any protocols that you later set up for use with Resolve.

You make protocol- and panel-level auditing changes in Design. If you use a database-wide audit start point rather than protocol-specific auditing, the audit start point also applies to Resolve. To set or change the database-wide default, use Admin to set the AUDIT_START_DEFAULT system parameter.

You can use Manage to select records from the audit table for reporting.

For more information on auditing and Resolve, see the Help.

20 Customizing Resolve

Overview

This chapter explains how to customize the following features:

- The layout of data discrepancy forms
- The discrepancy statuses that can be assigned to discrepancy records
- The transitions allowed between discrepancy statuses
- Investigator retrieval instructions

When you customize these aspects of Resolve, your changes affect all protocols set up for use with Resolve.

For information on parameters that you can set for specific protocols, see Chapter 19.

Redesigning data discrepancy forms

The source file for the standard data discrepancy form that is described in Chapter 16 is provided when you install Resolve. To modify the layout of this form, you must use PowerBuilder™ 12.5, which is sold and licensed by Sybase, Inc.

Note: To update existing data discrepancy forms from Clintrial 4.5, see the section "Upgrading Data Discrepancy Forms (DDFs) to Clintrial 4.7" in **Chapter 4: Upgrading to Clintrial 4.7 from release 4.5** in the *Clintrial 4.7 Getting Started* guide.

Reformatting the DataWindow

Each Clintrial software client has the CTV_REPT.PBL PowerBuilder source file in its directory. The DataWindow in CTV_REPT.PBL is discrepancy_form_print. You may reformat the DataWindow to meet the needs of your users, as follows:

- Add labels and columns; move them within a given band (portion of the form); and add fields to or delete fields from PowerBuilder's design window. You can also hide various items, and change the labels associated with particular fields.
- You may not delete fields from the selection window (the window that displays when you select the **Design** menu's **Data Source** command). Nor may you change the names of fields in that window, or move items from one

band to another. If you want to suppress certain fields, delete them only from the display portion of the window.

When your changes are complete, save the discrepancy_form_print DataWindow, and then compile the CTV_REPT.PBL file. This saves the changes in both the CTV_REPT.PBL source file and a corresponding.PBD object file. After testing, copy the new CTV_REPT.PBD file to each Clintrial software client computer, overwriting the standard version of that file.

The following sections provide more detailed instructions.

How to prepare to modify CTV_REPT.PBL

Before you can modify the source files, you must save the original version of the source files CTV_REPT.PBL and CTV_REPT.PBD that are installed with Resolve. Copy the files to another directory, and assign the copies different names, such as CTV_ORIG.PBL and CTV_ORIG.PBD. Then, proceed as follows:

1. Start PowerBuilder, and then select the Library painter.
2. To create a new library, from the **Library** menu, select **Create**. Name the library CTV_REPT.PBL, thereby overwriting the existing version of that file, and put it in the directory that contains the Clintrial software program files.
3. To copy each object in the original .PBL file (CTV_ORIG.PBL) to the new CTV_REPT.PBL library, from the **Entry** menu, select **Copy**. This creates a writable copy of each object in that library.
4. After you copy all of the objects, double-click on the object named “default” in your writable CTV_REPT.PBL. This action sets your default application for this session and all future PowerBuilder sessions. The Application painter opens.

How to make your changes

Select the Library painter, and open the discrepancy_form_print object. Then, proceed as follows:

1. Modify the discrepancy_form_print DataWindow. You can modify the information on this form that is determined by user preferences or protocol parameters, as explained in Chapter 19.

   For information on which changes are supported, see "Reformatting the DataWindow" on page 248.
2. Save the revised DataWindow in the CTV_REPT.PBL sources file.

Note: It is recommended that you make just a few changes at a time, and then save, rebuild, and test that version before you make another set of changes.

If you added any fields whose contents need to be retrieved, the following warning message appears when you try to save the DataWindow, or when you leave the selection window:

```
SELECT change has forced update specification change.
```

If this message appears, perform these steps:

1. From the **Rows** menu, select **Update**. A window opens.
2. Check the **Allow Updates** check box.
3. In the Table to Update field, select CTRESOLVEREF.VCT_ERRORSTATUS_UPDATE. To verify the name of an entry, click on it.
4. Some items in the Updatable Columns list will be highlighted. Do not modify them. Instead, click **Primary Key** at the right of this window. This action selects ct_recid as the Unique Key Column.
5. Click **OK**, and then save your DataWindow.

How to rebuild the run-time library

Once you complete your changes and are ready to examine the results, perform the following steps.

1. Exit the DataWindow painter, and then open the Library painter.
2. Select the CTV_REPT.PBL library that contains the modified object. Then, from the **Library** menu, select **Build Runtime Library**. A window opens.
3. Ensure that the Machine Code check box is cleared, and that the Build Type is Incremental. Then click **OK**.
4. If you are prompted to specify whether the existing CTV_REPT.PBD file can be overwritten, respond Yes.
5. Copy the resulting CTV_REPT.PBD file to the directory that contains the Clintrial software program files. Run Resolve and verify that your changes display as you intended.

Customizing discrepancy statuses

You can customize the predefined set of discrepancy statuses and the transitions allowed between them to meet the procedural needs of your data managers. This feature allows you to control the level of detail represented by the Resolve workflow, as well as the terminology used in discrepancy management. You can:

- Change the displayed names of supplied discrepancy statuses.
- Change other values for a discrepancy status.
- Add more discrepancy statuses.
- Prevent certain discrepancy statuses from being used.

You make these changes in Enter in the CTRESOLVEREF protocol.

Note: Any change made to a discrepancy status immediately affects all protocols that are set up for use with Resolve. Changes to discrepancy statuses cannot be made on a protocol-specific basis.

How to change discrepancy status names

The names of discrepancy statuses can be changed to clarify their purpose or to reflect a language other than English. To change a discrepancy status name, select the CTRESOLVEREF protocol and the DISCREP_STATE study book in Enter. Page down to find the discrepancy status record that you want to change. The following table shows the relevant fields in this study book:

Field:	**Description:**
Code	Name for display in dialog boxes and drop-down lists. ***Caution***: Do not change the values for the Code field.
Value	Name for display in dialog boxes and drop-down lists.
Label	Not used within Resolve. Accepts up to 40 characters. See the ***Note*** following this table.
Longlabel	Not used within Resolve. Accepts up to 80 characters. See the ***Note***. following this table.

For example, you can translate the description for the Ready to Send discrepancy status from English to French by entering Prêt à Envoyer in its Value field.

Note: The values in the Label and Longlabel fields are used only if you create a codelist for use in other Clintrial software modules. You can modify the default values in these fields as needed.

How to change other discrepancy status values

When Resolve is installed, each record in the DISCREP_STATE panel has a complete set of default values. In addition to using Enter to change to discrepancy status names, you can change the defaults for the following items:

Item:	**Description:**
Final?	Indicates that this is a terminal discrepancy status. Assigning a terminal status closes the discrepancy record. No changes can be made to a closed discrepancy record's values.
Propose Can Change To?	Indicates that users with Propose access can change a discrepancy record to this discrepancy status.
Propose Can Modify?	Indicates that users with Propose access can edit discrepancy records in this discrepancy status.
Allow duplicates?	Indicates that a new discrepancy record is created during validation or merging, regardless of whether a duplicate is found.
Manage Can Change To?	Indicates that users with Manage access can change a discrepancy record to this discrepancy status.
Manage Can Modify?	Indicates that users with Manage access can edit discrepancy records in this discrepancy status.

For each of these items, you enter either a 0 (zero) to indicate no, or a 1 to indicate yes. For example, a 0 for Final indicates that this is not a terminal discrepancy status, while a 1 for Allow duplicates indicates that Resolve should create a new discrepancy record even if a duplicate is already on file with this status.

For more information on access rights, see *Admin and Design*.

How to add discrepancy statuses

You can add records to the DISCREP_STATE_UPDATE table to provide additional Resolve discrepancy statuses. For example, a data manager might request a discrepancy status of Response Received to identify discrepancy records for which investigators have returned data discrepancy forms.

To define a new discrepancy status, add a record to the DISCREP_STATE study book in Enter. Specify a value for each item, including a unique Code for the new discrepancy status.

For more information, see the description of the DISCREP_STATE panel in the *Reference Guide*.

Note: After you add a discrepancy status record, you must define its allowable transitions. For more information on discrepancy transitions, see "Modifying discrepancy transitions" on page 254.

How to prevent discrepancy status use

You should never delete or change the discrepancy status codes installed with Resolve. Instead, you can prevent a discrepancy status code from being used by modifying the discrepancy transitions that provide access to it.

For example, to prevent users from using the Released discrepancy status, you modify the DISCREP_TRANSITION panel so that none of its records has an End State of Released, thereby ensuring that the status in question can never be assigned to any discrepancy record. For more information on discrepancy transitions, see "Modifying discrepancy transitions" on page 254.

Even for those discrepancy statuses that you defined yourself, deletion is not recommended. If deletion is unavoidable:

- Review all protocols set up to use Resolve to ensure that the extraneous status is not currently assigned to any discrepancy records in the update or audit table.
- Review the DISCREP_TRANSITION panel to ensure that there are no records that reference the extraneous status.

You use Enter to delete DISCREP_STATE records. For more information, see Chapter 12.

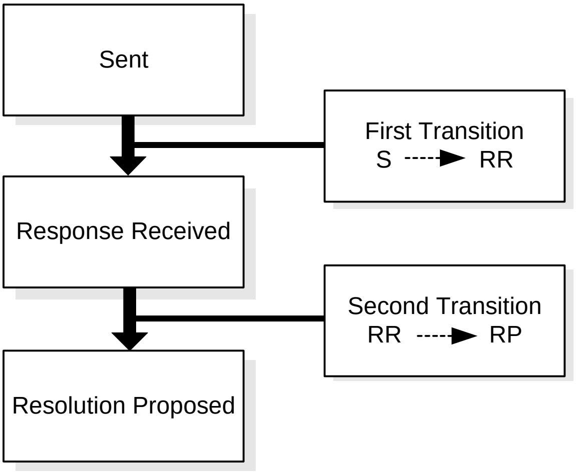
Modifying discrepancy transitions

What are discrepancy transitions?

Discrepancy transitions provide a logical structure for discrepancy status assignments. The DISCREP_TRANSITION panel in the CTRESOLVEREF protocol stores a different record for each transition permitted between discrepancy statuses. Each DISCREP_TRANSITION record contains only two items:

Item:	**Description:**
Start State	Code from DISCREP_STATE representing a discrepancy status.
End State	Code of another discrepancy status in DISCREP_STATE. Indicates that a transition from the start state to this status is allowed.

As a result, each discrepancy status will have at least one, and often several, transition records in the DISCREP_TRANSITION panel. For example, after you add the new Response Received discrepancy status, you need to add transitions to ensure that this status is assigned to discrepancy records after forms have been sent out to investigators, but before they are updated to Resolution Proposed status:



How to add a transition record

To integrate the new Response Received discrepancy status into the discrepancy management workflow, you add two different records to the CTRESOLVEREF protocol’s DISCREP_TRANSITION study book in Enter. First, add a record with S (Sent) as the Start State and RR (Response Received) as the End State. Then add a record with RR (Response Received) as the Start State and RP (Resolution Proposed) as the End State.

For a new discrepancy status to be eligible for automated discrepancy resolution, you must create transition records for its Autoclosed and Obsolete discrepancy statuses. To complete this task for the preceding example, you add a record to the CTRESOLVEREF protocol’s DISCREP_TRANSITION study book with RR (Response Received) as the Start State and ACD (Autoclosed) as the End State, then another record with RR (Response Received) as the Start State and ACM (Obsolete) as the End State.

Note: Any changes to DISCREP_TRANSITION immediately affect all protocols that are set up for use with Resolve. Transitions cannot be changed on a protocol-specific basis.

For more information on automated discrepancy resolution, see Chapter 17.

How transitions from a terminal status work

A *terminal discrepancy status* closes a discrepancy record, indicating that no further research is required. You indicate whether a discrepancy status is a terminal status in its DISCREP_STATE record. For each terminal discrepancy status, at least one DISCREP_TRANSITION record exists with that status as the End State.

You can add records to DISCREP_TRANSITION with a terminal status as the Start State. These DISCREP_TRANSITION records allow you to change closed discrepancy records to another status, indicating that they are active once again.

How to delete a transition record

To delete a predefined transition, delete the corresponding record from DISCREP_TRANSITION. Be aware that if you delete all transitions with a Start State of REITO (Reissue To), you cannot reissue discrepancy records, as explained in Chapter 17. For details about Reissue To, see the *Note* that concludes the description of the DISCREP_TRANSITION panel in the database structures section of the *Reference Guide*.

Modifying investigator retrieval instructions

Who are investigators?

An *investigator* is the person responsible for supplying clinical data, and for correcting errors in that data. Each discrepancy record has an associated investigator ID. This information is critical when you are generating data discrepancy forms, and it serves as a useful subsetting tool when you are reviewing discrepancies.

Identifying and retrieving investigator information

You can retrieve investigator information for display in:

- The summary Selected Discrepancies window
- The detailed Discrepancies # window
- Any of the Report windows that display investigators

You can retrieve investigator information for these types of display by one of the following methods:

- In Admin or Design, edit the value of the CTV_ONE_INVESTIGATOR protocol parameter to display investigator information from a context item.
- In the CT_RESOLVE_USER package, edit the CT_RESOLVE_USER.INV_ID function for more complex retrievals of investigator information.

You can retrieve investigator information for display in the Select By window or New Discrepancy window as follows:

- In Admin or Design, edit the value of the CTV_SELECT_INVESTIGA protocol parameter to display investigator information from a context item.
- In the CT_RESOLVE_USER package, edit the CT_RESOLVE_USER.INV_ID function for more complex retrievals of investigator information.

For more information on these three methods of retrieving investigator information for display, see the following sections.

How to retrieve investigator information using CTV_ONE_INVESTIGATOR

In Admin or Design, edit the CTV_ONE_INVESTIGATOR protocol parameter to display investigator information from a context item. The information is retrieved from a field in the discrepancies table (VCT_ERRORSTATUS_UPDATE).

The default value is:

```
CT_RESOLVE_USER.INV_ID(VCT_ERRORSTATUS_UPDATE.CTV_CONTEXT1’
<PROTOCOL>’)
```

This retrieves the value of the subject item.

You can make simple edits of the value of CTV_ONE_INVESTIGATOR to retrieve:

- Investigator information from other context items through a call to CT_RESOLVE_USER.INV_ID.
- Investigator information without a call to the CT_RESOLVE_USER.INV_ID function. Avoiding a call can improve performance.

Or, or you can edit the CT_RESOLVE_USER.INV_ID function for more complicated retrieval of investigator information. For information on editing this function, see "How to customize investigator retrieval using INV_ID" on page 258.

Examples: Editing CTV_ONE_INVESTIGATOR

Two examples of how to edit the value of the CTV_ONE_INVESTIGATOR protocol parameter follow:

- If the investigator ID is the first four characters of the subject item value, set this parameter to:

  ```
  SUBSTR(VCT_ERRORSTATUS_UPDATE.CTV_CONTEXT1,1,4)
  ```
- If the protocol contains a context item with investigator information, set the protocol parameter’s value to that item preceded by VCT_ERRORSTATUS_UPDATE:

```
VCT_ERRORSTATUS_UPDATE.INVNO
```

where INVNO is the investigator ID item.

If you use the CTV_ONE_INVESTIGATOR protocol parameter to retrieve investigator information, you can retrieve information only from a context item.

How to customize investigator retrieval using INV_ID

You can customize the CT_RESOLVE_USER.INV_ID function for more complex retrievals of investigator information; for example, using items other than context items.

If you edit the INV_ID function you ensure the retrieval of useful investigator information for all windows that display investigator information, but the call to the package may slow performance.

Typically, all PL/SQL function and procedures used by the Clintrial software run under CTPROC, which is a privileged Oracle account. Rather than making any changes to CTPROC, you should therefore copy the files you need, modify the copies, and then move the copied files to the CTSITEPROC account.

Caution: The CT_RESOLVE_USER.INV_ID function supplied with the Clintrial Release 4.5 software contains changes that must be used with this release. If you have customized the INV_ID function in previous releases of the Clintrial software, you must merge your changes into the Clintrial Release 4.5 software CT_RESOLVE_USER package in one of the following ways:

- Start with the Clintrial software-supplied CT_RUSER.SQH and CT_RUSER.SQL files and include any changes that you have made to the INV_ID function or any other functions.
- Start with your customized versions of the previous release’s files and add the declaration of on_status_change to your .SQH file and the body of the on_status_change to your .SQL file.

To customize the modified copies for use with your CTSITEPROC account, do as follows:

1. On the server, copy CT_RUSER.SQH (the header file) and CT_RUSER.SQL, so that you can edit those files without modifying the originals. For example, copy CT_RUSER.SQH to CT_RUSEX.SQH, and copy CT_RUSER.SQL to CT_RUSEX.SQL. These files reside with the other Resolve server files in the CT43_CTV subdirectory.
2. Remove the last line from CT_RUSEX.SQH, which is as follows:

   ```
   CREATE PUBLIC SYNONYM ct_resolve_user FOR ctproc.ct_resolve_user;
   ```

3. Edit CT_RUSEX.SQL to rewrite INV_ID, so that given a subject item and protocol name, it will return an investigator ID. Information about how to set up CTSITEPROC is included in the documentation of the sample study shipped with the Clintrial software.
4. To load and test your new routines in CTSITEPROC, use SQL*Plus to connect to the CTSITEPROC account. Enter the following commands to load your header and procedure files:

   ```
   SQL> @ct_rusex.sqh
   SQL> @ct_rusex.sql
   ```

5. If you find errors in CT_RUSEX.SQL, fix them with your text editor and reload the file. (You do not need to reload CT_RUSEX.SQH in CTSITEPROC to reload CT_RUSEX.SQL.)
6. After your files are in place, change the public synonym for CT_RESOLVE_USER from within SQL*Plus (connected as CTSITEPROC), and grant access to it by typing:

   ```
   SQL> drop public synonym ct_resolve_user;
   SQL> create public synonym ct_resolve_user for ctsiteproc.ct_resolve_user;
   SQL> grant execute on ct_resolve_user to ct_user;
   ```

7. To ensure that CTSITEPROC can execute all the routines in CT_RESOLVE_USER, find the AF_RUSER.SQL server script (in the sample study on the CD). If your site has changed the password for the Oracle account CTPROC, copy and edit AF_RUSER.SQL to include the correct password. From SQL*Plus, load AF_RUSER.SQL using the syntax:

   ```
   SQL> @af_ruser.sql
   ```

The following sections show examples of the customized.

Example: Static SQL SELECT statements in INV_ID

This example uses the following static SQL SELECT statements to reference each protocol's tables:

```
FUNCTION inv_id(i_context1 IN VARCHAR2, i_protocol IN VARCHAR2)

RETURN VARCHAR2

IS

invid VARCHAR2(80);

l_invnum VARCHAR2(10);

l_surname VARCHAR2(30);

l_firstname VARCHAR2(30);
```

```
        /* put more declarations here, if any needed */
BEGIN
/* To extract characters from the context1 argument, use
 * SUBSTR. For example, to start at the 5th character and
 * extract 3 characters, it would be SUBSTR(i_context1, 5, 3);
*/
IF upper(i_protocol) = 'MEDIKA_CLINICAL' THEN
SELECT INVNUM, SURNAME, FIRSTNAME INTO l_invnum, l_surname,
l_firstname
FROM MEDIKA_CLINICAL.INVESTIGATORS_ALL
WHERE ROWNUM < 2 AND SUBJECT = i_context1;
IF l_invnum IS NULL OR l_surname IS NULL OR l_firstname IS NULL
THEN invid := i_context1;
ELSE invid := l_invnum || ' ' || l_surname  || ', ' || l_firstname;
END IF;


RETURN invid;
ELSE
RETURN  i_context1 ;
END IF;
EXCEPTION
WHEN OTHERS THEN RETURN i_context1 ;
END inv_id;
```

Note: The tables referred to in a static SQL SELECT statement must exist at the time you enter the code into SQL*Plus. Also, CTSITEPROC must have SELECT access to those tables.

Example: Investigator ID in Subject Item string

In this example, the investigator ID is part of the subject item string. For example, the string 0004 033 00118 identifies an investigator ID of 0004 (the first four characters):

```
FUNCTION inv_id(i_context1 IN VARCHAR2, i_protocol IN VARCHAR2)
RETURN VARCHAR2
IS
l_investigator VARCHAR2(40);
BEGIN
/* To extract characters from the context1 argument, use
SUBSTR. For example, to start at the 5th character and
extract 3 characters, it would be SUBSTR(i_context1, 5, 3);
*/
IF i_context1 IS NULL THEN
l_investigator := NULL;
ELSE
l_investigator := SUBSTR(i_context1, 1, 4);
END IF;
RETURN l_investigator;
EXCEPTION
/* Execute this code when any error occurs. Here we
choose to return an error string.
We could choose to return NULL instead.
*/
WHEN OTHERS THEN
l_investigator := 'INV_ID:ERROR';
RETURN l_investigator;
END inv_id;
```

How to retrieve investigator information using CTV_SELECT_INVESTIGA

In Admin or Design, edit the CTV_SELECT_INVESTIGA protocol parameter to display investigator information from a context item in the Select By and New Discrepancies windows. The information is retrieved from a field in the discrepancies table (VCT_ERRORSTATUS_UPDATE).

The default value is:

```
SELECT UNIQUE
CT_RESOLVE_USER.INV_ID(<SUBJECT_ITEM>,'<PROTOCOL>') FROM
<PROTOCOL>.SUBSTITUTE*PANEL_SUBSTITUTE*DBTABLE
```

This retrieves the value of the subject item.

Note: The CT_RESOLVE_USER.INV_ID function defines the relationship between the subject and the investigator for this default protocol parameter. As supplied with the Clintrial software, this function runs in the CTPROC account and retrieves the value of the subject item to display as the investigator information.

Each parameter’s value reflects the actual protocol and subject item name. For example, for protocol MEDIKA_CLINICAL with subject item SUBID, the value in CTV_SELECT_INVESTIGA used at run time is:

```
SELECT UNIQUE CT_RESOLVE_USER.INV_ID(SUBID, 'MEDIKA_CLINICAL')
FROM MEDIKA_CLINICAL.SUBSTITUTE*PANEL_SUBSTITUTE*DBTABLE
```

The SUBSTITUTE* occurrences are replaced at runtime. For the **Discrepancy** menu’s **New** command, Resolve uses the panel and table selected in other dialog box lists. For the **View** menu’s **Select By** command, VCT_ERRORSTATUS replaces SUBSTITUTE*PANEL, and UPDATE replaces SUBSTITUTE*DBTABLE.

You can make simple edits of the value of CTV_SELECT_INVESTIGA to retrieve investigator information from other context items through a call to CT_RESOLVE_USER.INV_ID.

Or, you can edit the CT_RESOLVE_USER.INV_ID function for more complicated retrieval of investigator information. For information on editing this function, see "How to customize investigator retrieval using INV_ID" on page 258.

21 Preparing to Track Discrepancies

Overview

This chapter explains how:

- Validation automates the creation of discrepancy records in Resolve.
- You can write rules and derivations to maximize the information available for discrepancy records.
- You can implement rules to create discrepancy records from flags.
- Advanced users can create discrepancy records in nonstandard situations.

This chapter is intended for study designers or those responsible for panel design, and the PL/SQL programmers who write rules. It assumes that you are familiar with rules and derivations, as explained in the *Design* section of *Admin and Design*, and data validation, as explained in the Manage section of *Manage, Classify, and Lab Loader.* This chapter also assumes that you are familiar with PL/SQL, as described in your Oracle and PL/SQL program documentation.

Validation and discrepancy records

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 in Manage that validates clinical data before it is moved to the data table (merged). During validation, rules evaluate to TRUE or FALSE.

Note: PL/SQL is Oracle’s procedural extension language to the SQL database language.

Before you installed Resolve, you may have used Design to write rules for automated data validation. When rules for data validation run in Manage, the MANAGE_DISCREPANCY function adds discrepancy records in Resolve.

How MANAGE_DISCREPANCY works

The MANAGE_DISCREPANCY function is called automatically from the panel’s validation procedure when validation runs in Manage. This function compares current validation results to determine if new discrepancy records should be created. To create a new discrepancy record, MANAGE_DISCREPANCY creates a record with an error type of VALIDATE in the protocol’s VCT_ERRORSTATUS_UPDATE table. The

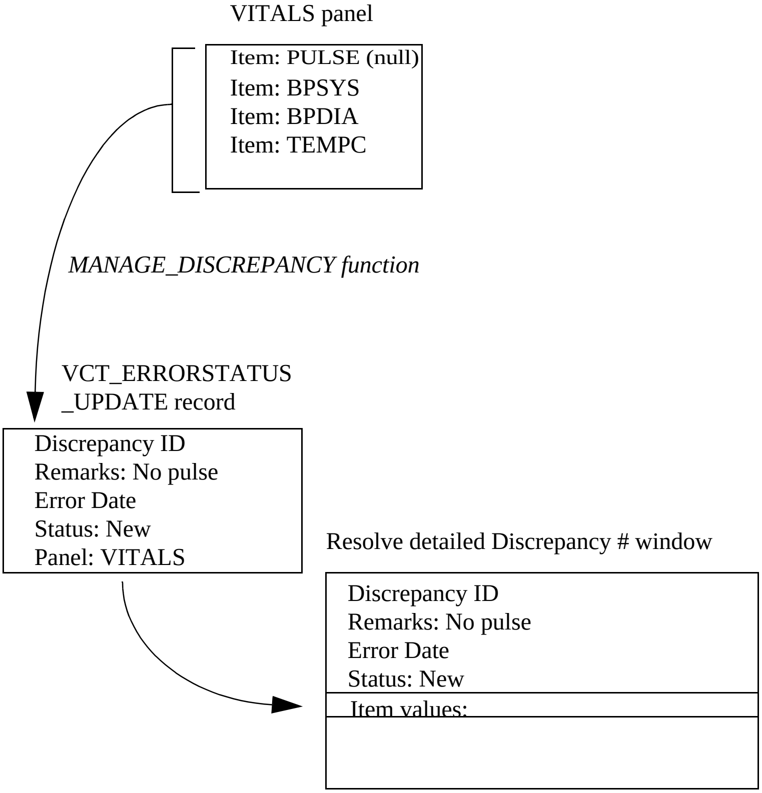
VCT_ERRORSTATUS panel is copied from the CTRESOLVEREF protocol to each protocol that is set up to use Resolve. Records in VCT_ERRORSTATUS contain the essential information for Resolve discrepancy records.

Before creating new discrepancy records, however, MANAGE_DISCREPANCY reviews each record that undergoes validation to find whether it returned TRUE or FALSE for each rule. It then checks whether a discrepancy record with an error type of VALIDATE and the same panel, ct_recid, and rule_name already exists. Based on the results of this review, MANAGE_DISCREPANCY takes one of the following actions:

TRUE or FALSE:	**Open Duplicate?**	**Action:**
FALSE	No	Create a new discrepancy record.
FALSE	Yes	None.
FALSE	Yes; however, error item value(s) do not match.	Assign the Autoclosed status to the discrepancy record, and create a new discrepancy record.
TRUE	No	None.
TRUE	Yes	Assign the Autoclosed status to the discrepancy record. This occurs, for example, when a value is edited in Enter to correct a discrepancy.

The following figure shows how MANAGE_DISCREPANCY creates a VCT_ERRORSTATUS_UPDATE record, which is displayed as a discrepancy record in a Resolve detailed Discrepancy # window:

Rule: Pulse should not be null on the VITALS panel.



Note: Discrepancy records also are created automatically for records that fail merging. When a record fails merging in Manage, MANAGE-_DISCREPANCY creates a new VCT_ERRORSTATUS record.

How to write rules

Rules are written in Design for a specific protocol and panel. When creating rules in Design, you can either enter PL/SQL directly, or call PL/SQL procedures that you have stored in PL/SQL packages in the Oracle database. Examples of both options are included in this section.

Refer to the *Design* section of *Admin and Design* for general information on writing rules. Once your rules are written, you execute them in Manage when you validate records.

How to use derivations to include more data

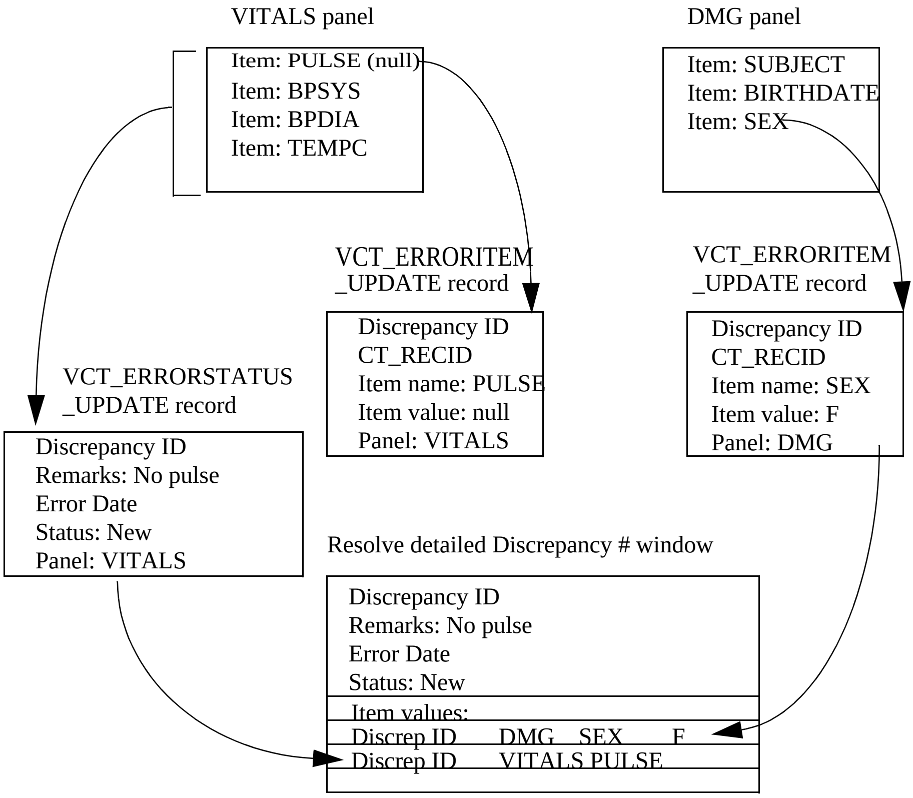
A *derivation* is a PL/SQL statement that is attached to a panel and calculates the value of an item or a temporary variable for records during validation. Derivations are part of the validation procedure that checks clinical data before it is moved to the data table. If these temporary variables contain item names and values that are relevant to the type of discrepancy detected by a rule, you may prompt MANAGE_DISCREPANCY to include that data in the discrepancy records it creates by embedding the names and values in calls to SETUP_ERRORITEM.

Note: You can include items from multiple panels.

The item names and values that are provided to SETUP_ERRORITEM by derivations are stored in VCT_ERRORITEM_UPDATE records. Each VCT_ERRORITEM record references a specific record in VCT_ERRORSTATUS; together, the VCT_ERRORSTATUS record and all of its associated VCT_ERRORITEM records make up a Resolve discrepancy record that can help data managers resolve discrepancies efficiently.

The following figure shows how a rule can create a VCT_ERROR-STATUS record, and how a derivation that includes a specific reference to that rule can save item values from data records as VCT_ERRORITEM records:

Rule: Pulse should not be null on the VITALS panel.
Derivation: For this rule, include the items SEX and BIRTHDATE from the DMG panel and PULSE from the VITALS panel.



You can also write a rule that calls a separately stored PL/SQL package to achieve the same result.

If you choose not to write rules or derivations that add VCT_ERROR-ITEM records, discrepancy records are still created in Resolve during validation by the MANAGE_DISCREPANCY function. In this case, discrepancy records do not have any VCT_ERRORITEM records associated with the VCT_ERRORSTATUS record.

Note: Derivations cannot be specified for merge processing. Discrepancy records with an error type of MERGE do not have associated error items.

How to write derivations

Derivations are written in Design for a specific protocol and panel. When you initiate validation in Manage, derivations are executed before rules.

For general information on derivations, see the *Design* section of *Admin and Design*.

Example rules and derivations

Rules are PL/SQL fragments that evaluate to TRUE or FALSE. Some common methods for writing rules for use in Resolve are:

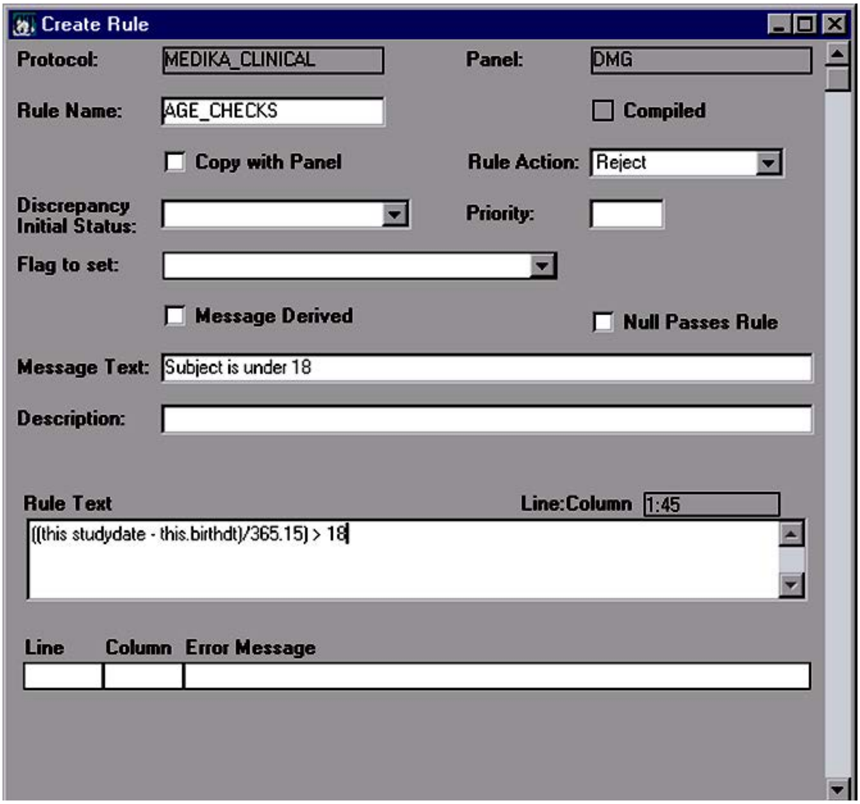
- Create the PL/SQL error logic in the rule’s Rule Text attribute.
- Create the PL/SQL error logic in the Rule Text, and create a related derivation that identifies a relevant item or items by calling the CTV_CORE.SETUP_ERRORITEM procedure.
- Create the PL/SQL error logic and the call to SETUP_ERRORITEM in a derivation. Include PL/SQL that passes a message to a related rule if the error logic detects an error. If the rule receives a message, it evaluates to FALSE and the rule fails validation.
- Create the error logic and the call to SETUP_ERRORITEM in a user-defined PL/SQL package. The rule only passes values to the function or procedure in the package.

The following sections show examples of these different methods from the MEDIKA_CLINICAL sample protocol.

Example: Rule with no related derivation

You can create a rule that contains the error logic, and not plan to have any relevant items supplied to the discrepancy record by a derivation that calls the CTV_CORE.SETUP_ERRORITEM procedure.

The following example shows the AGE_CHECKS rule in MEDIKA_CLINICAL's DMG panel:



Protocol: MEDIKA_CLINICAL

Panel: DMG

Rule Name: AGE_CHECKS

□ Compiled

□ Copy with Panel

Rule Action: Reject

Discrepancy Initial Status:

Priority:

Flag to set:

□ Message Derived

□ Null Passes Rule

Message Text: Subject is under 18

Description:

Rule Text

Line:Column 1:45

```
((this studydate - this.birthdt)/365.15) > 18
```

Line	Column	Error Message

When validation runs, MANAGE_DISCREPANCY reviews the records that return FALSE for this rule, and creates VCT_ERRORSTATUS records for those records that do not already have them.

Optionally, when you create or modify a rule, you can specify the discrepancy status and priority to use for any resulting discrepancy records. If you do not specify values in the Discrepancy Initial Status and Priority fields of Design's Modify Rule dialog box, the discrepancy record is created in New status and with a null priority.

Note: The AGE_CHECKS rule does not appear in MEDIKA_CLINICAL. Instead, the protocol uses the AGE_CHECK rule and the AGE_CHECK derivation, as described in "Example: Derivation with error logic and related rule" on page 275.

Writing a derivation for Resolve

When you install Resolve, the CTV_CORE.SETUP_ERRORITEM PL/SQL procedure is supplied. This procedure saves information that can be used later to create VCT_ERRORITEM records to enhance the discrepancy records created by MANAGE_DISCREPANCY.

The format for calling the SETUP_ERRORITEM function from a derivation is as follows:

CTV_CORE.SETUP_ERRORITEM(*rule_name, panel, item_name, ct_recid, item_value, repeat_id*)

The first argument, *rule_name*, indicates an existing rule. The other arguments can refer to the same data record currently being processed, or to any other record in the protocol. Additional data, such as the SYSDATE or values assigned by derivation, can also be referenced. These arguments refer to any item in a record related to the discrepancy, as follows:

- *panel* can be the same cts$panel, or the name of another panel.
- *item_name* is the name of the item in the given panel involved in the discrepancy. The item_name must be in UPPERCASE or duplicate entries occur in Resolve with the previous record marked as Auto closed after each validation run.
- validation run.
- in Resolve with the previous record marked as Auto closed after each
- validation run.
- *ct_recid* can be this.ct_recid for the current record, or another ID number.
- *item_value* is the value, if any, stored for the item. It can be text, a number, or a date.
- *repeat_id* is optional text used to provide contextual information for the item.

Note: Only the last two arguments described above (*item_value* and *repeat_id*) can be set to null.

The Clintrial software is delivered with system-defined variables that you can use within a rule or derivation. In addition, whenever you want to refer to an item in the current record, you can use the identifier ‘this.’ Therefore, you can supply several arguments in the calling sequence with the following variables:

Value:	**Variable or Reference:**	**Oracle type:**
panel	cts$panel	VARCHAR2
ct_recid	this.ct_recid	VARCHAR2

Value:	Variable or Reference:	Oracle type:
item_value	this.*item_value_name*	Same type as the Clintrial software item: VARCHAR2, NUMBER, or DATE

For information on system-defined variables and references to the current record, see the *Design* section of *Admin and Design*.

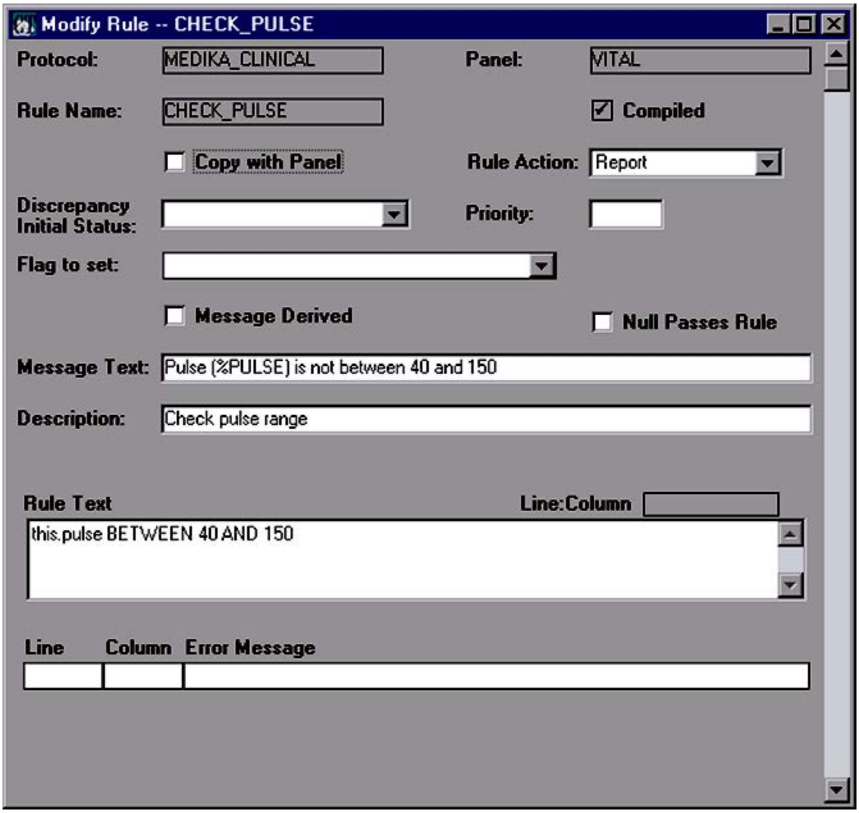
To complement the rule in the previous example and enhance the information stored with its resulting discrepancy records, you can add a derivation. The derivation uses the AGE_CHECK rule as the first argument to SETUP_ERRORITEM in each call. That way, the studydate and birthdt item values are included in the discrepancy records that are created when the AGE_CHECK rule fails, as follows:

```
CTV_CORE.SETUP_ERRORITEM(‘AGE_CHECK’, cts$panel, ‘STUDYDATE’,
this.ct_recid, this.studydate, ‘’);

CTV_CORE.SETUP_ERRORITEM(‘AGE_CHECK’, cts$panel, ‘BIRTHDT’,
this.ct_recid, this.birthdt, ‘’);
```

Example: Rule with error logic and related derivation

You can create a rule that contains the error logic and a related derivation that identifies the relevant item or items that the CTV_CORE.SETUP_ERRORITEM procedure supplies to the error item section of the detailed Discrepancy # window.

The following example shows the CHECK_PULSE rule in MEDIKA_CLINICAL's VITAL panel:



Modify Rule -- CHECK_PULSE

Protocol: MEDIKA_CLINICAL

Panel: VITAL

Rule Name: CHECK_PULSE

☑ Compiled

□ Copy with Panel

Rule Action: Report

Discrepancy Initial Status:

Priority:

Flag to set:

□ Message Derived

□ Null Passes Rule

Message Text: Pulse (%PULSE) is not between 40 and 150

Description: Check pulse range

Rule Text

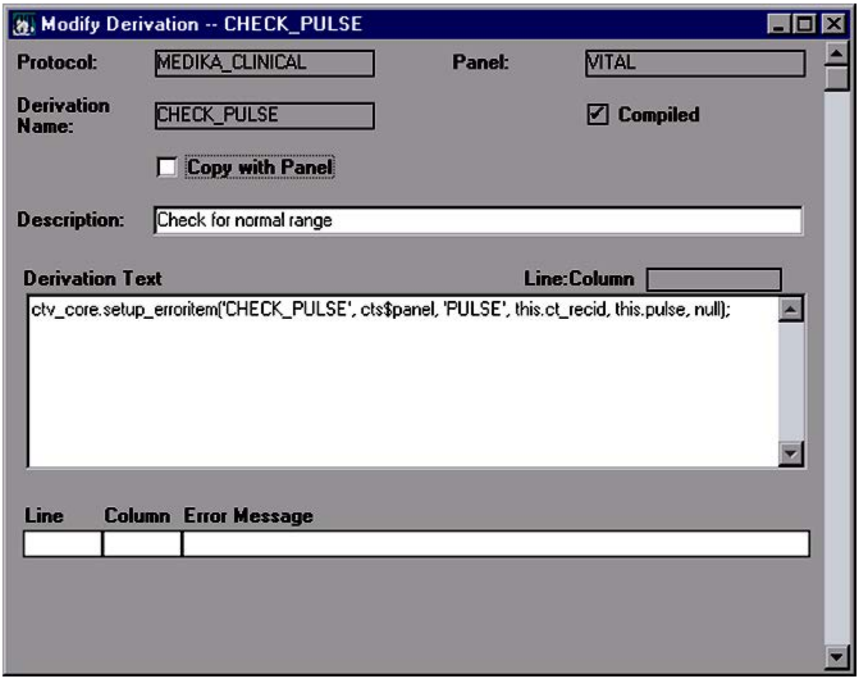
Line:Column

```
this.pulse BETWEEN 40 AND 150
```

Line	Column	Error Message

The CHECK_PULSE rule checks that the pulse is between 40 and 150. If the pulse is out of range, the rule supplies a message to the discrepancy record that gives the value that is out of range (%PULSE).

The following example shows the related derivation:



Protocol: MEDIKA_CLINICAL

Panel: VITAL

Derivation Name: CHECK_PULSE

☑ Compiled

□ Copy with Panel

Description: Check for normal range

```
ctv_core.setup_erroritem('CHECK_PULSE', cts$panel, 'PULSE', this.ct_recid, this.pulse, null);
```

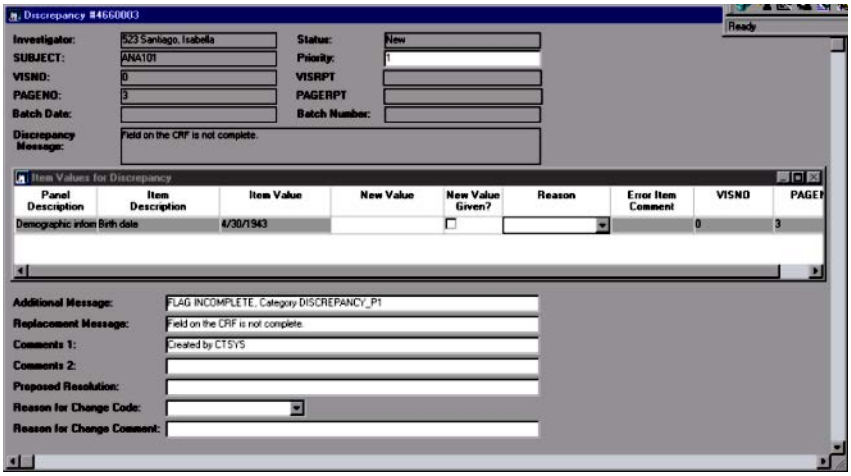
Line	Column	Error Message

The CHECK_PULSE derivation adds the relevant item to the error item record, referring to the rule CHECK_PULSE and the item PULSE.

When validation runs in Manage, a Resolve discrepancy record is created if it meets the criteria of the MANAGE_DISCREPANCY function. The item indicated in the derivation is included in the resulting discrepancy record, as shown in the following detailed Discrepancy # window:

VCT_ERRORSTATUS record

VCT_ERRORITEM record added by derivation



Discrepancy #4660003

Investigator: 523 Santiago, Isabella
SUBJECT: ANA101
VISNO: 0
PAGENO: 3
Batch Date:
Discrepancy Message: Field on the CRF is not complete.

Status: New
Priority: 1
VISRPT
PAGERPT
Batch Number:

Item Values for Discrepancy

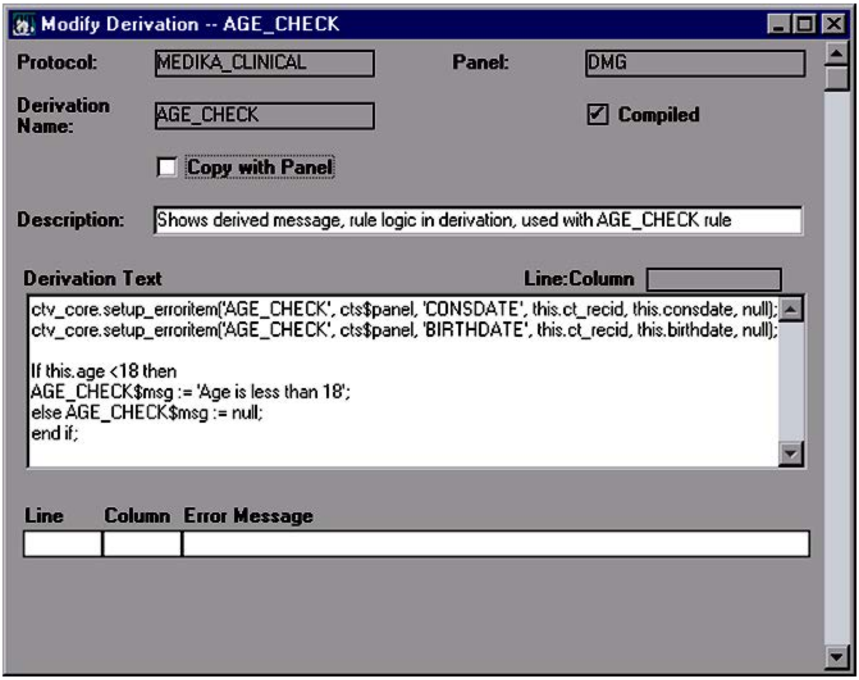
Panel Description	Item Description	Item Value	New Value	New Value Given?	Reason	Error Item Comment	VISNO	PAGE
Demographic infom	Birth date	4/30/1943		□			0	3

Additional Message: FLAG INCOMPLETE, Category DISCREPANCY_P1
Replacement Message: Field on the CRF is not complete.
Comments 1: Created by CTSYS
Comments 2:
Proposed Resolution:
Reason for Change Code:
Reason for Change Comment:

Example: Derivation with error logic and related rule

You can create a derivation that contains the error logic and identifies the relevant item or items that the CTV_CORE.SETUP_ERRORITEM procedure supplies to the error item section of the detailed Discrepancy # window. The related rule passes or fails based on whether the information supplied by the derivation evaluates to true or false.

The following example shows the AGE_CHECK derivation in MEDIKA_CLINICAL's DMG panel:



Modify Derivation -- AGE_CHECK

Protocol: MEDIKA_CLINICAL

Panel: DMG

Derivation Name: AGE_CHECK

☑ Compiled

□ Copy with Panel

Description: Shows derived message, rule logic in derivation, used with AGE_CHECK rule

Derivation Text

Line:Column

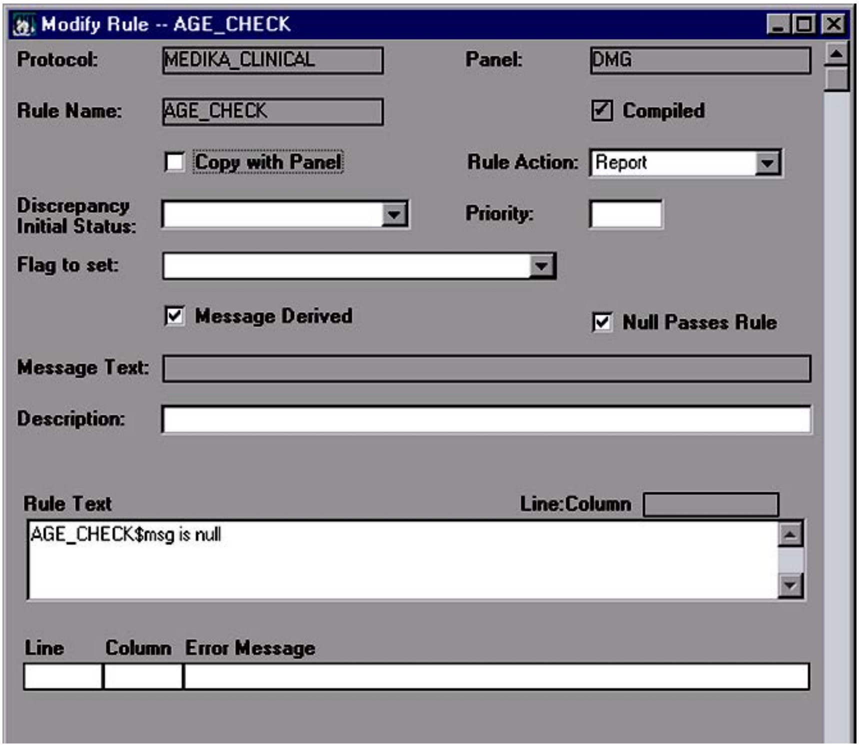
```
ctv_core.setup_erroritem('AGE_CHECK', cts$panel, 'CONSDATE', this.ct_recid, this.consdate, null);
ctv_core.setup_erroritem('AGE_CHECK', cts$panel, 'BIRTHDATE', this.ct_recid, this.birthdate, null);

If this.age <18 then
AGE_CHECK$msg := 'Age is less than 18';
else AGE_CHECK$msg := null;
end if;
```

Line	Column	Error Message

The AGE_CHECK derivation adds the relevant items to the error item record, referring to the rule AGE_CHECK and the items CONSDATE and BIRTHDATE. The derivation also contains the error logic, and passes to the rule a null message or the message that 'Age is less than 18'.

Resolve

The following example shows the AGE_CHECK rule in MEDIKA_CLINICAL's DMG panel:



Modify Rule -- AGE_CHECK

Protocol: MEDIKA_CLINICAL

Panel: DMG

Rule Name: AGE_CHECK

☑ Compiled

□ Copy with Panel

Rule Action: Report

Discrepancy Initial Status:

Priority:

Flag to set:

☑ Message Derived

☑ Null Passes Rule

Message Text:

Description:

Rule Text

Line:Column

```
AGE_CHECK$msg is null
```

Line	Column	Error Message

The rule receives the AGE_CHECK$msg value that was calculated by the related AGE_CHECK derivation. If the value is null, the rule evaluates to TRUE, and the record passes validation. If the value contains a message message ('Age is less than 18'), the rule evaluates to FALSE, and the record fails validation. The Message Derived attribute is checked, because the message text is in the derivation.

Example: Rule calling user-defined PL/SQL package for error and error items

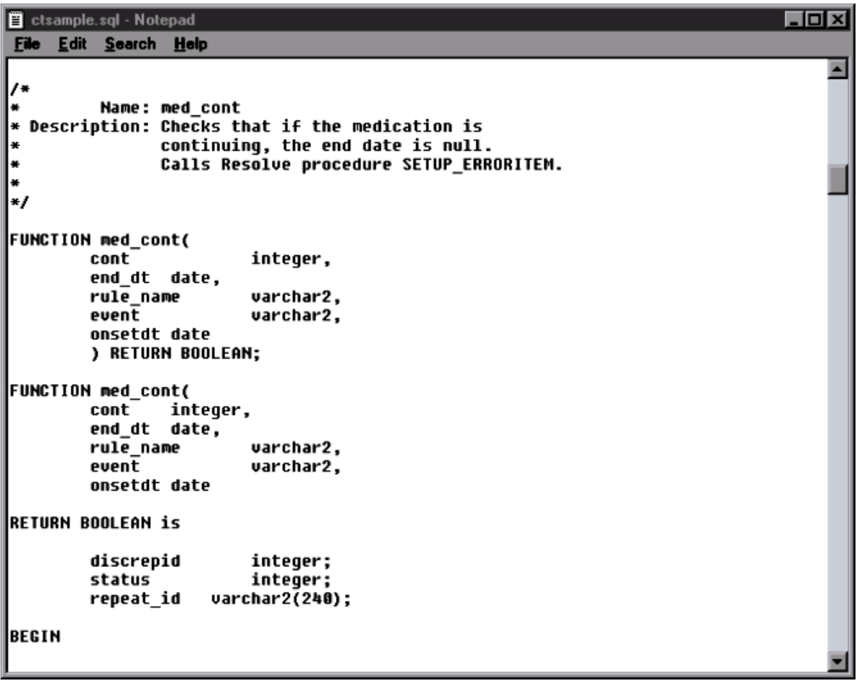
You can write a rule whose Rule Text contains PL/SQL that calls a user-defined PL/SQL package that:

- Handles the error logic.
- Identifies the relevant item or items that the CTV_CORE.SETUP_ERRORITEM procedure supplies to the error item section of the detailed Discrepancy # window.

In the following example the CT_SAMPLE package that is supplied with the Medika sample study contains a PL/SQL function named med_cont. This function:

- Contains the error logic to check that if the medication is continuing, the end date is null.
- Calls the CTV_CORE.SETUP_ERRORITEM procedure, and defines the values in the calling sequence that are not replaced by variables.

The following example shows the comments in the PL/SQL package that describe the med_cont function, its forward declaration, arguments, and returned values:



```
/*
*        Name: med_cont
* Description: Checks that if the medication is
*              continuing, the end date is null.
*              Calls Resolve procedure SETUP_ERRORITEM.
*
*/

FUNCTION med_cont(
        cont            integer,
        end_dt  date,
        rule_name       varchar2,
        event           varchar2,
        onsetdt date
        ) RETURN BOOLEAN;

FUNCTION med_cont(
        cont    integer,
        end_dt  date,
        rule_name       varchar2,
        event           varchar2,
        onsetdt date

RETURN BOOLEAN is

        discrepid       integer;
        status          integer;
        repeat_id   varchar2(240);

BEGIN
```

Comments

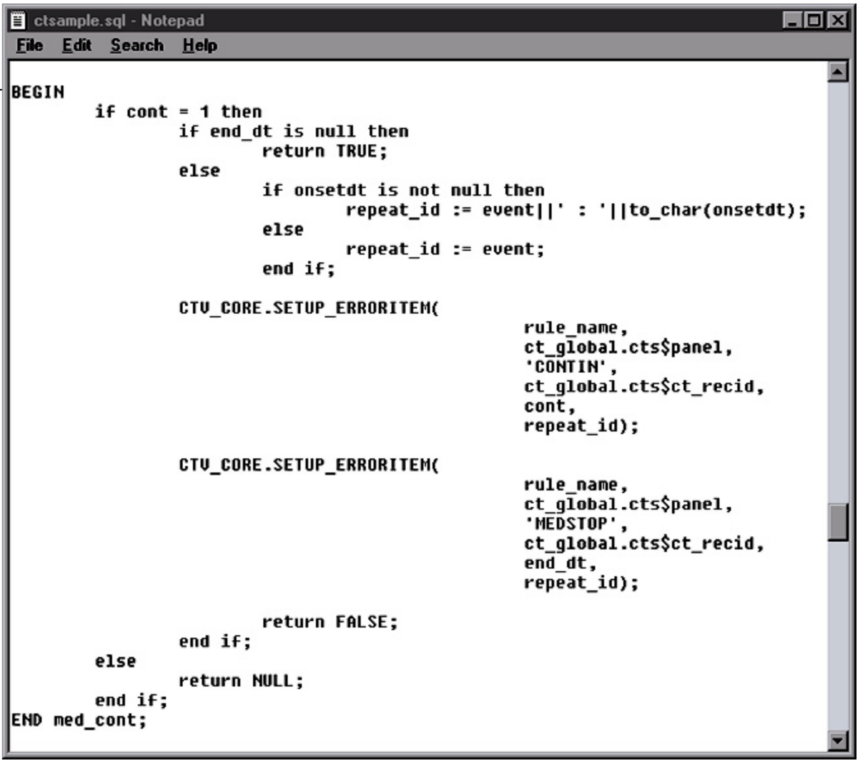
Forward declaration

Argument list

Returned values

Continued in next figure...

The following example shows the function definition:

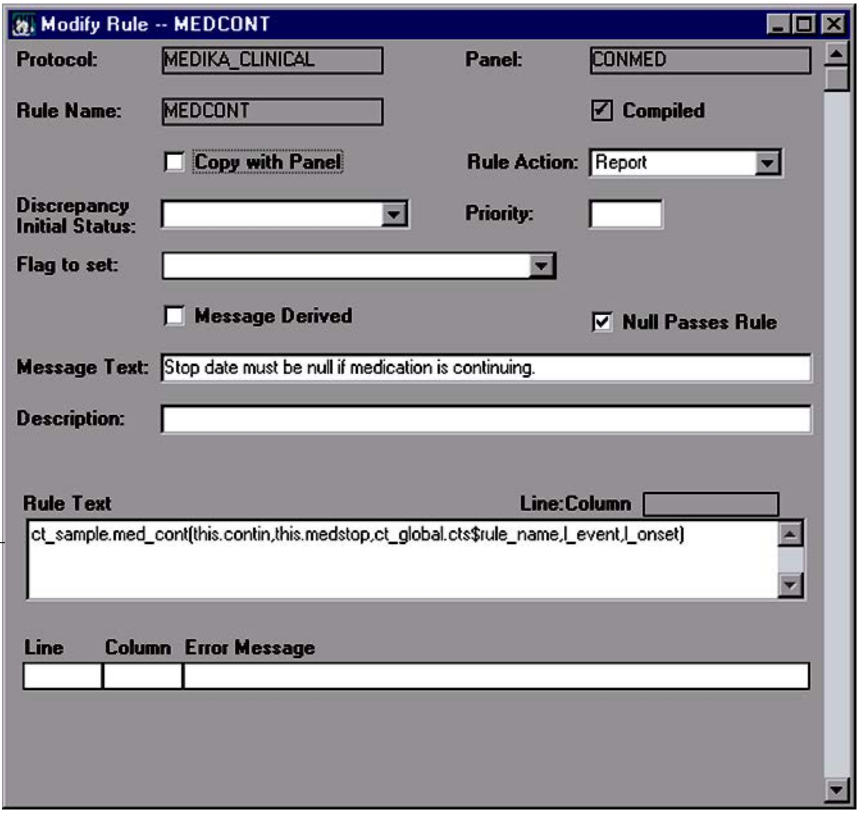


Function definition

```
BEGIN
        if cont = 1 then
                if end_dt is null then
                        return TRUE;
                else
                        if onsetdt is not null then
                                repeat_id := event||' : '||to_char(onsetdt);
                        else
                                repeat_id := event;
                        end if;

                CTV_CORE.SETUP_ERRORITEM(
                                                rule_name,
                                                ct_global.cts$panel,
                                                'CONTIN',
                                                ct_global.cts$ct_recid,
                                                cont,
                                                repeat_id);

                CTV_CORE.SETUP_ERRORITEM(
                                                rule_name,
                                                ct_global.cts$panel,
                                                'MEDSTOP',
                                                ct_global.cts$ct_recid,
                                                end_dt,
                                                repeat_id);

                        return FALSE;
                end if;
        else
                return NULL;
        end if;
END med_cont;
```

Note: You cannot reference ‘this’ or any cts$values in PL/SQL functions or procedures; they are accessible only in rules and derivations. However most of the cts$values are accessible through references to them as ct_global_cts$variables. The names are the same with the addition of the prefix ct_global.

In MEDIKA_CLINICAL's CONMED panel, the MEDCONT rule calls the PL/SQL function med_cont:



Protocol: MEDIKA_CLINICAL

Panel: CONMED

Rule Name: MEDCONT

☑ Compiled

□ Copy with Panel

Rule Action: Report

Discrepancy Initial Status:

Priority:

Flag to set:

□ Message Derived

☑ Null Passes Rule

Message Text: Stop date must be null if medication is continuing.

Description:

Rule Text

Line:Column

```
ct_sample.med_cont(this.contin,this.medstop,ct_global.cts$rule_name,l_event,l_onset)
```

Line	Column	Error Message

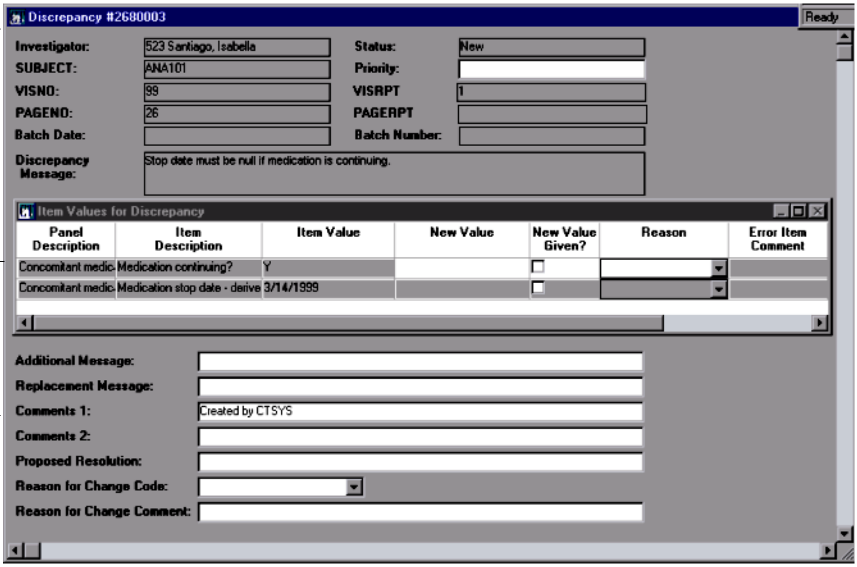
Rule text referring to PL/SQL function

When validation runs in Manage, a Resolve discrepancy record is created if it meets the criteria of the MANAGE_DISCREPANCY function:

VCT_ERRORSTATUS record

VCT_ERRORITEM records

VCT_ERRORSTATUS record (cont.)



Investigator:	523 Santiago, Isabella	Status:	New
SUBJECT:	ANA101	Priority:	
VISNO:	99	VISRPT	1
PAGENO:	26	PAGERPT	
Batch Date:		Batch Number:	

Discrepancy Message: Stop date must be null if medication is continuing.

Panel Description	Item Description	Item Value	New Value	New Value Given?	Reason	Error Item Comment
Concomitant medic	Medication continuing?	Y		□		
Concomitant medic	Medication stop date - derive	3/14/1999		□		

Additional Message:
Replacement Message:
Comments 1: Created by CTSYS
Comments 2:
Proposed Resolution:
Reason for Change Code:
Reason for Change Comment:

Resolve and flags

As data-entry operators enter clinical data in Enter, they can use flags to indicate suspected errors. In addition to the rules that test records and create discrepancy records, you can also write rules that detect flags attached to data and create discrepancy records for them. To use this feature, you must use Design to add one or more Resolve flag categories and to write rules that detect flags in these categories.

You can define specific flag names for each new flag category, such as Comments illegible or Missing data.

For details on adding flag categories, see the Help.

Types of Resolve flag categories

When you add flag categories to automate the creation of discrepancy records, each flag category name must begin with DISCREPANCY_P. You can also add a numeric suffix to indicate the priority code to assign when discrepancy records are created. For example:

Category:	**Priority:**
DISCREPANCY_P1	1
DISCREPANCY_P2	2
DISCREPANCY_P3	3

You can define one flag category for each priority level, as shown in the preceding example, or reuse the same one repeatedly.

What are flag to discrepancy rules?

A PL/SQL function for detecting discrepancy flags is supplied when you install Resolve. The CTV_CORE.FLAG_TO_DISCREPANCY function searches data for any discrepancy flags with a category name that begins with DISCREPANCY_P. When such a flag is detected, the FLAG_TO-_DISCREPANCY function creates a new discrepancy record in Resolve.

Note: FLAG_TO_DISCREPANCY uses the same process as MANAGE_DISCREPANCY to determine whether a duplicate discrepancy exists and should be closed before a new record is created. FLAG_TO_DISCREPANCY produces and uses discrepancies whose error type is FLAG.

The calling sequence for the FLAG_TO_DISCREPANCY function is:

result := CTV_CORE.FLAG_TO_DISCREPANCY(*protocol, panel, Oracle_table, ct_recid, dstatus, rule_name*);

protocol is the current protocol.

panel is the current panel.

Oracle_table is either UPDATE or DATA.

ct_recid is the CT_RECID of the record that is checked for a discrepancy flag.

dstatus is a Resolve discrepancy status code to assign if a discrepancy record is created.

rule_name is the name of a specific rule. When this function is called from a rule, FLAG_TO_DISCREPANCY uses the current rule name if you leave this argument undefined.

In a rule, these values can be replaced by the following system-defined variables and current record references:

Value:	**Variable or Reference:**	**Oracle type:**
protocol	cts$protocol	VARCHAR2
panel	cts$panel	VARCHAR2
Oracle_table	cts$table	VARCHAR2 (‘UPDATE’ or ‘DATA’)
ct_recid	this.ct_recid	VARCHAR2
rule_name	cts$rule_name	VARCHAR2

For *dstatus*, you can supply any discrepancy status code from the DISCREP_STATE panel. Enclose it within single quotation marks; for example, ‘REL’. Codes are case-sensitive and must be entered accurately. If null or invalid, the status selected at the rule level will be supplied. If no status is selected for the rule, N (New) is supplied.

For *rule_name*, you can supply a specific rule name, such as ‘AGE_CHECK’. If null and called from a rule, the current rule name (cts$rule_name) is used. As a result, your rule may resemble the following example:

```
CTV_CORE.FLAG_TO_DISCREPANCY(cts$protocol, cts$panel, cts$table,
this.ct_recid, ‘REL’, cts$rule_name) IS NOT NULL
```

Alternatively, a shorter call to FLAG_TO_DISCREPANCY that provides the current protocol, panel, table, ct_recid, and rule name may be used from within a rule, as follows:

```
result := CTV_CORE.FLAG_TO_DISCREPANCY;
```

When a VCT_ERRORSTATUS record is created as the result of a call to FLAG_TO_DISCREPANCY, the data for the discrepancy record includes:

Item:	**Supplied value:**
Additional Message	Comments in flag
Discrepancy Message	Flag name description
Error Action	Null
Error Type	FLAG
Priority	Number in the flag category name; NULL if the flag category name is DISCREPANCY_P
Rule Name	Rule name provided or the name of the current rule
Source	FLAG

How flag types work for discrepancy records

When you attach any flag to clinical data in Enter, you choose a flag type of item, record, or observation. The flag type identifies the level of detail to which the flag applies: a specific item in a record, the record as a whole, or an observation about a repeating item's value in a record.

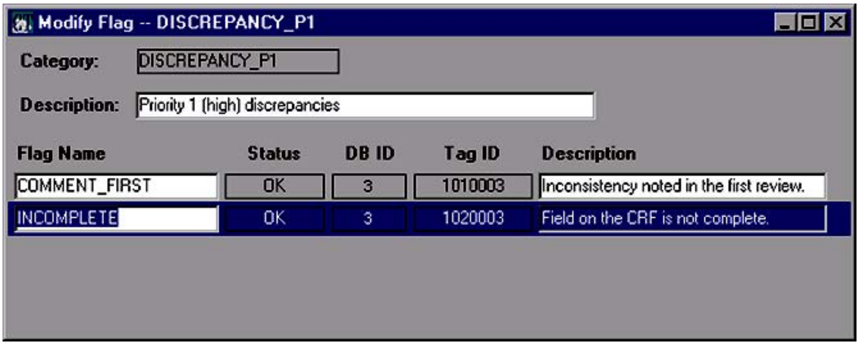
When FLAG_TO_DISCREPANCY encounters a discrepancy flag, the flag type can affect the discrepancy record that is created as follows:

- For item flags, FLAG_TO_DISCREPANCY also ensures that a VCT_ERRORITEM record is included automatically for the flagged item.
- For observation flags, only one discrepancy record is created. The first record of the observation (with a CT_RECID ending in .001) is used as the primary record.

Example

The following example shows how you use Design to create new discrepancy flag categories and write rules that call the FLAG_TO_DISCREPANCY procedure. After flags in the flag category are attached to data in Enter, and validation runs in Manage, discrepancy records can be generated automatically in Resolve.

The following example shows the discrepancy flag category DISCREPANCY_P1, as created in Design in the MEDIKA_CLINICAL protocol. There are two flag names in the flag:



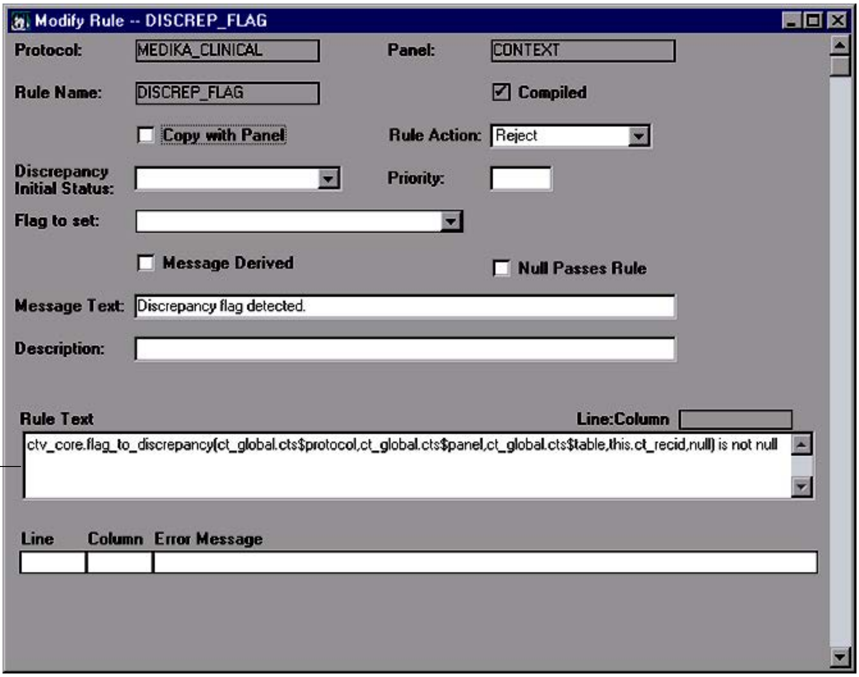
Modify Flag -- DISCREPANCY_P1

Category: DISCREPANCY_P1

Description: Priority 1 (high) discrepancies

Flag Name	Status	DB ID	Tag ID	Description
COMMENT_FIRST	OK	3	1010003	Inconsistency noted in the first review.
INCOMPLETE	OK	3	1020003	Field on the CRF is not complete.

The following example shows the DISCREP_FLAG rule created in Design in MEDIKA_CLINICAL to search for instances of flags in this category:



Modify Rule -- DISCREP_FLAG

Protocol: MEDIKA_CLINICAL

Panel: CONTEXT

Rule Name: DISCREP_FLAG

☑ Compiled

☐ Copy with Panel

Rule Action: Reject

Discrepancy Initial Status:

Priority:

Flag to set:

☐ Message Derived

☐ Null Passes Rule

Message Text: Discrepancy flag detected.

Description:

Rule Text

Line:Column

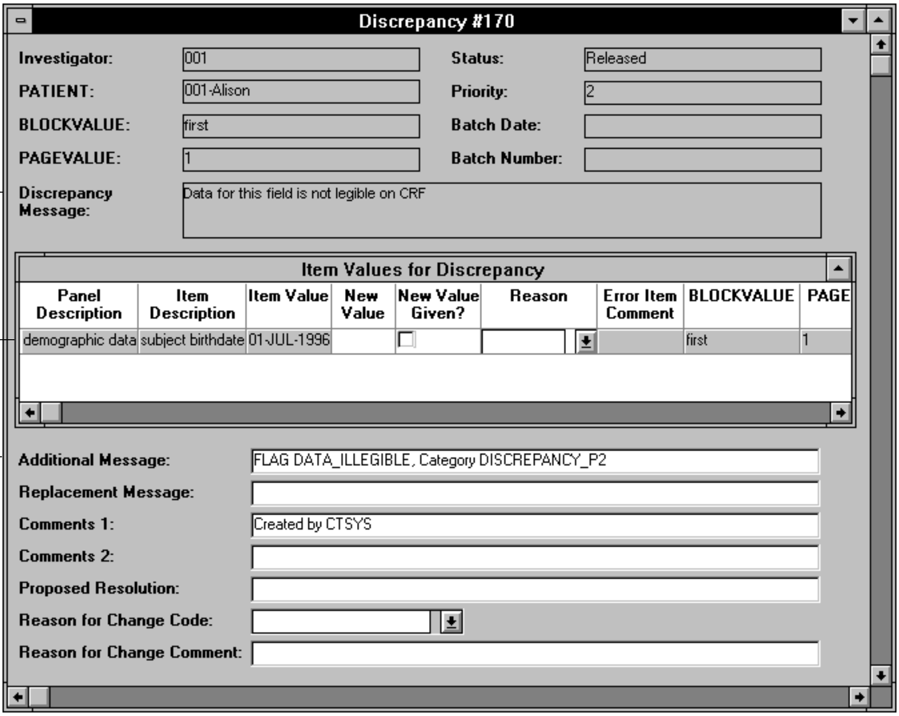
```
ctv_core.flag_to_discrepancy(ct_global.cts$protocol,ct_global.cts$panel,ct_global.cts$table,this.ct_recid,null) is not null
```

Line	Column	Error Message

Rule text

Note: When you create a rule, you create it in a specified panel. If discrepancy flags will be attached to records or items in more than one panel, a rule must be created in each of these panels for validation to generate discrepancy records. Alternatively, you can create the rule in the CONTEXT panel; it will run when any panel (except those of Type 0) is validated. For MEDIKA_CLINICAL, the rule is attached to the CONTEXT panel.

The following example shows the detailed Discrepancy # window of a discrepancy record generated by this process:



Investigator:	001	Status:	Released
PATIENT:	001-Alison	Priority:	2
BLOCKVALUE:	first	Batch Date:	
PAGEVALUE:	1	Batch Number:	

Discrepancy Message: Data for this field is not legible on CRF

Item Values for Discrepancy

Panel Description	Item Description	Item Value	New Value	New Value Given?	Reason	Error Item Comment	BLOCKVALUE	PAGE
demographic data	subject birthdate	01-JUL-1996		☐			first	1

Additional Message:	FLAG DATA_ILLEGIBLE, Category DISCREPANCY_P2
Replacement Message:	
Comments 1:	Created by CTSYS
Comments 2:	
Proposed Resolution:	
Reason for Change Code:	
Reason for Change Comment:	

Flag name description

Item flag created VCT_ERRORITEM record

Comments entered in flag

Advanced techniques for creating discrepancy records

This section explains how to create discrepancy records when the methods described previously do not adequately address your needs. These advanced techniques involve directly calling these routines in the proper sequence, as follows:

- CLEAR_ERRORITEMS clears memory.
- SETUP_ERRORITEM saves in memory information about discrepancy-related item values (See "Writing a derivation for Resolve" on page 271.)
- MANAGE_DISCREPANCY detects duplicate discrepancy records and creates new records using the information provided to it as arguments and by SETUP_ERRORITEM.
- ENTER_NEW_DISCREPANCY creates discrepancy records without checking for duplicates.

- INSERT_DISCREPANCY_*data_type*_ITEM inserts into a discrepancy record an error item of the specified type.
- FINISH_NEW_DISCREPANCY allows for discrepancies created by ENTER_NEW_DISCREPANCY to be audited and allows them to be replicated if Multisite is in use.

The following sections explain how to create discrepancy records using these routines.

How to check for duplicates and create records

If you want to check for a duplicate and, if none exists, create a new discrepancy record using MANAGE_DISCREPANCY. This routine is typically used within the validation environment (e.g., from a derivation or from PL/SQL code called by a derivation); it is called automatically after each test during validation.

Note: You should use MANAGE_DISCREPANCY only in special cases—for example, if an entire panel must be checked.

Before you call MANAGE_DISCREPANCY, consider doing one or both of the following tasks:

- To clear the memory set by previous calls to SETUP_ERRORITEM, call CLEAR_ERRORITEMS prior to validating a new record. This routine has no arguments.
- To give Resolve information about item values that you want saved with a discrepancy, call SETUP_ERRORITEM. MANAGE_DISCREPANCY and SETUP_ERRORITEM can link error item data with the appropriate discrepancy only if you ensure that the *i_rule_name* arguments provided to both routines have the same non-null value.

How to call MANAGE_DISCREPANCY

When you are ready to call MANAGE_DISCREPANCY, use the following syntax:

new_id := CTV_CORE.MANAGE_DISCREPANCY(*i_passfail, i_protocol, i_panel, i_src_recid, i_subject_id, i_block_item, i_page_item, i_orctable, i_errdt, i_errtype, i_remarks, i_erract, i_rec_moddate, i_rule_name, i_source, i_priority, i_dstatus, i_block_repeat_item, i_page_repeat_item*);

new_id is the value of the discrepancy ID if a new discrepancy record is created; otherwise the value is null.

i_passfail must be FALSE to indicate a rule failure (the kind of situation that leads to a discrepancy record being created). It must be TRUE to indicate that data no longer fails, in which case MANAGE_DISCREPANCY looks for and closes every pre-existing discrepancy record with the same panel, src_recid, and rule name.

i_protocol is the current protocol. If null, MANAGE_DISCREPANCY uses the protocol where validation is being run: ct_global.cts$protocol. (If it is called outside the validation environment, do not pass null.)

i_panel is the panel containing the source data. If null, the current validation panel (ct_global.cts$panel) is used.

i_src_recid is the ct_recid of the source record that contains the discrepancy. As all discrepancies must be associated with existing source data, this value must be provided. If null, the ct_recid of the record being processed (ct_global.cts$ct_recid) is used.

i_subject_id is the integer identified for the current patient, as assigned by the Clintrial software; it is not the value of the subject item. If null, the subject_id of the record being processed (ct_global.cts$subject_id) is used.

i_block_item is the value (as a VARCHAR2) of the block item for the record containing the discrepancy. If null, the block item value of the record being processed (ct_global.cts$block) is used.

i_page_item is the value (as a VARCHAR2) of the page item for the record containing the discrepancy. If null, the page item value of the record being processed (ct_global.cts$page) is used.

i_orctable must be either ‘UPDATE’ or ‘DATA’ indicating which table contains the source data. If null, the table currently being processed (ct_global.cts$table) is used.

i_errdt is the date you want stored as the discrepancy date. It must be an Oracle DATE and not a VARCHAR2. You could pass SYSDATE for this argument. If null, the current value of the error date (typically SYSDATE when validation started on the record) is used (ct_global.cts$err_date).

i_errtype must be either VALIDATE or MANUAL. (For MANUAL, you would be more likely to use ENTER_NEW_DISCREPANCY, which is described later in this section.) If you are running in the validation environment and leave this null, ct_global.cts$err_type is used, and the type is set to VALIDATE. Duplicate detection is not done for discrepancies of type MANUAL.

i_remarks is the discrepancy message. If null, the current rule's message (ct_global.cts$err_msg) is used. (You are unlikely to be calling MANAGE_DISCREPANCY as part of rule processing, so you should provide this argument.)

i_erract is what you want stored in the error action column. Typically, you specify 'REPORT' or 'REJECT.' If null, the current rule's error action (ct_global.cts$err_action) is used. If you are not within a rule, this value is null.

i_rec_moddate is the latest modification date of the source record. Typically, that is the record's MERGE_DATETIME. This must be passed as an Oracle DATE. If null, the modification date of the record being processed (ct_global.cts$rec_moddate) is used.

i_rule_name is the name of the rule you want stored. It is also used by MANAGE_DISCREPANCY to find data associated with this discrepancy that were stored in memory by SETUP_ERRORITEM. If null and you are calling this from within a routine called by a rule, the current rule's name (ct_global.cts$rule_name) is used.

i_source is stored as the source of the discrepancy. If null, 'RULE' is used. Resolve saves, but does not use, this value.

i_priority is the priority to assign the discrepancy. Null is acceptable.

i_dstatus is the code for the initial status of the discrepancy. If null, 'N' (for New) is used.

i_block_repeat_item is the value of the record's block repeat item, if any. If null, the value of the block repeat item for the record being validated (ct_global.cts$block_repeat_item) is used.

i_page_repeat_item is the value of the record's page repeat item, if any. If null, the value of the page repeat item for the record being validated (ct_global.cts$page_repeat_item) is used.

Note: MANAGE_DISCREPANCY inserts a record into the Clintrial software Error Log if i_passfail is FALSE, the error type is not MANUAL, and either the system parameter CTV_USE_ERRORLOG is True or the current protocol (provided as the *i_protocol* argument) is not set up for Resolve. If the protocol is set up for Resolve, this routine proceeds normally—that is, by autoclosing some discrepancy records and creating new records, as appropriate.

How to create records without checking for duplicates

To create a new discrepancy record without checking for duplicates, call ENTER_NEW_DISCREPANCY using the following syntax:

new_id := CTV_CORE.ENTER_NEW_DISCREPANCY(*i_protocol, i_panel, i_src_recid, i_subject_id, i_block_item, i_page_item, i_orctable, i_errdt_s, i_remarks, i_rec_moddate_s, i_source DEFAULT 'MANUAL', i_priority DEFAULT NULL, i_dstatus DEFAULT NULL*);

new_id is the returned value of the new discrepancy's unique ID. This value is used in calls to the INSERT_DISCREPANCY_*_ITEM routines. If an error occurs and the discrepancy is not created, *new_id* is null.

i_protocol is the name of the protocol containing the discrepancy.

i_panel is the name of the panel containing the discrepancy.

i_src_recid is the CT_RECID of the discrepancy record's source record.

i_subject_id is the Clintrial software identification number for the patient. (You can determine a subject's ID by looking in the SUBJECT_ID column of the DATA table of the enrollment panel.)

i_block_item is the value of the block item for the discrepancy's source record.

i_page_item is the value of the page item for the discrepancy's source record.

i_orctable indicates the type of table where the source data resides. It must be 'UPDATE' or 'DATA.'

i_errdt_s is the date of the discrepancy in a special string format. To convert a date into the special format, use the following syntax:

TO_CHAR(*errdt*, ct_global.param_date_fmt)

replacing *errdt* with your DATE value or variable. If you leave this argument null, the special string format version of SYSDATE is used.

i_remarks is the discrepancy message.

i_rec_moddate_s is the last modification date of the source record passed in the special string format (like *i_errdt_s*). Typically, a record's MERGE_DATETIME should be used. For example, if the date is in the DATE variable *chgdate*, you would pass this argument as:

TO_CHAR(chgdate, ct_global.param_date_fmt) *i_source* is the source of the discrepancy. If you leave it null, it is set to 'MANUAL.'

i_priority is the integer priority to assign the discrepancy. Null is acceptable.

i_dstatus is the intial discrepancy status (code). If you leave it null, the initial status is set to 'N' (New).

The error type of every record that you create using ENTER_NEW_DISCREPANCY is MANUAL. Once you create a new discrepancy record this way, use the routines summarized in the following sections to add error items to that record.

How to create discrepancies that can be audited and replicated in Multisite

To create a new discrepancy record that can be audited, and that can be replicated in a Multisite environment, you must call FINISH_NEW_DISCREPANCY after the discrepancy is created with ENTER_NEW_DISCREPANCY. Use the following calling sequence:

CT_RESOLVE4.FINISH_NEW_DISCREPANCY(*i_protocol, i_error_id*);

where:

i_protocol is the name of the protocol containing the discrepancy.

i_error_id is the ID of the discrepancy just created. It is returned by ENTER_NEW_DISCREPANCY.

How to control whether the results of a rule failure on EDC data go to the EDC discrepancy tables or to the Resolve discrepancy panels.

The results of a rule failure on EDC data by default go to the EDC discrepancy tables (INF_ERRORSTATUS and INF_ERRORITEM). With the creation and proper configuration of the **$edc_usect** variable, the results of such failures on EDC data may now be sent to the Resolve discrepancy panels VCT_ERRORSTATUS and VCT_ERRORITEM.

In order for this solution to be available, a user, in a derivation, must set a variable named **<rulename>$edc_usect** to TRUE. Then in protocols that are setup for Resolve, when a rule with that variable set TRUE fails on a record with first 4 characters of 9999 (which would normally go into the INF_ tables), it sends the results of any failures to the VCT_ERRORSTATUS and VCT_ERRORITEM panels.

When the discrepancy is sent to the EDC tables, the discrepancy can managed with EDC or Inform and that when the discrepancy is sent to the VCT panels, it can be managed via Resolve.

The default is FALSE.

For example, if you have a rule named XYZ and you want to always manage discrepancies caused by it with Resolve no matter where the original data comes from, you would create a derivation (any name) in the panel that contains rule XYZ and in that derivation you would put the line:

xyz$edc_usect := true;

How to insert error items of type VARCHAR2

To insert into a discrepancy record error items of type VARCHAR2, use the following syntax:

retval := CTV_CORE.INSERT_DISCREPANCY_STRING_ITEM(*i_protocol, i_error_id, i_panel, i_item_recid, i_item_name, i_item_value, i_item_comment*);

i_protocol is the name of the protocol containing the discrepancy.

i_error_id is the ID of the discrepancy with which to associate this error item. It is returned by ENTER_NEW_DISCREPANCY.

i_panel is the name of the panel containing the error item.

i_item_recid is the ct_recid of the record containing the error item.

i_item_name is the name of the item being provided as an error item.

i_item_value is the value of the item (VARCHAR2).

i_item_comment is any comment you want inserted into the error item record. Null is acceptable.

How to insert error items of type NUMBER

To insert into a discrepancy record error items of type NUMBER, use CTV_CORE.INSERT_DISCREPANCY_NUMBER_ITEM. This routine has the same arguments as INSERT_DISCREPANCY_STRING_ITEM, except that the *i_item_value* argument must be a NUMBER.

How to insert error items of type DATE

To insert into a discrepancy record error items of type DATE, use CTV_CORE.INSERT_DISCREPANCY_DATE_ITEM. Again, the arguments are the same as those of INSERT_DISCREPANCY_STRING_ITEM, except that the *i_item_value* argument must be an Oracle DATE.

22 *Resolve and Multisite*

General information

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

- The Resolve protocol
- Setting up protocols for use with Resolve
- Working with Discrepancy records

The Resolve protocol

You can distribute the Resolve protocol, CTRESOLVEREF, just as you distribute other protocols in Multisite.

Because the CTRESOLVEREF protocol must be available at every site where Resolve is used, you must either import CTRESOLVEREF at one site and then distribute it and replicate its data to the other sites, or import the protocol at every site. A combination of these techniques is also acceptable.

If you want all changes to CTRESOLVEREF to be centralized, then import CTRESOLVEREF at the Distribution Master site, make changes there, and distribute the protocol to Distribution Subordinate sites.

Otherwise, you can import CTRESOLVEREF at each site.

Setting up protocols for use with Resolve

If you want a distributed protocol to be set up for Resolve at each site, you must take the following steps:

1. At the Distribution Master site, set up the protocol for Resolve. From the **Setup** menu, select **Setup for Resolve**.

   Caution: You can set up a protocol for Resolve only if it has not been distributed, or if it is open for revision. If a protocol that does not contain the VCT_ERRORSTATUS and VCT_ERRORITEM panels was already distributed, you must open this protocol for revision at the Distribution Master site before you run Setup for Resolve.

2. Distribute the protocol.
3. At each Distribution Subordinate site:

   a. Accept the protocol.

b. Install database tables.

c. Set up the protocol for Resolve.

Working with discrepancy records

When using Resolve for a protocol that is in replication, you can only modify discrepancy records for subjects owned at the current site. Records owned by other sites are accessible only in read-only mode.

Enter, Resolve, and Retrieve

Retrieve

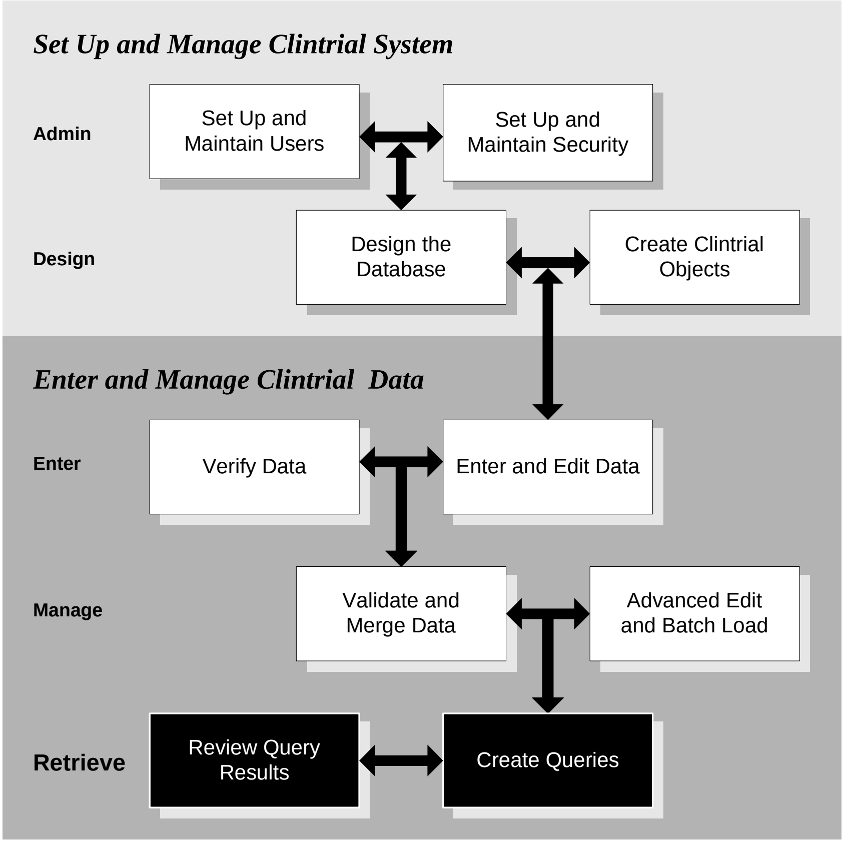
23 Introduction to Retrieve

Overview

How Retrieve fits in the Clintrial software workflow

The Clintrial software Retrieve module is the core module that you use to access and extract data from the Clintrial software database. You retrieve the data by creating queries, then running the queries to extract the data.

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



What are the Retrieve tasks?

Using Retrieve, you can:

- Create queries
- Save queries
- Work in the Query Library
- Save and display query results
- Print queries
- Create reports

Who are Retrieve users?

Data managers and medical reviewers are the main users of Retrieve, including anyone else who needs to extract clinical data from the database and work with query results.

Clintrial software database structure

This section describes the various objects that make up the Clintrial software, and how clinical data in a Clintrial software protocol is stored in Oracle database tables.

What is clinical data?

Clinical data is data that is collected during a clinical study, such as data collected about a subject on a case report form (CRF). For example, clinical data may include demographic data, previous medication data, or laboratory test results. In the Clintrial software, clinical data is stored in *clinical data protocols*.

What is a protocol?

A Clintrial software *protocol* is a logical container that organizes both 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.

A Clintrial software protocol and its protocol account have the same name. When you create a query, you specify the protocol that contains the data you want to retrieve.

Note: You can also create a query to retrieve data from multiple protocols (a *cross-protocol query*) using Query By SQL.

What are database tables, panels, and items?

Oracle *database tables* contain the clinical data in columns and rows:

- Each column in the table corresponds to an *item*.
- Each row in the table corresponds to a *record*, or group of related columns.

For example, the following table containing demographic data has columns corresponding to the items SUBJECT ID, INITIALS, DOB, SEX, HEIGHT, and WEIGHT, and rows corresponding to subject records:

Column/Item

Row/Record

SUBJECT ID	INITIALS	DOB	SEX	HEIGHT	WEIGHT
K004298	KC	09-03-53	F	65	140
K008743	LM	04-20-55	F	68	153
K000976	PR	06-14-61	M	72	185
K006647	ST	10-04-52	F	62	125
K003992	EJ	12-12-48	M	67	143

In the Clintrial software, the Oracle database tables are defined by panels. A *panel* is a collection of logically or clinically related items. An *item* stores the data collected by a single field in a study page or by a single field in a batch-loaded file. For example, the PHY_EXAM panel might define items for storing physical examination results.

Each panel in a protocol defines three Oracle database tables for the storage of clinical data. The items in the panel correspond to the columns of the following three database tables:

- Update

  The *update table* stores clinical data when it is first entered in the Clintrial software, and is a holding area for clinical data while it is being cleaned. The name of the update table is *panel-name*_UPDATE (for example, PHY_EXAM_UPDATE).

- Data

  The *data table* stores clinical data that has passed validation and has been merged. This data is considered to be *clean data*. The name of the data table is *panel-name*_DATA (for example, PHY_EXAM_DATA).

- Audit

  The *audit table* stores copies of original clinical data as they existed before modification or deletion. This table is used to keep a record of the clinical data modifications and deletions from a point in time determined by your data manager. The name of the audit table is *panel-name*_AUDIT (for example, PHY_EXAM_AUDIT).

Note: The value of the user parameter RTV_DEF_SOURCE determines the default type of database table the Clintrial software uses as source data for a Query By Form, Query By Panel, or Ad Hoc Query.

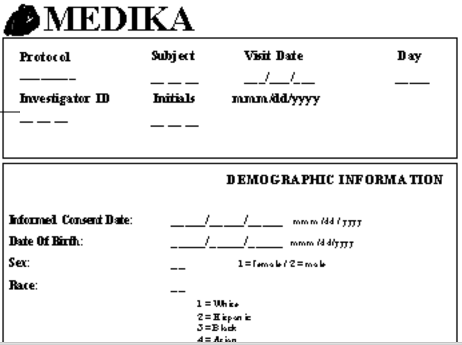
For information on validating and merging data, see the *Manage* section in *Manage, Classify, and Lab Loader.*

For information on Retrieve user parameters, see "What are user preferences?" on page 315.

Example

The following example shows the relationship among panels, items, and database tables:

CRF page contains data to be entered in Enter.



MEDIKA

Protocol ______ Subject ___ Visit Date __/__/__ Day ___

Investigator ID ___ Initials ___ mmm/dd/yyyy

DEMOGRAPHIC INFORMATION

Informed Consent Date: ____/____/____ mmm /dd / yyyy

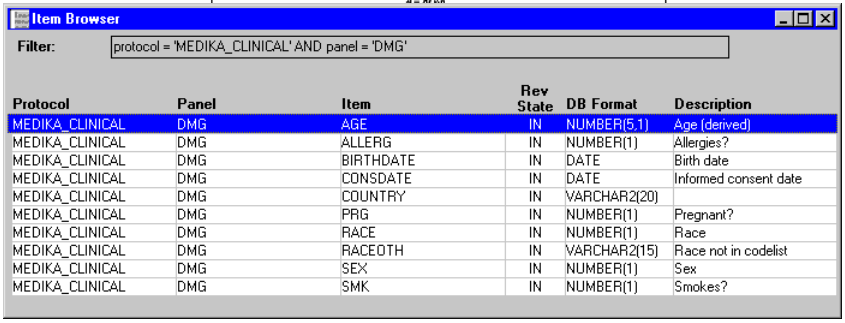
Date Of Birth: ____/____/____ mmm /dd/yyyy

Sex: __ 1=female / 2=male

Race: __

1 = White
2 = Hispanic
3 = Black
4 = Asian

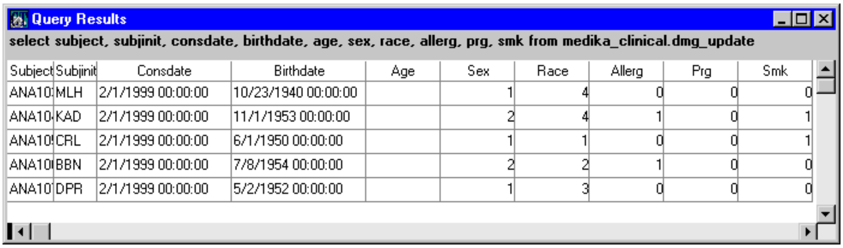
DMG panel and items that define clinical data tables.



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?

Clinical data table DMG_UPDATE stores entered data.



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

Clintrial software data-entry structure

Overview

In a clinical study, clinical data is often collected using a case report form (CRF). A *case report form* (CRF) is the paper form on which clinical data for all visits of a particular subject during a particular study is collected. Data-entry operators then interactively enter the clinical data from the CRF into a database using online study books, which are composed of study pages. You can later use these same study pages to create queries to retrieve the data using Query By Form.

For more information, see Chapter 25.

What are study books, blocks, and study pages?

A *study book* is an ordered list of related study pages that corresponds to a CRF and provides the data-entry interface to the underlying Oracle database tables.

Study books are organized by *blocks*, which group related study pages. For example, a block named VISIT1 might include the following study pages: VITAL SIGNS, PHYSICAL EXAM, and DRUG ADMIN.

A *study page* (or *page*) is a data-entry window that represents a single page in a CRF. A study page consists of a context section (for context items), and one or more additional page sections (for clinical data).

What are context sections and page sections?

Although you do not need to understand context sections and page sections to create a Query By Form, you may want to know how the various fields on a study page are stored in the Oracle database.

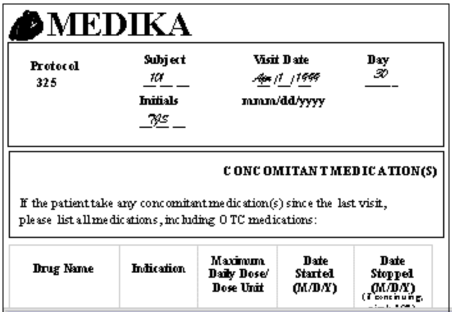
A *context page section* is a section of a study page that contains context items. *Context items* are items that uniquely identify records in the database tables. A context section appears on each study page.

A *page section* is a section of a study page that collects a particular type of clinical data, which is then stored in a corresponding panel. For example, a single study page might include a section for demographic data (stored in the DMG panel), a section for vital signs (stored in the VITALS panel), and a section for physical examination results (stored in the PHY_EXAM panel).

Example

The following example shows the Clintrial software data-entry structure:

Clinical data is typically entered in a paper CRF.



MEDIKA

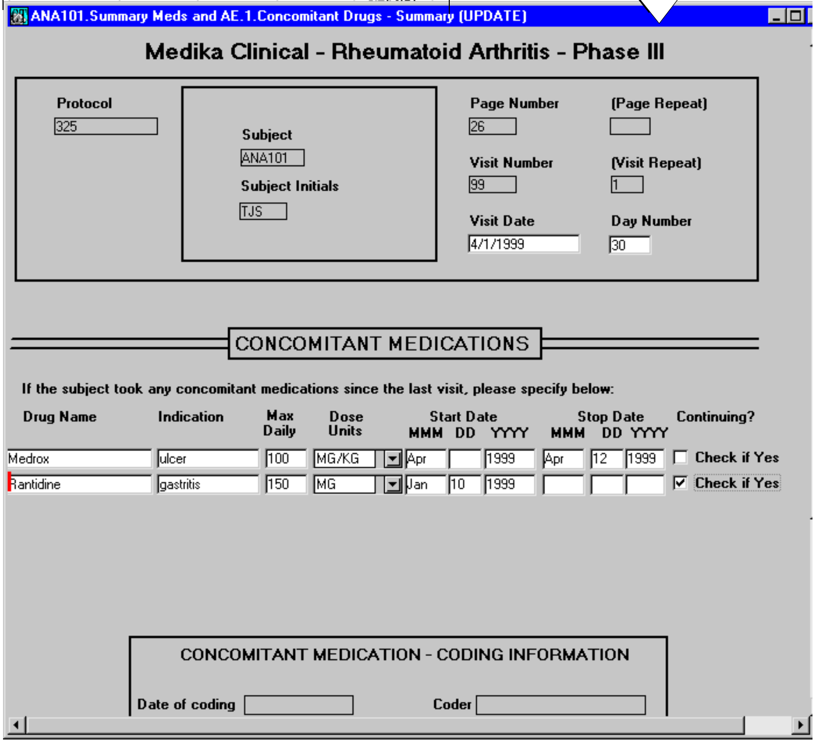
Protocol 325	Subject 101 Initials TJS	Visit Date Apr /1 /1999 mmm/dd/yyyy	Day 30

CONCOMITANT MEDICATION(S)

If the patient take any concomitant medication(s) since the last visit, please list all medications, including OTC medications:

Drug Name	Indication	Maximum Daily Dose/ Dose Unit	Date Started (M/D/Y)	Date Stopped (M/D/Y) (if continuing,

Clinical data values are entered interactively in the fields of online study pages.



ANA101.Summary Meds and AE.1.Concomitant Drugs - Summary (UPDATE)

Medika Clinical - Rheumatoid Arthritis - Phase III

Protocol: 325

Subject: ANA101

Subject Initials: TJS

Page Number: 26

(Page Repeat):

Visit Number: 99

(Visit Repeat): 1

Visit Date: 4/1/1999

Day Number: 30

CONCOMITANT MEDICATIONS

If the subject took any concomitant medications since the last visit, please specify below:

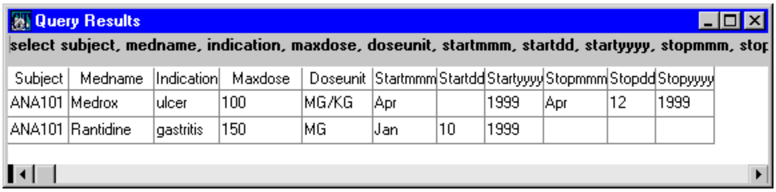
Drug Name	Indication	Max Daily	Dose Units	Start Date MMM	DD	YYYY	Stop Date MMM	DD	YYYY	Continuing?
Medrox	ulcer	100	MG/KG	Apr		1999	Apr	12	1999	□ Check if Yes
Rantidine	gastritis	150	MG	Jan	10	1999				☑ Check if Yes

CONCOMITANT MEDICATION - CODING INFORMATION

Date of coding:

Coder:

The Clintrial software stores the clinical data values as records in database tables.



Query Results

select subject, medname, indication, maxdose, doseunit, startmmm, startdd, startyyyy, stopmmm, stop

Subject	Medname	Indication	Maxdose	Doseunit	Startmmm	Startdd	Startyyyy	Stopmmm	Stopdd	Stopyyyy
ANA101	Medrox	ulcer	100	MG/KG	Apr		1999	Apr	12	1999
ANA101	Rantidine	gastritis	150	MG	Jan	10	1999			

Using Retrieve to query the database

What is a query?

A *query* is a statement composed of conditions that identifies the data you want to retrieve from the database. In Retrieve, you can create queries using one of four query tools:

- Query By Form (QBF) — Chapter 25
- Query By SQL — Chapter 26
- Query By Panel (QBP) — Chapter 27
- Ad Hoc Query — Chapter 28

For example, you can create a query to retrieve records that answer the following question:

Which adverse experiences have been reported by Investigator 31?

What are query results?

Query results are records that meet the conditions specified by a query. Once you have created a query that identifies the data you want to retrieve, you can run the query and save the query results in a variety of formats. Then, you can use the query results in other applications to perform statistical analysis, create reports, or perform other necessary tasks.

For example, you can send the query results to a SAS Data file, then open the results in SAS to perform statistical analysis.

For more information on query results, see Chapter 30.

Creating queries

You can create queries in Retrieve using one of four query tools:

- Query By Form

  If you are familiar with your company’s study pages, you can use them to create a Query By Form by selecting the fields you want to retrieve and entering criteria directly into fields on a study page.

- Query By SQL

  If you are familiar with the structure of the database and know SQL, you can use Query By SQL to create SELECT and DESCRIBE statements.

- Query By Panel

  If you are familiar with panels and items within the Clintrial software as well as with the syntax of SQL, you can use Query By Panel (QBP) to create queries. QBP offers an interface that is easy to use and can help you build an SQL statement with simple and outer joins, WHERE clauses, functions, and operators.

- Ad Hoc Query

  If you are not familiar with SQL or with the data-entry study pages, you can use Ad Hoc Query to create queries. Ad Hoc Query is a graphical query tool that allows you to create queries by selecting panels, items, functions, and operators.

Saving queries

Once you have created a query, you can save the query to a Query Library, where you or other users with the appropriate access rights can access it in the future.

For information on working with saved queries, see Chapter 24.

Opening the Query Library

The Query Library contains a list of all saved queries to which you have access. If you have the appropriate access rights, you can copy queries between protocols, delete queries, modify saved queries, and run saved queries.

For information on working with the Query Library, see Chapter 24.

Running queries

You can run saved queries, or queries you have just created or modified, and send the query results to a window on your monitor, to an Oracle database table, or to a SAS Data file.

For information on running queries, see Chapter 30.

Saving query results

Once you have created and run a query, you can save the query results in a variety of formats. For example, you can save the data to an Oracle database table, to a Microsoft Excel spreadsheet, or to a SAS Data file.

For information on saving query results, see Chapter 30.

Displaying query results

You can display the query results on your monitor so you can review the results before saving them. For example, you can resize columns, and filter and sort the records. If you are reviewing results for an Ad Hoc Query, you can display data from the panels for the query results records.

For information on reviewing query results, see Chapter 30.

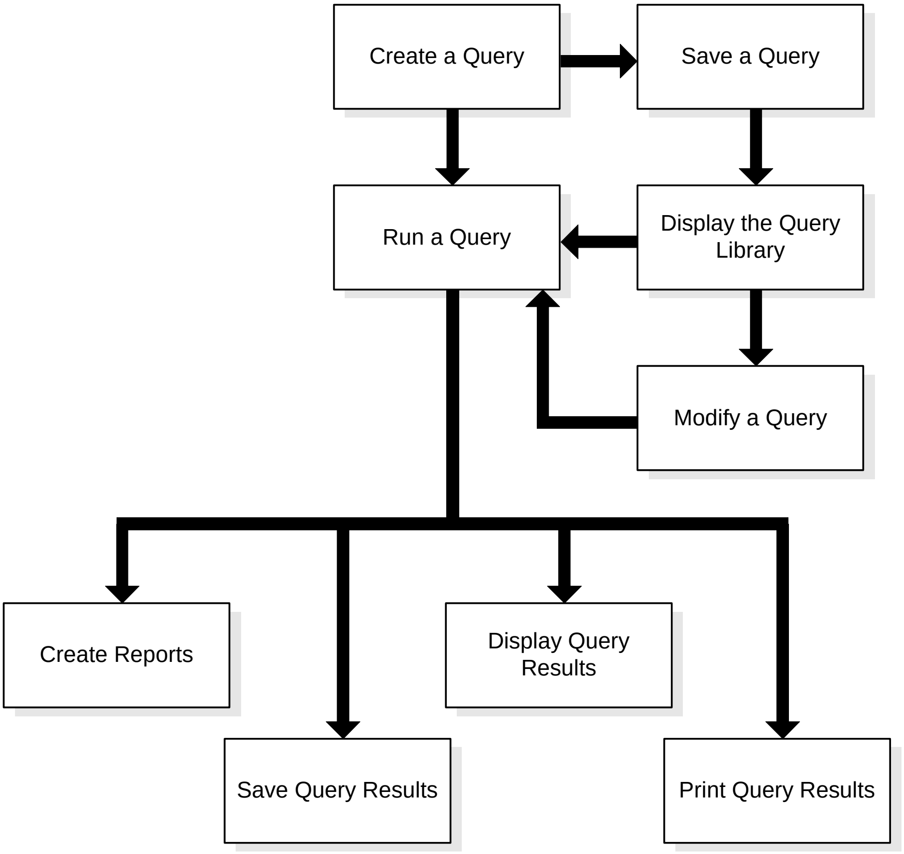
Printing query results

You can print the query results after you display them in a window on your monitor.

For information on printing query results, see Chapter 30.

Sample workflow in Retrieve

The following figure shows a typical workflow within Retrieve:



Required access rights and access levels

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 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 or Clintrace 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.

Setting your user preferences

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 Retrieve:

User preference:	**Description:**
RTV_DEF_DECODE Default: 1	Determines what value is retrieved for a coded item when you run a Query By Form or Ad Hoc Query, or when you use the CT_DECODE procedure in a Query By SQL. Possible values: 0 – Code, 1 – Value, 2 – Short label, 3 – Long label.
RTV_DEF_SOURCE Default: DATA	Sets the default type of database table the Clintrial software uses as source data for a Query By Form, Query By Panel, or Ad Hoc Query. Possible values: DATA – Retrieves data from data table. UPDATE – Retrieves data from update table. ALL – Retrieves data from both data and update tables.
RTV_INCL_SYS Default: No	Determines whether the Clintrial software retrieves system items when you run a Query By Form or a Query By Panel. Possible values: Yes – In QBF, automatically retrieves system items. In Query By Panel (QBP), system items are available for selection. No – In QBF, does not retrieve system items. In QBP, system items are not available for selection.
RTV_SAS_RECLEN Default: 80 for databases using BYTE semantics and 400 for database using CHAR semantics.	Determines the record length for records generated when you save query results to a SAS Data file. Possible values: Any numeric value between 80 and 1024. If you choose the ANSI option in the **Encoding** field in the window in "Saving results to a SAS Data file" on page 413, the length for items with the VARCHAR2 datatype will be double the setting in this field. For the UTF-8 option, the length will be five times the setting in this field.

User preference:	Description:
SAS_LIBNAME Default: None	Specifies the default directory and file number in which to store the SAS Format Library you create.

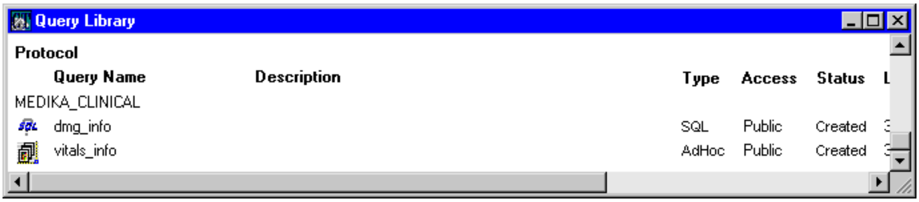
24 Using the Query Library

Overview

What is the Query Library?

Retrieve allows you to create and run queries, then save these queries in a *Query Library* that is stored in the Oracle database with your clinical data. Once the queries are saved in the Query Library, users with the appropriate access rights can copy queries between protocols, delete queries, modify saved queries, and run saved queries.

The Query Library displays a list of all the queries to which you have access:



Query Library

Protocol	Query Name	Description	Type	Access	Status	L
MEDIKA_CLINICAL						
	dmg_info		SQL	Public	Created	3
	vitals_info		AdHoc	Public	Created	3

To work with a specific query, click that query. Retrieve then highlights the selected query.

Note: You can select only one query at a time.

Information in the Query Library

The Query Library displays the following information about the queries to which you have access:

Column:	**Description:**
Protocol	The name of the protocol for which the query retrieves data.
Query Name	The name of the query.
Description	A description of the query.
Type	The method used to create the query. The query type can be SQL, QBF, QBP, or Ad Hoc.

Column:	Description:
Access	The security access type of the query. The query can be either a public query or a private query.
Status	The status of a query. When a query is first saved, its status is Created. Any subsequent modifications to the query by any user change the query status to Modified.
Last Modified	The date and time the query was created or last modified by any user.
By user	The user ID of the user who created or last modified the query.
Last run	The date and time any user last ran the query.

Types of security access

When you save a query, you can assign it a security access type. The security access type works with the Query Library access right to control which users have access to the query.

You can assign a query one of the following security access types:

- Public query

  If you save a query as a public query, any user with the appropriate privileges can display, edit, or run the query. The query is displayed in the Query Library of all users who have access to the corresponding protocol.

  Only users with Publish access to the Query Library can save queries as public queries.

- Private query

  If you save a query as a private query, only you (that is, your user account) can revise or rerun the query. The query is displayed only in your Query Library. You can make the query available to other users by resaving the query and changing the security level to public.

  Only users with Publish or Write access to the Query Library can save queries as private queries.

The following table lists the Query Library access levels (set in Admin), and identifies which tasks in Retrieve these access levels permit:

Access Level:	You can do the following:
Publish	• Create and save public and private queries. • Open and modify public and private queries. You can save changes to the queries as either public or private queries. • Copy public and private queries. The copied queries are saved as private queries. • Delete public and private queries. • Run public and private queries.
Write	• Create and save private queries. • Open and modify public and private queries, but you can save changes to a query only as a private query. • Copy public and private queries. The copied queries are saved as private queries. • Delete private queries. • Run public and private queries.
Read	• Create and run queries, but you cannot save them. • Open and modify public queries, but you cannot save the changes. • Run public queries.
None	Create and run queries but you cannot save them.

For more information on security, see *Getting Started*.

Displaying the Query Library

This section describes opening and closing the Query Library, and customizing the display of queries in the Query Library. For detailed instructions on performing these tasks, see the Help.

Opening the Query Library

To open the Query Library, do one of the following:

- From the **File** menu, select **Query Library**.
- On the toolbar, click **Query Library**.

Closing the Query Library

You can close the Query Library at any time by doing one of the following:

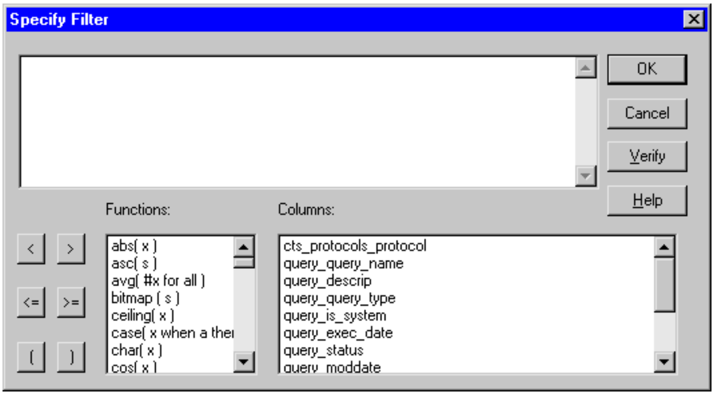
- From the **File** menu, select **Close Library**.
- On the toolbar, click **Close**.

Note: Closing the Query Library does not end your Retrieve session.

Filtering the Query Library

You can control which queries are displayed in the Query Library by using the **Filter** command. This command allows you to display a specific subset of queries to help you locate the query you want.

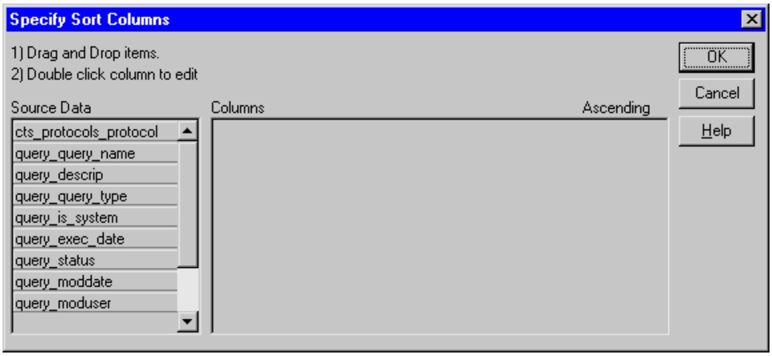
To specify a filter, from the **View** menu, select **Filter**. The Specify Filter dialog box opens:



Sorting the Query Library

You can control the order in which the queries are displayed in the Query Library by using the **Sort** command. This command allows you to specify criteria to control the sort order of the query list.

To specify a sort order, from the **View** menu, select **Sort**. The Specify Sort Columns dialog box opens:



Query Library tasks

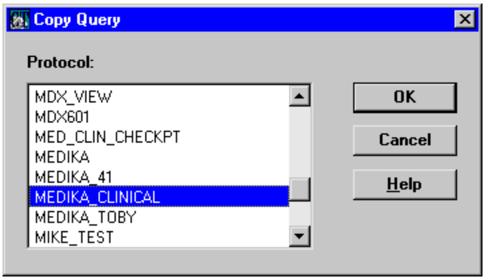
This section describes the tasks you can perform from the Query Library. For step-by-step instructions on performing these tasks, see the Retrieve Help.

Copying a query

You can copy a query from the current protocol to another protocol to which you have access. You can copy only one query at a time.

Note: When you copy a query to another protocol, Retrieve replaces any references to the original protocol with the name of the new protocol. If you copy a query to a protocol that does not have the same database table structure as the first protocol, you may need to edit the query.

To copy a query, select it from the Query Library. Then from the **File** menu, select **Copy Query**. The Copy Query dialog box opens:



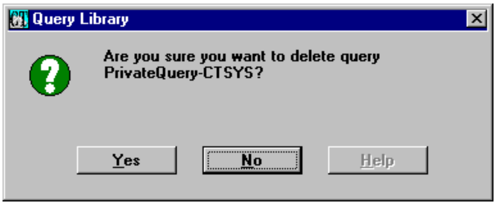
If you have Publish or Write access to the Query Library, you can copy both public and private queries. The copied queries are saved as private queries.

Note: To copy a query within a protocol, you must first open the query, then save it under a different name. To save the query, from the **File** menu, select **Save As**. If you have Publish access, you can also use the **Save As** command to create a public copy of a query.

Deleting a query

You can delete those public and private queries for which you have the appropriate access rights from the Query Library. You can delete only one query at a time.

To delete a query, first select the query to delete. Then from the **File** menu, select **Delete Query**. The following confirmation message displays:



Are you sure you want to delete query PrivateQuery-CTSYS?

If you have Publish access to the Query Library, you can delete both public and private queries.

If you have Write access to the Query Library, you can delete only your own private queries.

Opening a query

If a query exists in the database, and you have the appropriate access rights, you can open it to review or revise its contents.

To open a query, first select it from the Query Library. Then from the **File** menu, select **Open Query**.

- If you open a Query By Form, a Query By Panel, or a Query By SQL, an SQL window opens and displays the query in SQL format.

  Note: Once you save and close a Query By Form or a Query By Panel query, the query can be displayed in SQL format only.

- If you open an Ad Hoc Query, a Selection Criteria window opens and displays the query selection criteria.

For more information on working with each of the query types, see Chapter 25 through Chapter 28.

Running a query

You can run existing queries directly from the Query Library without opening the query. Only those queries to which you have appropriate access rights appear in the Query Library and are available for you to run.

To run the query, from the **Results** menu, select the results destination. Then from the **File** menu, select **Run Query**.

For information on results destinations and running queries, see Chapter 30.

Queries for use with CT_MEDDRA

The Clintrial software contains queries in the Retrieve Query Library that you can use to query the panels in the CT_MEDDRA thesaurus protocol for terms that are associated with data in your study.

The use of the MedDRA dictionary hierarchy makes it possible to use the CT_MEDDRA queries to derive all associated higher levels of dictionary terms from the dictionary, without any additional external information, if a dictionary term is known at any hierarchical level.

For more information on MedDRA and the MedDRA dictionary, see the *Manage* section in *Manage, Classify, and Lab Loader.*

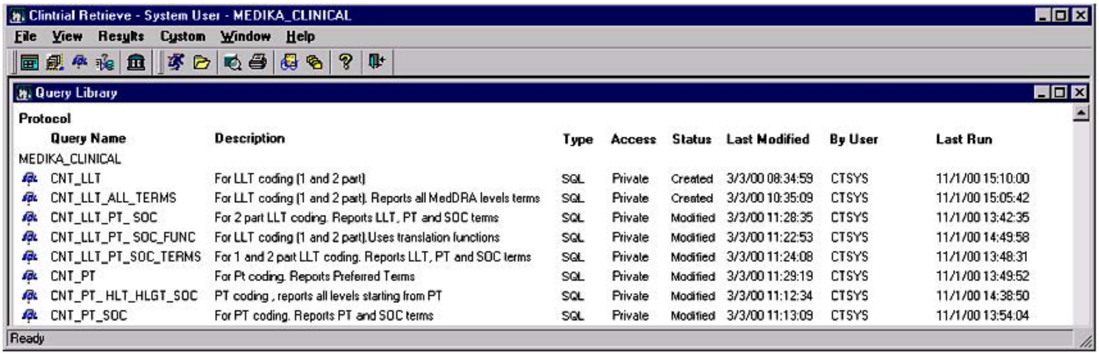
Using predefined queries with CT_MEDDRA

There are eight queries for use with CT_MEDDRA, five queries that can be used when coding against Lowest Level Terms (LLT) using the ADV_LLT panel, and three queries that can be used for coding against Preferred Terms (PT) using the ADV_PT panel.

For example, in the MEDIKA_CLINICAL protocol, the ADV_PT panel is set up for one-part coding, where only PT_CODE is captured and stored in the MEDIKA_CLINICAL panel and all queries associated with it can be used with either one or two-part coding. The ADV_LLT panel is set up for two-part coding, where both PT and LLT codes are captured and stored in the MEDIKA_CLINICAL panel. Some queries based on the ADV_LLT panel specifically require two-part coding, and others can be used with either one or two part coding.

You can use queries supplied for CT_MEDDRA in the Query Library to capture High Level Terms (HLT), High Level Group Terms (HLGT), and System Organ classes (SOC), associated with Preferred Terms (PT) or Lowest Level Terms (LLT) code found by the encoding procedure.

The following example shows the queries in the Query Library that you can use to. You select the query that you need, depending on the type of information that you want reported.



Protocol Query Name	Description	Type	Access	Status	Last Modified	By User	Last Run
MEDIKA_CLINICAL							
CNT_LLT	For LLT coding (1 and 2 part)	SQL	Private	Created	3/3/00 08:34:59	CTSYS	11/1/00 15:10:00
CNT_LLT_ALL_TERMS	For LLT coding (1 and 2 part). Reports all MedDRA levels terms	SQL	Private	Created	3/3/00 10:35:09	CTSYS	11/1/00 15:05:42
CNT_LLT_PT_ SOC	For 2 part LLT coding. Reports LLT, PT and SOC terms	SQL	Private	Modified	3/3/00 11:28:35	CTSYS	11/1/00 13:42:35
CNT_LLT_PT_ SOC_FUNC	For LLT coding (1 and 2 part).Uses translation functions	SQL	Private	Modified	3/3/00 11:22:53	CTSYS	11/1/00 14:49:58
CNT_LLT_PT_SOC_TERMS	For 1 and 2 part LLT coding. Reports LLT, PT and SOC terms	SQL	Private	Modified	3/3/00 11:24:08	CTSYS	11/1/00 13:48:31
CNT_PT	For Pt coding. Reports Preferred Terms	SQL	Private	Modified	3/3/00 11:29:19	CTSYS	11/1/00 13:49:52
CNT_PT_ HLT_HLGT_SOC	PT coding , reports all levels starting from PT	SQL	Private	Modified	3/3/00 11:12:34	CTSYS	11/1/00 14:38:50
CNT_PT_SOC	For PT coding. Reports PT and SOC terms	SQL	Private	Modified	3/3/00 11:13:09	CTSYS	11/1/00 13:54:04

Run Query icon

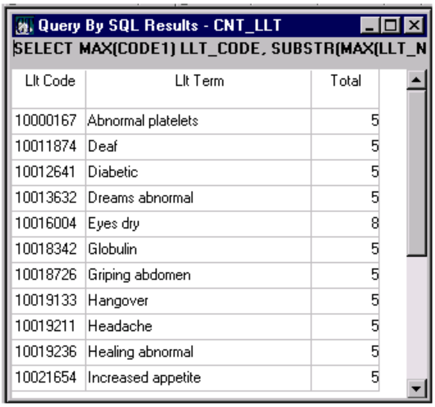
To select a specific query in the list, click the query, then from the **File** menu, select **Run Query**, or on the toolbar, click the Run Query icon.

Retrieve returns all corresponding data that is contained in the database at the specific levels of the MedDRA panels that you requested.

The CNT_LLT query

Use the CNT_LLT query if the coding was done against the LLTs, with either one or two-part coding. For example, a matching term is found in the L_LOW_LEVEL_TERM_DATA panel in CT_MEDDRA for every Lowest Level code in the ADV_LLT panel of MEDIKA_CLINICAL. All LLTs used during coding, and the number of times each LLT term was used, will be reported.

The following example shows the results of a CNT_LLT query:



```
SELECT MAX(CODE1) LLT_CODE, SUBSTR(MAX(LLT_N
```

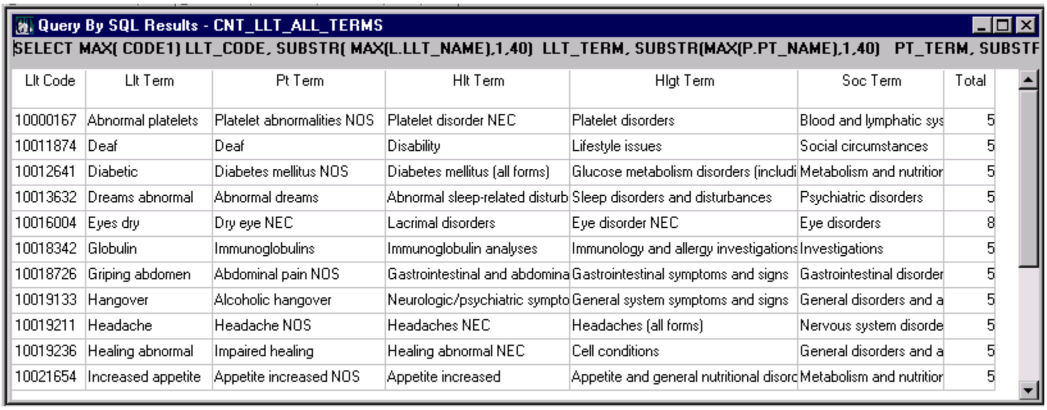
Llt Code	Llt Term	Total
10000167	Abnormal platelets	5
10011874	Deaf	5
10012641	Diabetic	5
10013632	Dreams abnormal	5
10016004	Eyes dry	8
10018342	Globulin	5
10018726	Griping abdomen	5
10019133	Hangover	5
10019211	Headache	5
10019236	Healing abnormal	5
10021654	Increased appetite	5

The CNT_LLT_ALL_TERMS query

Use the CNT_LLT_ALL_TERMS query if the coding was done against the LLTs, with either one or two-part coding. For example, matching terms found in the L_LOW_LEVEL_TERM_DATA panel, the L_ PREF_TERM_DATA panel, and the L_MD_HIERARCHY_DATA panel in CT_MEDDRA for every Lowest Level code in the ADV_LLT panel of MEDIKA_CLINICAL will be returned. The following information will be reported:

- All LLTs used during coding
- The number of times each LLT was used
- The PT associated with every LLT
- The HLT associated with every PT
- The HLGT associated with every PT
- The Preferred SOC term associated with every PT

The following example shows the results of a CNT_LLT_ALL_TERMS query:



```
SELECT MAX( CODE1) LLT_CODE, SUBSTR( MAX(L.LLT_NAME),1,40) LLT_TERM, SUBSTR(MAX(P.PT_NAME),1,40) PT_TERM, SUBSTF
```

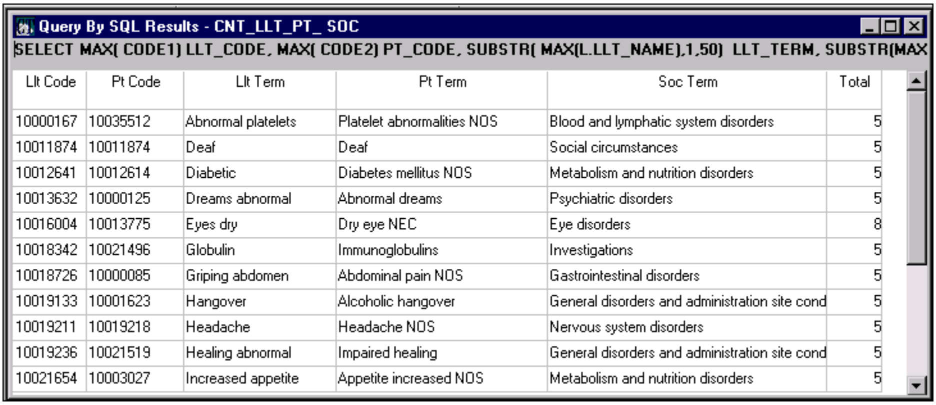
Llt Code	Llt Term	Pt Term	Hlt Term	Hlgt Term	Soc Term	Total
10000167	Abnormal platelets	Platelet abnormalities NOS	Platelet disorder NEC	Platelet disorders	Blood and lymphatic sys	5
10011874	Deaf	Deaf	Disability	Lifestyle issues	Social circumstances	5
10012641	Diabetic	Diabetes mellitus NOS	Diabetes mellitus (all forms)	Glucose metabolism disorders (includi	Metabolism and nutritior	5
10013632	Dreams abnormal	Abnormal dreams	Abnormal sleep-related disturb	Sleep disorders and disturbances	Psychiatric disorders	5
10016004	Eyes dry	Dry eye NEC	Lacrimal disorders	Eye disorder NEC	Eye disorders	8
10018342	Globulin	Immunoglobulins	Immunoglobulin analyses	Immunology and allergy investigations	Investigations	5
10018726	Griping abdomen	Abdominal pain NOS	Gastrointestinal and abdomina	Gastrointestinal symptoms and signs	Gastrointestinal disorder	5
10019133	Hangover	Alcoholic hangover	Neurologic/psychiatric sympto	General system symptoms and signs	General disorders and a	5
10019211	Headache	Headache NOS	Headaches NEC	Headaches (all forms)	Nervous system disorde	5
10019236	Healing abnormal	Impaired healing	Healing abnormal NEC	Cell conditions	General disorders and a	5
10021654	Increased appetite	Appetite increased NOS	Appetite increased	Appetite and general nutritional disorc	Metabolism and nutritior	5

The CNT_LLT_PT_SOC query

Use the CNT_LLT_PT_SOC query if the coding was done against the LLTs, with two-part coding. For example, it uses the PT_CODE stored in the panel in MEDIKA_CLINICAL instead of querying the PREF_TERMS panel of CT_MEDDRA. Matching terms found in the L_LOW_LEVEL_TERM_DATA and the L_MD_HIERARCHY_DATA panels in CT_MEDDRA for every Lowest Level code in the ADV_LLT panel of MEDIKA_CLINICAL will be returned. The following information will be reported:

- All LLTs used during coding
- The number of times each LLT was used
- The PT associated with every LLT
- The Preferred SOC term associated with every PT
- The HLT associated with every PT
- The HLGT associated with every PT

The following example shows the results of a CNT_LLT_PT_SOC query:



```
SELECT MAX( CODE1) LLT_CODE, MAX( CODE2) PT_CODE, SUBSTR( MAX(L.LLT_NAME),1,50)  LLT_TERM, SUBSTR(MAX
```

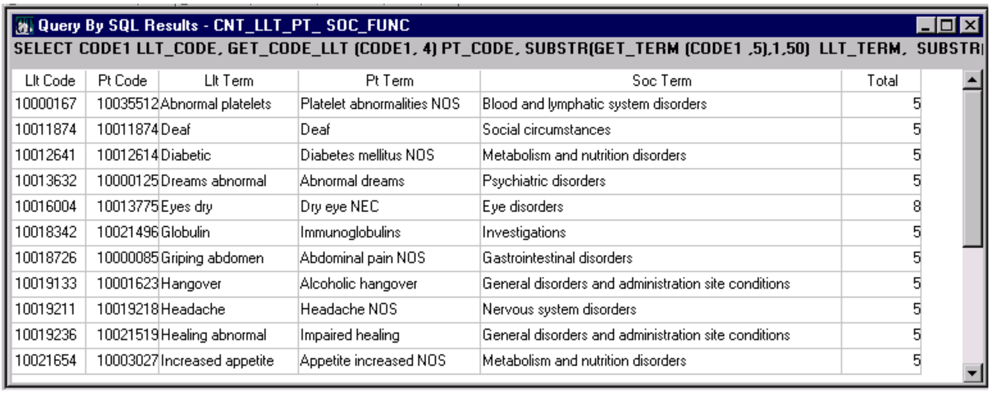
Llt Code	Pt Code	Llt Term	Pt Term	Soc Term	Total
10000167	10035512	Abnormal platelets	Platelet abnormalities NOS	Blood and lymphatic system disorders	5
10011874	10011874	Deaf	Deaf	Social circumstances	5
10012641	10012614	Diabetic	Diabetes mellitus NOS	Metabolism and nutrition disorders	5
10013632	10000125	Dreams abnormal	Abnormal dreams	Psychiatric disorders	5
10016004	10013775	Eyes dry	Dry eye NEC	Eye disorders	8
10018342	10021496	Globulin	Immunoglobulins	Investigations	5
10018726	10000085	Griping abdomen	Abdominal pain NOS	Gastrointestinal disorders	5
10019133	10001623	Hangover	Alcoholic hangover	General disorders and administration site cond	5
10019211	10019218	Headache	Headache NOS	Nervous system disorders	5
10019236	10021519	Healing abnormal	Impaired healing	General disorders and administration site cond	5
10021654	10003027	Increased appetite	Appetite increased NOS	Metabolism and nutrition disorders	5

The CNT_LLT_PT_SOC_FUNC query

The CNT_LLT_PT_SOC_FUNC query can be used if the coding was done against the LLTS, with either one or two-part coding. It uses translation functions provided with the CT_MEDDRA protocol to get the PT and SOC codes, rather than querying the CT_MEDDRA panels. The following information will be reported:

- All LLTs used during coding
- The number of times each LLT was used
- The PT associated with every LLT
- The Preferred SOC term associated with every PT

The following example shows the results of a CNT_LLT_PT_SOC_FUNC query:



```
SELECT CODE1 LLT_CODE, GET_CODE_LLT (CODE1, 4) PT_CODE, SUBSTR(GET_TERM (CODE1 ,5),1,50)  LLT_TERM,  SUBSTR
```

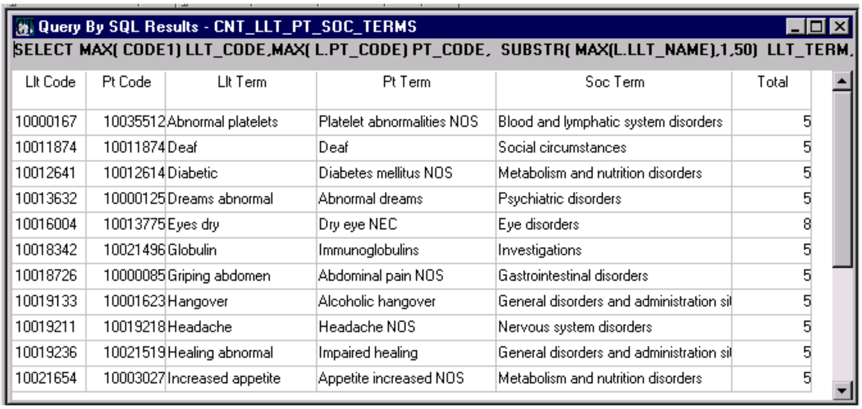
Llt Code	Pt Code	Llt Term	Pt Term	Soc Term	Total
10000167	10035512	Abnormal platelets	Platelet abnormalities NOS	Blood and lymphatic system disorders	5
10011874	10011874	Deaf	Deaf	Social circumstances	5
10012641	10012614	Diabetic	Diabetes mellitus NOS	Metabolism and nutrition disorders	5
10013632	10000125	Dreams abnormal	Abnormal dreams	Psychiatric disorders	5
10016004	10013775	Eyes dry	Dry eye NEC	Eye disorders	8
10018342	10021496	Globulin	Immunoglobulins	Investigations	5
10018726	10000085	Griping abdomen	Abdominal pain NOS	Gastrointestinal disorders	5
10019133	10001623	Hangover	Alcoholic hangover	General disorders and administration site conditions	5
10019211	10019218	Headache	Headache NOS	Nervous system disorders	5
10019236	10021519	Healing abnormal	Impaired healing	General disorders and administration site conditions	5
10021654	10003027	Increased appetite	Appetite increased NOS	Metabolism and nutrition disorders	5

The CNT_LLT_PT_SOC_TERMS query

The CNT_LLT_PT_SOC_TERMS query can be used if either one or two- part coding was done against the LLTs. For example, for every Lowest Level code in the ADV_LLT panel of MEDIKA_CLINICAL, the query will return matching terms found in the L_LOW_LEVEL_TERM_DATA panel, the PREF_TERM_DATA panel, and the L_MD_HIERARCHY_DATA panel in CT_MEDDRA. The following information will be reported:

- All LLTs used during coding
- The number of times each LLT was used
- The PT associated with every LLT
- The Preferred SOC term associated with every PT

The following example shows the results of a CNT_LLT_PT_SOC_TERMS query:



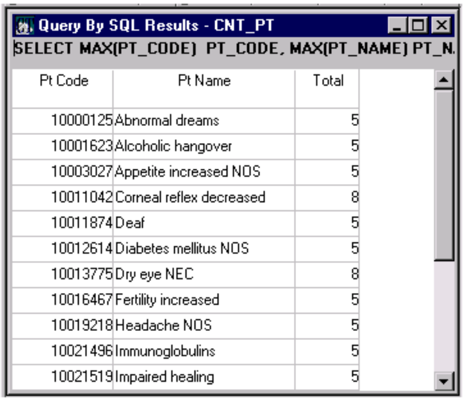
```
SELECT MAX( CODE1) LLT_CODE,MAX( L.PT_CODE) PT_CODE,  SUBSTR( MAX(L.LLT_NAME),1,50)  LLT_TERM,
```

Llt Code	Pt Code	Llt Term	Pt Term	Soc Term	Total
10000167	10035512	Abnormal platelets	Platelet abnormalities NOS	Blood and lymphatic system disorders	5
10011874	10011874	Deaf	Deaf	Social circumstances	5
10012641	10012614	Diabetic	Diabetes mellitus NOS	Metabolism and nutrition disorders	5
10013632	10000125	Dreams abnormal	Abnormal dreams	Psychiatric disorders	5
10016004	10013775	Eyes dry	Dry eye NEC	Eye disorders	8
10018342	10021496	Globulin	Immunoglobulins	Investigations	5
10018726	10000085	Griping abdomen	Abdominal pain NOS	Gastrointestinal disorders	5
10019133	10001623	Hangover	Alcoholic hangover	General disorders and administration si	5
10019211	10019218	Headache	Headache NOS	Nervous system disorders	5
10019236	10021519	Healing abnormal	Impaired healing	General disorders and administration si	5
10021654	10003027	Increased appetite	Appetite increased NOS	Metabolism and nutrition disorders	5

The CNT_PT query

The CNT_PT query can be used if either one or two-part coding was done against the PTs. For example, for every Preferred code in the ADV_PT panel of MEDIKA_CLINICAL, the query will return matching terms found in the L_PREF_TERM_DATA panel and the CT_MEDDRA thesaurus protocol panel in CT_MEDDRA. All PTs used during coding, and the number of times each PT term was used, will be reported.

The following example shows the results of a CNT_PT query:



```
SELECT MAX(PT_CODE) PT_CODE, MAX(PT_NAME) PT_N
```

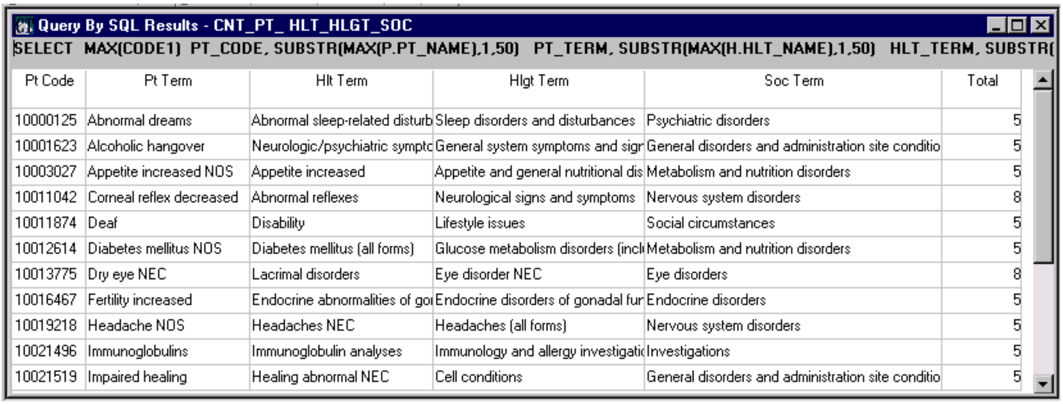
Pt Code	Pt Name	Total
10000125	Abnormal dreams	5
10001623	Alcoholic hangover	5
10003027	Appetite increased NOS	5
10011042	Corneal reflex decreased	8
10011874	Deaf	5
10012614	Diabetes mellitus NOS	5
10013775	Dry eye NEC	8
10016467	Fertility increased	5
10019218	Headache NOS	5
10021496	Immunoglobulins	5
10021519	Impaired healing	5

The CNT_PT_HLT_HLGT_SOC query

The CNT_PT_HLT_HLGT_SOC query can be used if either one or two-part coding was done against the PTs. For example, for every Preferred code in the ADV_PT panel of MEDIKA_CLINICAL, it will return matching terms found in the PREF_TERM_DATA panel in CT_MEDDRA. The following information will be reported:

- All PTs used during coding
- The number of times each PT was used
- The HLT associated with every PT
- The HLGT associated with every PT
- The Preferred SOC term associated with every PT

The following example shows the results of a CNT_PT_HLT_HLGT_SOC query:



Query By SQL Results - CNT_PT_ HLT_HLGT_SOC

```
SELECT  MAX(CODE1)  PT_CODE, SUBSTR(MAX(P.PT_NAME),1,50)  PT_TERM, SUBSTR(MAX(H.HLT_NAME),1,50)  HLT_TERM, SUBSTR(
```

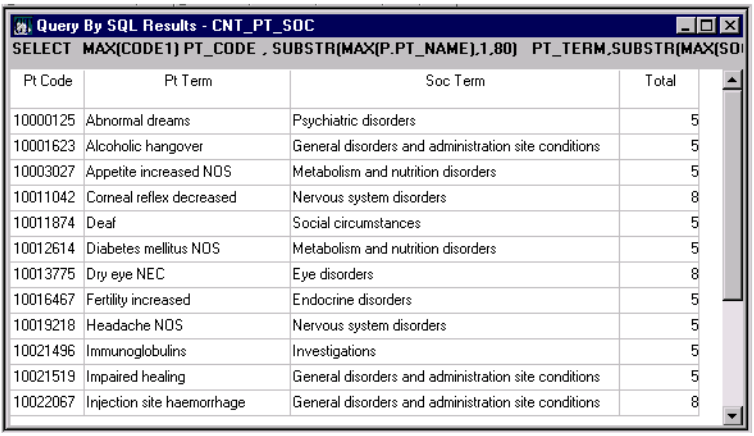
Pt Code	Pt Term	Hlt Term	Hlgt Term	Soc Term	Total
10000125	Abnormal dreams	Abnormal sleep-related disturb	Sleep disorders and disturbances	Psychiatric disorders	5
10001623	Alcoholic hangover	Neurologic/psychiatric symptc	General system symptoms and sigr	General disorders and administration site conditio	5
10003027	Appetite increased NOS	Appetite increased	Appetite and general nutritional dis	Metabolism and nutrition disorders	5
10011042	Corneal reflex decreased	Abnormal reflexes	Neurological signs and symptoms	Nervous system disorders	8
10011874	Deaf	Disability	Lifestyle issues	Social circumstances	5
10012614	Diabetes mellitus NOS	Diabetes mellitus (all forms)	Glucose metabolism disorders (incl	Metabolism and nutrition disorders	5
10013775	Dry eye NEC	Lacrimal disorders	Eye disorder NEC	Eye disorders	8
10016467	Fertility increased	Endocrine abnormalities of goı	Endocrine disorders of gonadal fur	Endocrine disorders	5
10019218	Headache NOS	Headaches NEC	Headaches (all forms)	Nervous system disorders	5
10021496	Immunoglobulins	Immunoglobulin analyses	Immunology and allergy investigatic	Investigations	5
10021519	Impaired healing	Healing abnormal NEC	Cell conditions	General disorders and administration site conditio	5

The CNT_PT_SOC query

The CNT_PT_SOC query can be used if either one or two-part coding was done against the PTs. For example, for every Preferred code in the ADV_PT panel of MEDIKA_CLINICAL, it will return matching terms found in the PREF_TERM_DATA panel in CT_MEDDRA. The following information will be reported:

- All PTs used during coding
- The number of times each PT was used
- The Preferred SOC term associated with every PT

The following example shows the results of a CNT_PT_SOC query:



```
SELECT  MAX(CODE1) PT_CODE , SUBSTR(MAX(P.PT_NAME),1,80)   PT_TERM,SUBSTR(MAX(SO
```

Pt Code	Pt Term	Soc Term	Total
10000125	Abnormal dreams	Psychiatric disorders	5
10001623	Alcoholic hangover	General disorders and administration site conditions	5
10003027	Appetite increased NOS	Metabolism and nutrition disorders	5
10011042	Corneal reflex decreased	Nervous system disorders	8
10011874	Deaf	Social circumstances	5
10012614	Diabetes mellitus NOS	Metabolism and nutrition disorders	5
10013775	Dry eye NEC	Eye disorders	8
10016467	Fertility increased	Endocrine disorders	5
10019218	Headache NOS	Nervous system disorders	5
10021496	Immunoglobulins	Investigations	5
10021519	Impaired healing	General disorders and administration site conditions	5
10022067	Injection site haemorrhage	General disorders and administration site conditions	8

25 Using Query By Form

Overview

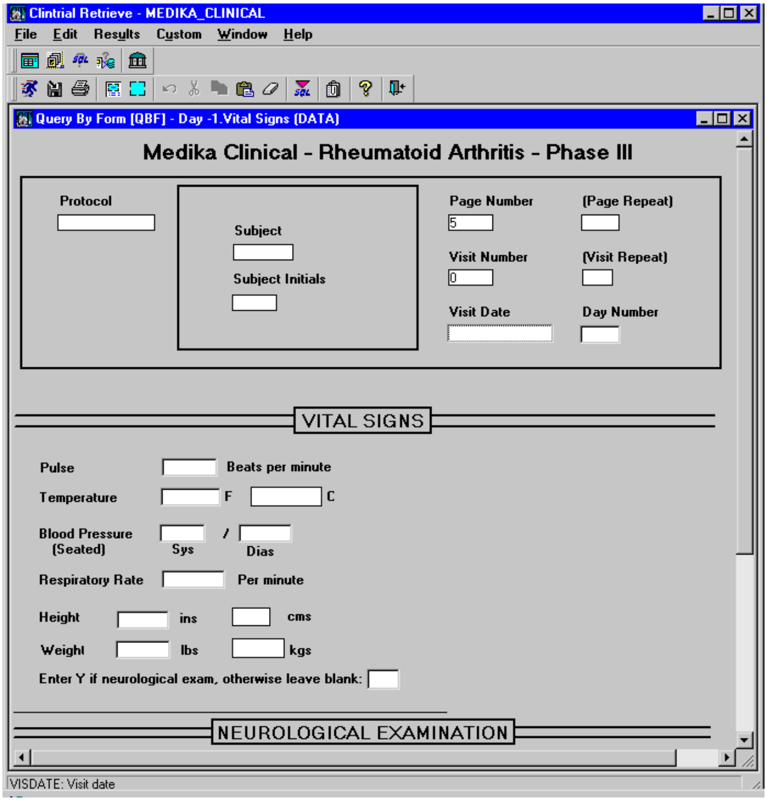
What is Query By Form (QBF)?

Query By Form (QBF) is a query method that helps you create a query using the same online study pages that you use to enter data interactively into your clinical data tables.

Using QBF, you can:

- Select the study pages that you want to use to retrieve data.
- Select the fields for which you want to retrieve data.
- Enter criteria in fields to specify a subset of data to retrieve.
- Create a sort order for the retrieved records.
- Run the query to retrieve the query results, and send the results to different locations in various export formats.
- Save the query to the database for later use by you or by other users.

The following example shows a QBF window for the study page Vital Signs:


Clintrial Retrieve - MEDIKA_CLINICAL
File
Edit
Results
Custom
Window
Help
Query By Form [QBF] - Day -1.Vital Signs (DATA)

Medika Clinical - Rheumatoid Arthritis - Phase III

Protocol

Subject

Subject Initials

Page Number: 5

(Page Repeat)

Visit Number: 0

(Visit Repeat)

Visit Date

Day Number

VITAL SIGNS

Pulse ______ Beats per minute

Temperature ______ F ______ C

Blood Pressure (Seated) ______ Sys / ______ Dias

Respiratory Rate ______ Per minute

Height ______ ins ______ cms

Weight ______ lbs ______ kgs

Enter Y if neurological exam, otherwise leave blank:

NEUROLOGICAL EXAMINATION

VISDATE: Visit date

Who should use QBF?

QBF is designed for those users who are familiar with their company's case report forms (CRFs) and data-entry standards, but are unfamiliar with either SQL or with the structure of the database tables.

How QBF retrieves records

When you create a QBF, you specify which records you want to retrieve by selecting fields and entering selection criteria in study pages. However, these study pages often refer to more than one database table.

For example, the study page Vital Signs/Neurological Exam in the MEDIKA study book refers to both the VITAL database tables and the NEUROL database tables. Selecting multiple pages also results in retrieving records from multiple database tables.

To help you retrieve meaningful data, QBF uses a default join to retrieve records.

- For panel Types 1 to 4, QBF implements a join on the SUBJECT_ID system item, by inserting the following WHERE condition:

  ```
  WHERE A.SUBJECT_ID = B.SUBJECT_ID
  ```

- For panel Types 3 and 4, QBF implements a join on the block key items and repeat key items, by inserting the following WHERE condition:

  ```
  WHERE A.SUBJECT_ID = B.SUBJECT_ID AND A.VISTNO = B.VISTNO
    AND A.VISTNO = B.VISITNO
    AND NVL(A.BLOCKREP, ‘.’ = NVL(B.BLOCKREP, ‘.’);
  ```

Note: Each QBF study page you open also automatically displays the corresponding block key item value and page key item value. You can delete these values if you want the study page to refer to multiple visits.

For more information on panel types, see *Admin and Design*. For more information on joins, see your Oracle SQL documentation.

For more information

For more information about study books and study pages, see "What are study books, blocks, and study pages?" on page 308.

Note: For more information on using study books and study pages, see Chapters 4, 5, and 11.

Creating a Query By Form

How to specify the type of table to query

Before you create a query using QBF, you must specify whether the clinical data that you want to query is in update tables, data tables, or both. To select the type of tables to query, do the following:

1. From the **Edit** menu, select **Preferences**.
2. In the Retrieve tab, set the RTV_DEF_SOURCE to one of the following:

- DATA — To select from clinical data tables of type Data
- UPDATE — To select from clinical data tables of type Update
- ALL — To select from both Data and Update tables

How to create a QBF

You create a QBF by performing the following steps:

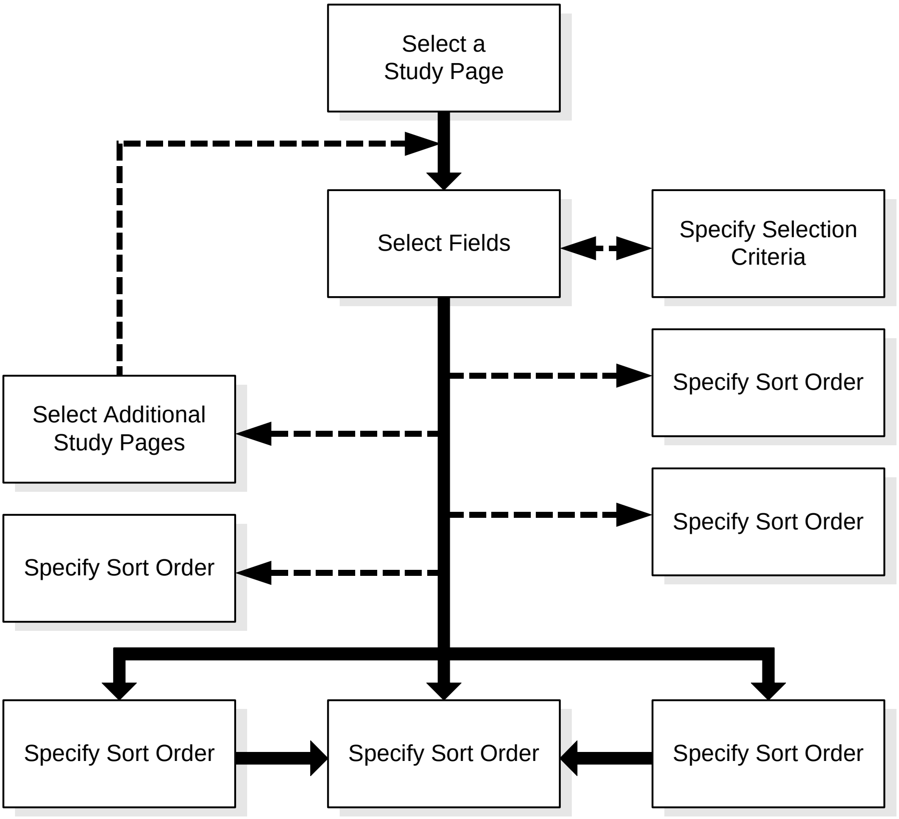
1. Select the study page with which you want to work.
2. Select the fields to retrieve.
3. Optionally, enter selection criteria.

Then, to enhance your QBF, you can perform optional tasks such as adding flag or note restrictions.

Note: Where appropriate, SQL notes in this section indicate how QBF tasks correspond to SQL statement clauses. However, you do not need to understand SQL to use QBF.

QBF workflow

The following figure shows the tasks you can perform using QBF. Dotted lines indicate optional tasks.

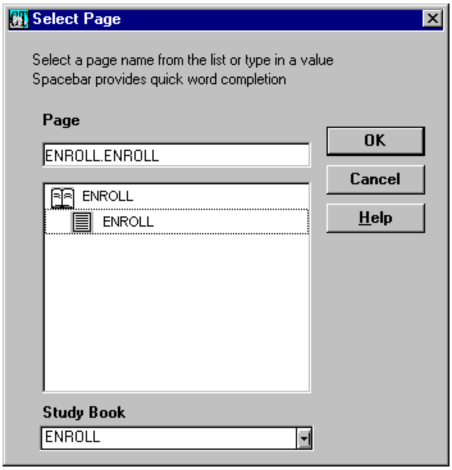


Selecting a study page

When you create a QBF, you work with the same study pages you use to enter data interactively. To begin creating the query, you must select a study page in the current protocol.

Note: Selecting a study page corresponds to creating a FROM clause in an SQL statement.

To select a study page, from the **File** menu, select **New Query** >> **By Form (QBF)**. The Select Page dialog box opens:



Select Page

Select a page name from the list or type in a value
Spacebar provides quick word completion

Page: ENROLL.ENROLL

- ENROLL
  - ENROLL

Study Book: ENROLL

This dialog box displays the default study book in the current protocol. To select a study page, click on the current study book, or select another study book from the Study Book drop-down list, then select the specific study book to use in your query.

You can add study pages to the query at a later time. For more information, see "Creating multipage queries" on page 343.

Selecting fields for retrieval

When you create a QBF, you select fields to specify what data you want to retrieve from the database.

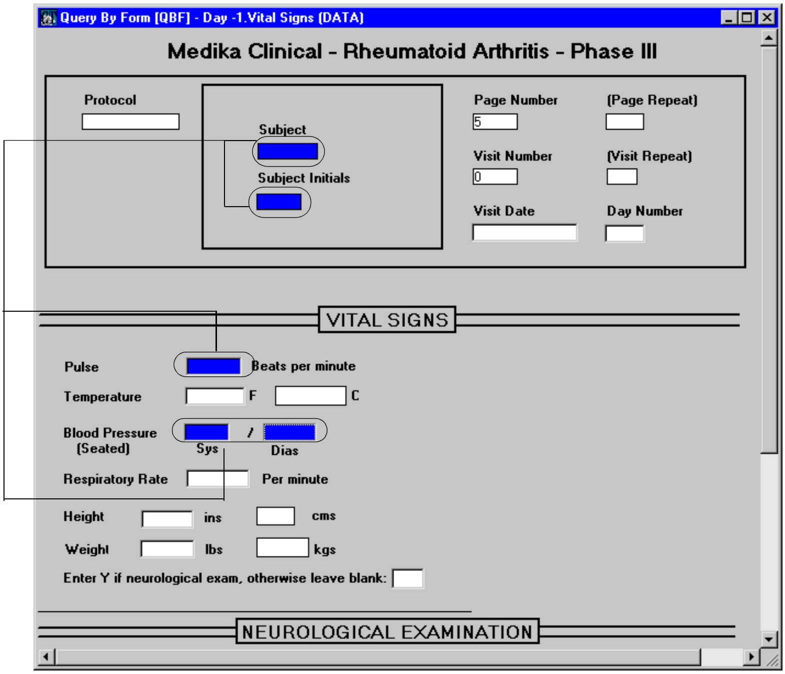
Note: Selecting fields corresponds to creating a SELECT clause in an SQL statement.

You select fields using the **Edit** menu's commands **Select Item** and **Select All**. When you select a field, Retrieve highlights that field. When you run the query, the Clintrial software retrieves only data for the selected fields. The fields appear in the query results table in the tab order defined for the study page.

You can deselect a field using the **Edit** menu's command **Select Item**. If all fields are selected, you can deselect them using the **Edit** menu's command **Select All**.

In the following example, the highlighted fields Subject, Subject Initials, Pulse, and Blood pressure seated are all selected for retrieval:

These fields are selected for retrieval.



Query By Form [QBF] - Day -1.Vital Signs (DATA)

Medika Clinical - Rheumatoid Arthritis - Phase III

Protocol

Subject

Subject Initials

Page Number: 5

(Page Repeat)

Visit Number: 0

(Visit Repeat)

Visit Date

Day Number

VITAL SIGNS

Pulse Beats per minute

Temperature F C

Blood Pressure (Seated) Sys / Dias

Respiratory Rate Per minute

Height ins cms

Weight lbs kgs

Enter Y if neurological exam, otherwise leave blank:

NEUROLOGICAL EXAMINATION

Note: If you select a field for retrieval, you do not have to specify selection criteria for that field.

In addition to selecting fields that appear on the study page, you can also retrieve system items (such as ENTRY_DATETIME and CT_RECID). The Retrieve user preference RTV_INCL_SYS determines whether the Clintrial software retrieves system item values when you run the query. These items appear as the first columns in the query results table. The default is No.

If you select a field for which a codelist is attached to the item, Retrieve uses the value of the user preference RTV_DEF_DECODE to determine which value to display in the query results table. The default is 1.

For more information on system items, see the *Reference Guide*.

For more information on navigating in study pages, see Chapter 3.

For more information on user preferences, see "Setting your user preferences" on page 315.

Entering selection criteria

When you create a query, you can specify the criteria you want to use to retrieve data. In QBF, you enter selection criteria by typing values directly into the fields on the study page.

Note: When you enter selection criteria in multiple fields, the Clintrial software adds the logical operator AND between the fields. If you want to change the query to use a logical OR between the fields instead, you can modify the query using Query By SQL.

Note: Entering selection criteria corresponds to creating a WHERE clause in an SQL statement.

You can enter selection criteria in the following ways:

- Enter an exact value.

  When you enter an exact value, you do not need to enclose the value in quotation marks (“ “).

  If a codelist or checklist is attached to the field, you can also select the value from the codelist by clicking the arrow, then selecting the value from the drop-down list.

- Enter a value pattern.

  Using the wildcard characters% (for multiple characters) and _ (for single characters), you can create a pattern that you want the values for fields of data type TEXT to match. If you use value patterns, you must use the relational operators LIKE and NOT LIKE, and enclose the text in single quotation marks (‘ ‘).

  Note: Values for fields of data type TEXT are case-sensitive. You must enter the pattern to match using the same case that is stored in the database.

- Enter relational operators and values.

  Text fields are case sensitive. You can enclose text strings in single quotes, but single quotes are not required.

  The following table lists the relational operators that you can use:

Operator:	**Description:**	**Example:**
=	Equal to the specified value	Systolic_bp = 130
<>	Not equal to the specified value	Systolic_bp <> 130

Operator:	Description:	Example:
>	Greater than the specified value	Systolic_bp > 130
>=	Greater than or equal to the specified value	Systolic_bp >= 130
<	Less than the specified value	Systolic_bp < 130
<=	Less than or equal to the specified value	Systolic_bp <= 130
BETWEEN *x* AND *y*	Is between (or equal to) the specified values	Systolic_bp BETWEEN 100 AND 135
IN	Is in the specified list of values	Systolic_bp IN (100, 125, 130, 135)
LIKE	Has the specified value pattern	Drug_name LIKE ‘ANTI%’
IS NULL	Has no value	Systolic_bp IS NULL
NOT BETWEEN *x* AND *y*	Is not between (or equal to) the specified values	Systolic_bp NOT BETWEEN 100 AND 135
NOT IN	Is not in the specified list of values	Systolic_bp NOT IN (100, 125, 130, 135)
NOT LIKE	Does not have the specified value pattern	Drug_name NOT LIKE ‘ANTI%’
IS NOT NULL	Has a value	Systolic_bp IS NOT NULL

Valid date formats

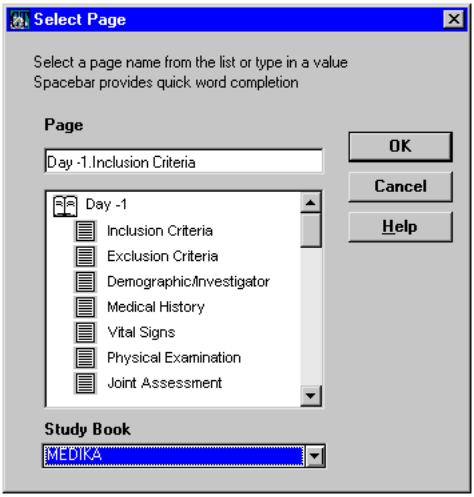
When you enter a value in a field of data type DATE, you can use the following date formats:

Date format:	**Example:**
’DD-Mon-YY’	’1-Aug-01’
Mon DD, YYYY	Aug 1, 2001
Mon DD, YY	Aug 1, 01
Month DD, YYYY	August 1, 2001
Month DD, YY	August 1, 01
MM/DD/YY	8/1/01
MM/DD/YYYY	8/1/2001
MM-DD-YY	8-1-01
MM-DD-YYYY	8-1-2001
BETWEEN ‘DD-Mon-YY’ AND ‘DD-Mon-YY’	BETWEEN ‘1-Aug-01’ AND ‘31-Aug-01’
BETWEEN MM/DD/YY AND MM/DD/YY	BETWEEN 8/1/01 AND 8/31/01
BETWEEN MM-DD-YY AND MM-DD-YY	BETWEEN 8-1-01 AND 8-31-01

Creating multipage queries

After you choose the primary study page, you can add study pages from study books in the protocol to create multipage queries.

To add additional study pages, from the **Edit** menu, select **Select Additional Page**. The Select Page dialog box opens:



Select Page

Select a page name from the list or type in a value
Spacebar provides quick word completion

Page: Day -1.Inclusion Criteria

- Day -1
  - Inclusion Criteria
  - Exclusion Criteria
  - Demographic/Investigator
  - Medical History
  - Vital Signs
  - Physical Examination
  - Joint Assessment

Study Book: MEDIKA

In this dialog box, you can select a new study page from either the same or a different study book in the current protocol.

Note: The number of study pages that you can open in the Clintrial software Enter or Retrieve is controlled by the user preference ENT_OPEN_PAGES. For more information, see "Setting your user preferences" on page 35.

After you have opened multiple study pages, you can use the **Window** menu to move among study pages in the query.

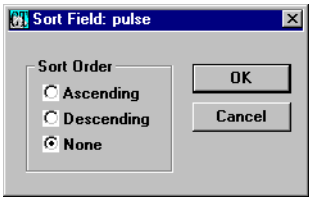
Specifying sort orders

When you run your query, Retrieve displays the query results in the order in which they appear in the database. You can specify a sort order for these records.

Note: Specifying a sort order corresponds to creating an ORDER BY clause in an SQL statement.

You sort the records in ascending or descending order by the values of specific fields. For example, you can sort the records in ascending order by visit number, then within each visit number by subject ID.

To specify the sort order for the values of a particular field, select the field. Then, from the **Edit** menu, select **Sort**. The Sort Field dialog box for the selected field opens:



Sort Field: pulse

Sort Order

○ Ascending

○ Descending

● None

The records in the query results table are sorted in the tab order set for fields on the study page. That is, if visit ID occurs in the study page tab order before subject ID, Retrieve sorts the records first by visit ID, then by subject ID.

You can sort by field in a different order by doing one of the following:

- Convert the query to SQL, then edit the ORDER BY clause.
- Run the query. Then, in the Query Results window, from the **View** menu, select **Sort**.

For more information on SQL, see the Oracle SQL documentation.

For more information on sorting, see "Sorting the query results" on page 406.

Querying with flags and notes

You can also add criteria to the query to retrieve records to which specific flags and notes are attached.

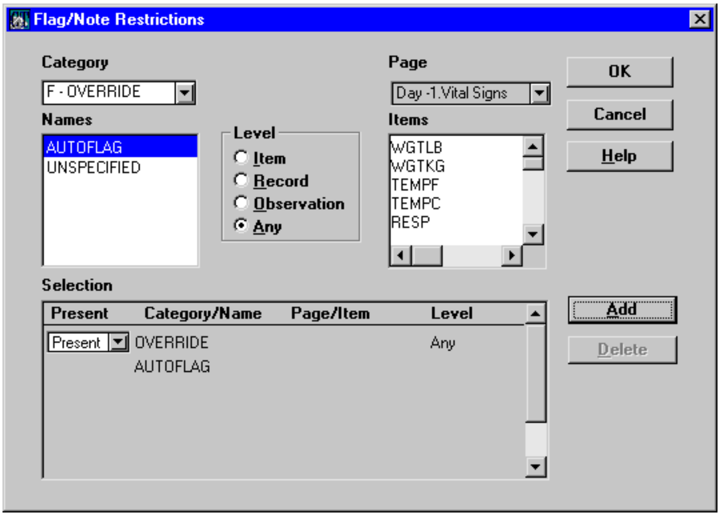
Flags and *notes* are information about clinical data attached to items, records, or observations. Flags and notes are used primarily to label and monitor data quality problems, and to record information not accommodated by a case report form (CRF) or the matching clinical data form.

Flags and notes differ as follows:

- A *flag* is a statement about clinical data. For example, a data-entry operator might add a flag to a field if data from the CRF is illegible, missing, or out of the expected range. A flag consists of a *flag category*, which characterizes the clinical data, and a *flag name*, which describes the action to be taken on the clinical data.
- A *note* is an annotation about clinical data, made by a sponsor or by the investigator. A note consists of a *note category*, which identifies the source of

the note as either Sponsor or Investigator, and a *note name*, which further characterizes the source of the note. Text is required when defining a note.

To create flag and note restrictions, from the **Edit** menu, select **Flags/Notes Restrictions**. The Flag/Note Restrictions dialog box opens:



Category: F - OVERRIDE

Names:
AUTOFLAG
UNSPECIFIED

Level:
○ Item
○ Record
○ Observation
● Any

Page: Day -1.Vital Signs

Items:
WGTLB
WGTKG
TEMPF
TEMPC
RESP

Selection

Present	Category/Name	Page/Item	Level
Present	OVERRIDE AUTOFLAG		Any

In this dialog box, you specify the following information:

- Category and category name

  You identify the flag or note for which you want to search by specifying the flag or note category and, if necessary, the flag or note name.

- Level

  Flags and notes can be attached to three levels of data. When creating the flag or note restriction, you specify either one of the three levels or all of the levels (by selecting **Any**).

- Page and item

  If the flag or note level for which you are creating the restriction is **Item** or **Any**, you specify which page and item you want associated with the flag or note restriction.

- Presence or absence of the flag or note

  After adding the flag or note restriction, you complete the restriction by specifying whether you want the query to retrieve records that either contain or do not contain the flag or note by selecting **Present** or **Absent**. For example, you can create a flag or note restriction that retrieves records for which a verification flag is attached at any level.

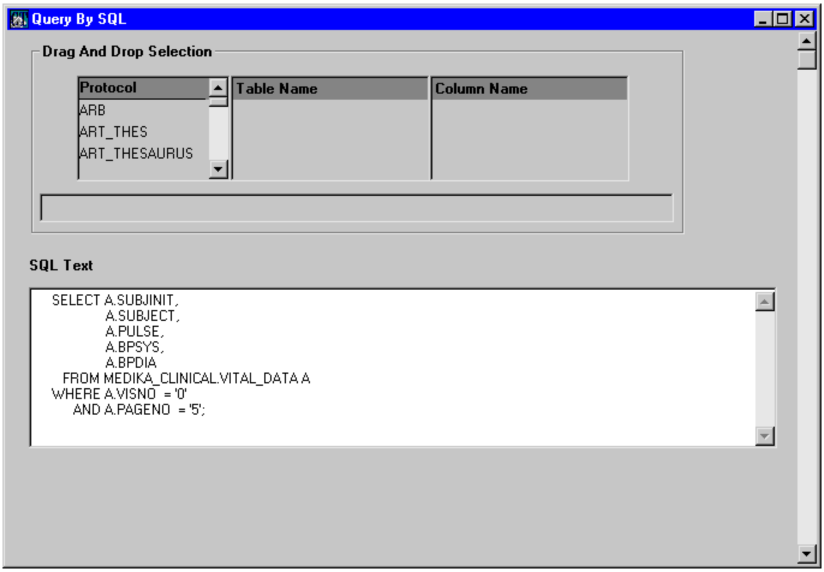
Displaying the query SQL statement

When you have finished creating the QBF, you may want to use SQL to modify certain characteristics of the query.

For example, using Query By SQL you can:

- Change the logical operator between fields from AND to OR.
- Change the sorting order of fields.
- Create joins within and between protocols.
- Add additional SQL selection criteria.

To display the SQL statement for the QBF, from the **File** menu select **Convert to SQL**. A Query By SQL text editor window for the query opens:



```
SELECT A.SUBJINIT,
            A.SUBJECT,
            A.PULSE,
            A.BPSYS,
            A.BPDIA
  FROM MEDIKA_CLINICAL.VITAL_DATA A
WHERE A.VISNO  = '0'
     AND A.PAGENO  = '5';
```

Caution: Any changes that you make to a query using SQL can appear only in the Query By SQL window. These changes will not appear in the QBF study pages. Once you convert a QBF to SQL, you do not return to the study pages to modify the query.

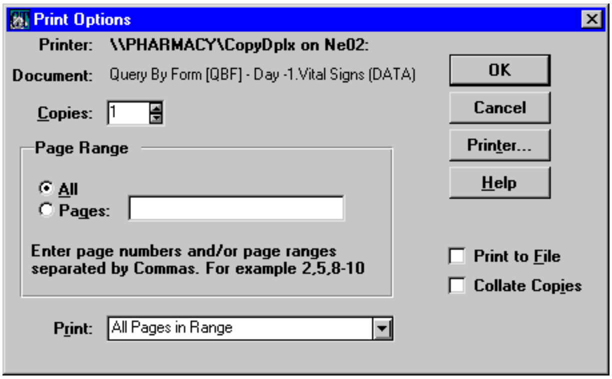
For information on using Query By SQL, see Chapter 26.

Printing a Query By Form

You can print the QBF study page to keep a paper record of your query.

Note: Clintrial prints what can fit on a page, as determined by paper size, page orientation (landscape vs. portrait), etc.

To print the query, from the **File** menu, select **Print**. The Print Options dialog box opens:



Print Options

Printer: \\PHARMACY\CopyDplx on Ne02:

Document: Query By Form [QBF] - Day -1.Vital Signs (DATA)

Copies: 1

Page Range

● All

○ Pages:

Enter page numbers and/or page ranges separated by Commas. For example 2,5,8-10

Print: All Pages in Range

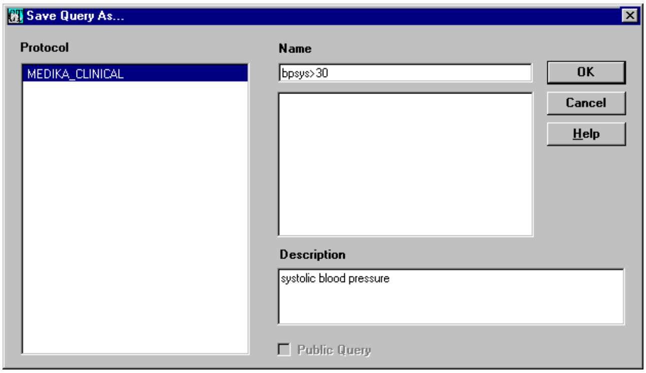
□ Print to File

□ Collate Copies

Saving a Query By Form

You can save any QBF to the Query Library, so that you or other users can use the query in the future.

To save a query, from the **File** menu, select **Save Query As**. The Save Query As dialog box opens:



Protocol: MEDIKA_CLINICAL

Name: bpsys>30

Description: systolic blood pressure

□ Public Query

In this dialog box, you specify the following:

- A name for the query

  The name can be up to 20 characters and is displayed in the Query Library.

- A description of the query

  The description can be up to 70 characters and is displayed in the Query Library.

- Whether to save the query as a public query or as a private query

  For more information on public and private queries, see "Types of security access" on page 319.

Once you save a QBF, you or another user with the appropriate access rights can modify and run it.

Note: When you open a saved QBF, the query SQL statement appears in a Query By SQL window. You cannot redisplay the query in QBF study page windows.

For more information on saved queries and the Query Library, see Chapter 23.

Running a Query By Form

Once you create the query, you can run it to obtain the query results.

To run the query, from the **File** menu, select **Run Query**. Retrieve runs the query and sends the results to the specified results destination.

For more information on results destinations and running queries, see Chapter 30.

26 *Using Query By SQL*

Overview

What is Query By SQL?

Query By SQL is a query method that allows you to enter SQL statements directly into an SQL text editor window.

You can also use the Query By SQL method to modify Query By Form and Query By Panel queries that have been saved or that you are currently creating.

Using Query By SQL, you can:

- Enter SQL SELECT and DESCRIBE statements.
- Create queries that retrieve data from multiple protocols.
- Use the logical operators AND and OR between items and protocols.
- Modify existing Query By SQL, Query By Form, or Query By Panel queries.
- Run the query to retrieve the query results, and send the results to different locations in various export formats.
- Save the query to the database for use by you or by other users.

Who should use Query By SQL?

Query By SQL is designed for those users who are familiar with SQL and with the structure of the database, and who want to create cross-protocol queries or modify existing queries.

Note: If you plan to create cross-protocol queries using Query By SQL, your user account must first have access rights to each protocol. However, for query access to multiple protocols, it is recommended that you assign the **Tools SQL** access right, rather than assign access rights for each protocol. Your Clintrial software administrator assigns this (non-protocol) access right in Admin.

For more information on user access rights, see *Admin and Design*.

Creating a Query By SQL

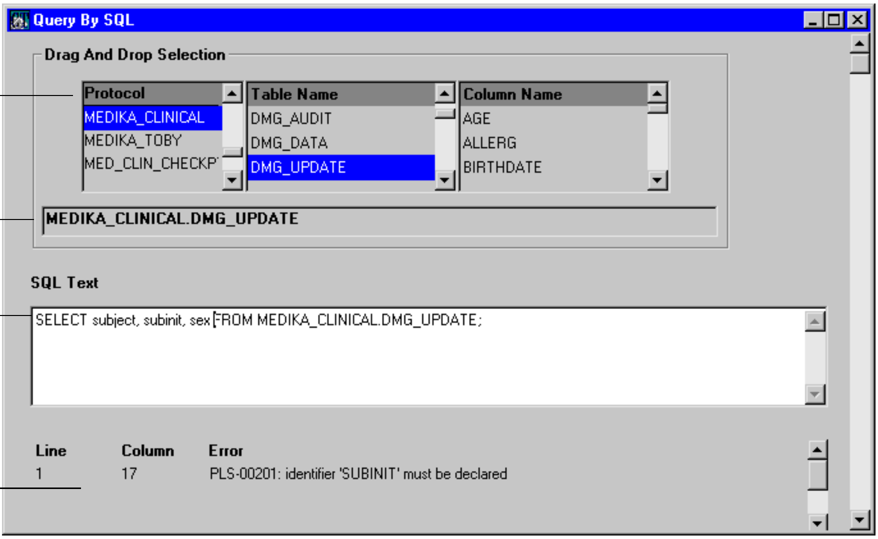
Overview

You create a Query By SQL using the Query By SQL text editor window. This window displays lists of the protocols, tables, and columns in the Clintrial software database, that you can use to help create your query. You can select the protocols, tables, and columns to create SQL text fragments, then drag and drop the SQL fragments into the SQL text area, where you can create a query including SELECT and DESCRIBE statements.

You can substitute the CTS$PROTOCOL parameter for the protocol name in a query and create global queries that operate on the current protocol for any given user.

How to create a Query By SQL

To open the Query By SQL text editor window, from the **File** menu, select **New Query**, then select **By SQL**. The Query By SQL window opens:



MEDIKA_CLINICAL.DMG_UPDATE

```
SELECT subject, subinit, sex FROM MEDIKA_CLINICAL.DMG_UPDATE;
```

```
Line    Column    Error
1       17        PLS-00201: identifier 'SUBINIT' must be declared
```

Protocol table and column lists

SQL fragment

SQL text area

SQL error messages appear in this area.

You create a Query By SQL by performing the following steps in the various areas within the Query By SQL window:

1. Select the protocol that you want to use for your query.
   (When you create a Query By SQL, the query is associated with the current protocol. You can use the CTS$PROTOCOL protocol parameter to operate on the current protocol for any user.)
2. Select the name of the database table for which you want to run the query.
3. Select the column name for the item that you want to retrieve.
4. Drag and drop the resulting SQL fragment from the SQL fragment area to the SQL text area.
5. Repeat for additional items.

   Note: See the following section for a simplified method of including multiple items in your SQL statement.
6. In the SQL text area, create a query including SELECT and DESCRIBE statements, including the closing semicolon (;).
7. Run the query. The SQL error checker then checks your SQL statement for SQL syntax errors and displays any resulting errors in the error message area below the SQL text area.

   You can edit your SQL statement, and then rerun the query. Once your SQL statement is free of errors, you can rerun the query to:

   - Display the SQL query results in a Query Results window, or
   - Send the results to an Oracle database table or a SAS Data file.

     Note: If you changed the original name of the protocol during the import, the query will not work. You must either edit the query by replacing the source protocol account name with the current protocol account name, or create a new query.

Example

Typically, if you use an SQL SELECT statement to retrieve records, you use the following format:

SELECT *item1, item2,...*FROM *protocol.table;*

in which *item1* and *item2* are column names for two items, *protocol* is the protocol account name, and *table* is the clinical data table.

For example:

```
SELECT SUBJECT, AGE FROM MEDIKA_CLINICAL.DMG_DATA;
```

However, if you use Query By SQL to create a SELECT statement by dragging and dropping the protocol name, table name, and column name, the syntax is more fully specified, as follows:

SELECT *protocol.table.item1, protocol.table.item2, ...*
FROM *protocol.table;*

For example:

```
SELECT MEDIKA_CLINICAL.DMG_DATA.SUBJECT,
MEDIKA_CLINICAL.DMG_DATA.AGE
FROM MEDIKA_CLINICAL.DMG_DATA;
```

You may find it more convenient to select only the protocol and table to create an SQL fragment to drag and drop for the FROM part of the statement, and use the column field only to view a list of the names of the items. Then, you can type in the names of the items in the SELECT part of the statement for the more concise syntax, without specifying the protocol and table with each item.

SQL statements

Types of SQL statements

When you create a query using Query By SQL, you can specify SQL SELECT and DESCRIBE statements:

- SELECT statements enable you to retrieve records from the database that meet certain conditions.
- DESCRIBE statements enable you to display the name and data type of all items in a database table, and determine whether the items can have a null value.

Statement syntax

To create an SQL statement, you need to know the following:

- The column names for the items that you want to retrieve
- The name of the protocol or account for which you want to run the query
- The name of the database table for which you want to run the query

The following example shows the syntax for a SELECT statement:

SELECT *column* FROM *account.table* WHERE *column operator value*;

The following example shows the syntax for a DESCRIBE statement:

DESCRIBE *account.table*;

Note: You must end SQL statements with a semicolon (;). Pressing **Enter** after typing a semicolon does not run the query. To run the query, from the **File** menu, select **Run Query**, or on the toolbar, click **Run**.

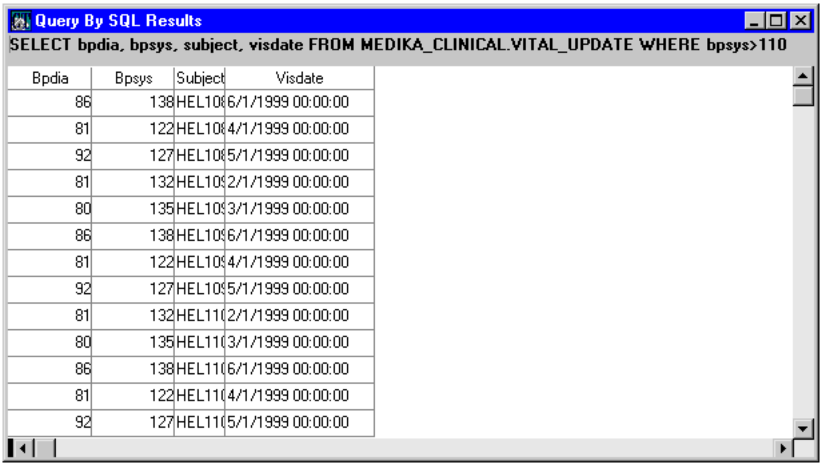
Examples

The following examples show the results of two SQL statements, where MEDIKA_CLINICAL is the study name and VITAL_UPDATE is the database table name.

If you enter the following SELECT statement:

```
SELECT bpsys, bpdia, subject, visdate
FROM MEDIKA_CLINICAL.VITAL_UPDATE WHERE bpsys >110;
```

and then from the **File** menu, select **Run Query**, the following query results are displayed:



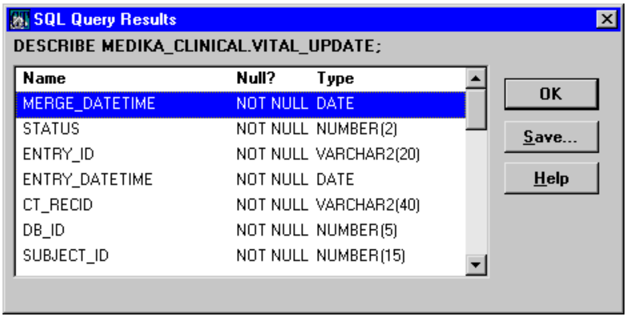
SELECT bpdia, bpsys, subject, visdate FROM MEDIKA_CLINICAL.VITAL_UPDATE WHERE bpsys>110

Bpdia	Bpsys	Subject	Visdate
86	138	HEL108	6/1/1999 00:00:00
81	122	HEL108	4/1/1999 00:00:00
92	127	HEL108	5/1/1999 00:00:00
81	132	HEL109	2/1/1999 00:00:00
80	135	HEL109	3/1/1999 00:00:00
86	138	HEL109	6/1/1999 00:00:00
81	122	HEL109	4/1/1999 00:00:00
92	127	HEL109	5/1/1999 00:00:00
81	132	HEL110	2/1/1999 00:00:00
80	135	HEL110	3/1/1999 00:00:00
86	138	HEL110	6/1/1999 00:00:00
81	122	HEL110	4/1/1999 00:00:00
92	127	HEL110	5/1/1999 00:00:00

If you enter the following DESCRIBE statement:

```
DESCRIBE MEDIKA_CLINICAL.VITAL_UPDATE;
```

and then from the **File** menu, select **Run Query**, the following query results are displayed:



DESCRIBE MEDIKA_CLINICAL.VITAL_UPDATE;

Name	Null?	Type
MERGE_DATETIME	NOT NULL	DATE
STATUS	NOT NULL	NUMBER(2)
ENTRY_ID	NOT NULL	VARCHAR2(20)
ENTRY_DATETIME	NOT NULL	DATE
CT_RECID	NOT NULL	VARCHAR2(40)
DB_ID	NOT NULL	NUMBER(5)
SUBJECT_ID	NOT NULL	NUMBER(15)

Decoding values

When a Query By Form or Ad Hoc Query retrieves an item with a codelist attached, it uses the user preference RTV_DEF_DECODE to determine which value to retrieve.

To retrieve the value, the underlying SQL uses a PL/SQL stored procedure named CT_DECODE in the SELECT statement. You can use this procedure in Query By SQL to retrieve a specific codelist value.

You use the following syntax for the CT_DECODE procedure:

CT_DECODE (*item_name*, *codelist_name*, *decode_type*)

Where *decode_type* can take one of the following values:

- V — The value of the item is retrieved.
- S — The short label of the item is retrieved.
- L — The long label of the item is retrieved.

For example, the following statement retrieves the items VISDATE and INCLUS1 from the Inclusion panel. The short label of the values YESNO are retrieved:

```
SELECT visdate, substr(ctproc.CT_DECODE(inclus1, ‘YESNO’, ’S’), 1,20)
”inclusion_1” FROM MEDIKA_CLINICAL.INCLUSION_UPDATE WHERE visdate =
‘0’;
```

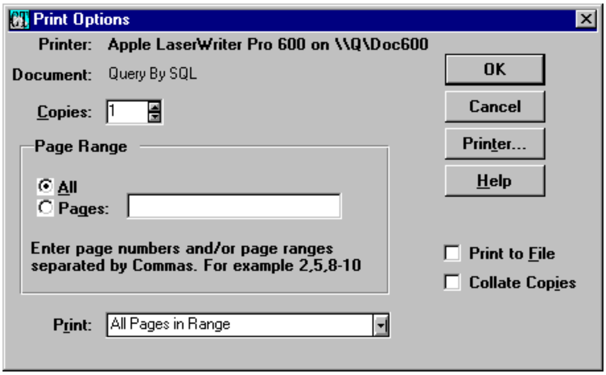
Note: In this example, the substr function is used to create a name for the column “inclus1” and to limit the length of the item to 20 characters in the query results.

For more information on user preferences, see "Setting your user preferences" on page 315.

Printing a Query By SQL

You can print the Query By SQL to keep a paper record of your query.

To print the query, from the **File** menu, select **Print**. The Print Options dialog box opens:



Printer: Apple LaserWriter Pro 600 on \\Q\Doc600

Document: Query By SQL

Copies: 1

● All

○ Pages:

Enter page numbers and/or page ranges separated by Commas. For example 2,5,8-10

□ Print to File

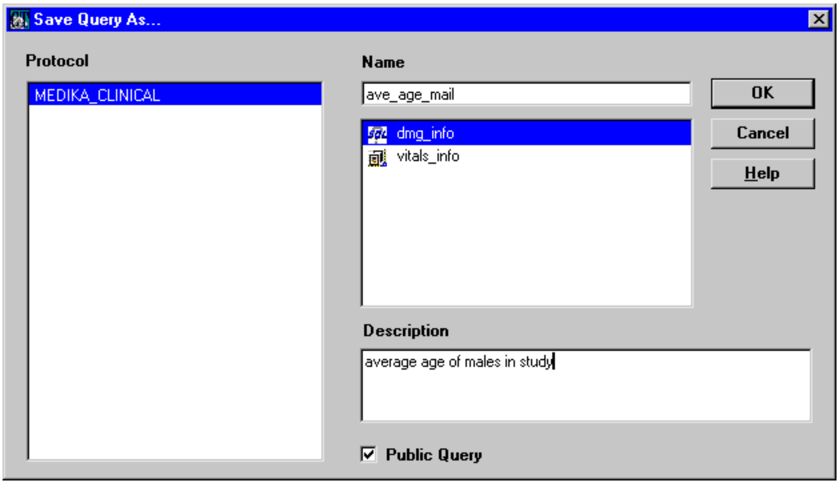
□ Collate Copies

Print: All Pages in Range

Saving a Query By SQL

You can save any Query By SQL to the Query Library, so that you or other users can use the query.

To save a query, from the **File** menu, select **Save Query As**. The Save Query As dialog box opens:



MEDIKA_CLINICAL

ave_age_mail

dmg_info

vitals_info

average age of males in study

☑ Public Query

In this dialog box, you specify the following:

- A name for the query

  The name can be up to 20 characters and is displayed in the Query Library.

- A description of the query

  The description can be up to 70 characters and is displayed in the Query Library.

- Whether to save the query as a public query or as a private query

  For more information on public and private queries, see "Types of security access" on page 319.

After you save a Query By SQL, you or another user with the appropriate access rights can modify and run it.

For more information on saved queries and the Query Library, see Chapter 24.

Running a Query By SQL

Once you create the query, you can run it to obtain the query results.

To run the query, from the **File** menu, select **Run Query**. Retrieve runs the query and sends the results to the specified results destination.

For more information on results destinations and running queries, see Chapter 30.

27 *Using Query By Panel*

Overview

What is Query By Panel (QBP)?

Query By Panel (QBP) is a query method that offers an easy to use interface to help guide nonprogrammers in developing queries for retrieval of Clintrial software data. Using QBP, you can specify panels and items from which to retrieve data and build an SQL statement that includes simple and outer joins, WHERE clauses, functions, and operators.

Using QBP, you can:

- Select multiple panels (up to eight) from which you want to retrieve data from a single protocol.
- Select multiple items within the selected panels from which you want to retrieve data.
- Build a complex query, using the Function Browser and the Select Builder.
- Assign group functions such as AVG, COUNT, and MAX to columns within tables.
- Add simple and outer joins to a query, enabling you to retrieve more specific results from your search.
- Run the query to retrieve the query results, and send the results to different locations in various export formats.
- Save the query to the database for use by you or by other users.
- Print the query.
- Open multiple QBPs at the same time.

Who should use QBP?

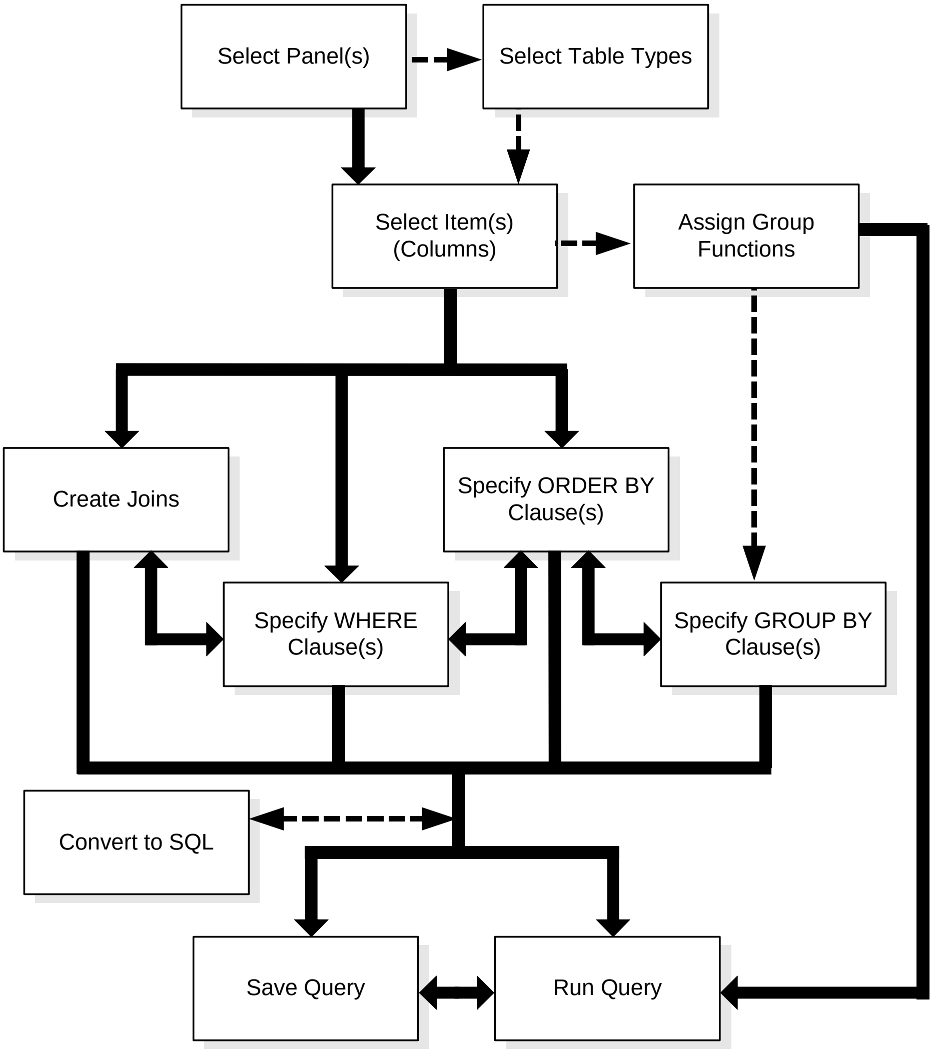
QBP is designed for the following types of users:

- Users who are familiar with panels and items within the Clintrial software and who know SQL syntax, but who are not necessarily proficient in SQL. Using QBP, you can define more detailed search criteria than you can using Query by Form (QBF), through the use of simple and outer joins, WHERE clauses, functions, and operators.
- Users who know SQL and want to build an SQL statement quickly using QBP’s convenient table joining and advanced querying functionality, and then add more detailed search criteria from within an SQL window.

Creating a Query By Panel

QBP workflow

The following figure shows the QBP workflow. Dotted lines indicate optional tasks.

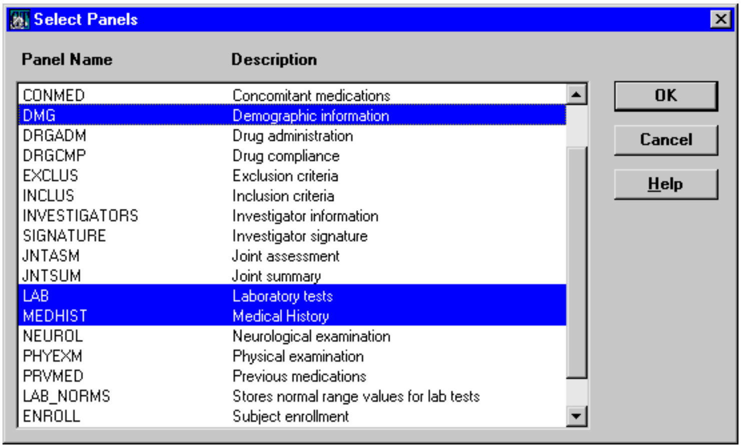


Caution: Once you convert a QBP to SQL, you cannot return to the QBP to modify it further.

Selecting panels

You create a QBP by selecting one or more panels from which to retrieve data in the Select Panels dialog box, which contains all panels defined in the current protocol. To open the Select Panels dialog box, from the **File** menu, select **New Query>>By Panel (QBP)**.

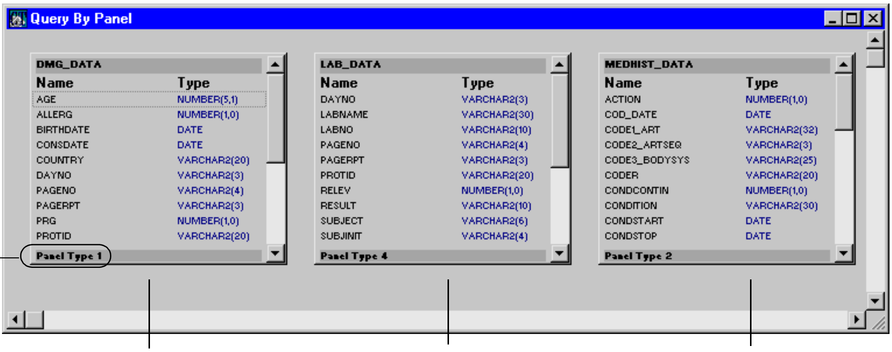
The Select Panels dialog box opens:



Panel Name	Description
CONMED	Concomitant medications
DMG	Demographic information
DRGADM	Drug administration
DRGCMP	Drug compliance
EXCLUS	Exclusion criteria
INCLUS	Inclusion criteria
INVESTIGATORS	Investigator information
SIGNATURE	Investigator signature
JNTASM	Joint assessment
JNTSUM	Joint summary
LAB	Laboratory tests
MEDHIST	Medical History
NEUROL	Neurological examination
PHYEXM	Physical examination
PRVMED	Previous medications
LAB_NORMS	Stores normal range values for lab tests
ENROLL	Subject enrollment

In this dialog box, you can select up to eight panels from which to retrieve data by using a detailed query. For example, if you want to run a query to retrieve data from panels with information on medical histories, demographic information, and laboratory exams, select **MEDHIST**, **DMG**, and **LAB**, and then click **OK**.

A window opens, displaying an Oracle database table for each panel selected in the Select Panels dialog box. For example, if you select **DMG**, **LAB**, and **MEDHIST,** there will be three tables, one for each panel selected.



DMG_DATA

Name	Type
AGE	NUMBER(5,1)
ALLERG	NUMBER(1,0)
BIRTHDATE	DATE
CONSDATE	DATE
COUNTRY	VARCHAR2(20)
DAYNO	VARCHAR2(3)
PAGENO	VARCHAR2(4)
PAGERPT	VARCHAR2(3)
PRG	NUMBER(1,0)
PROTID	VARCHAR2(20)

Panel Type 1

LAB_DATA

Name	Type
DAYNO	VARCHAR2(3)
LABNAME	VARCHAR2(30)
LABNO	VARCHAR2(10)
PAGENO	VARCHAR2(4)
PAGERPT	VARCHAR2(3)
PROTID	VARCHAR2(20)
RELEV	NUMBER(1,0)
RESULT	VARCHAR2(10)
SUBJECT	VARCHAR2(6)
SUBJINIT	VARCHAR2(4)

Panel Type 4

MEDHIST_DATA

Name	Type
ACTION	NUMBER(1,0)
COD_DATE	DATE
CODE1_ART	VARCHAR2(32)
CODE2_ARTSEQ	VARCHAR2(3)
CODE3_BODYSYS	VARCHAR2(25)
CODER	VARCHAR2(20)
CONDCONTIN	NUMBER(1,0)
CONDITION	VARCHAR2(30)
CONDSTART	DATE
CONDSTOP	DATE

Panel Type 2

Displays the panel type for each table.

Demographic Information table

Laboratory Examinations table

Medical History table

To select additional panels, from the **Panels** menu, select **Add Panel**.

Selecting the table type

You can retrieve data from the update table, the data table, or both (All). For each panel, you can select a different table from which to retrieve data. For example, you can set the default table type to the one that you will use most frequently; before running a query, you can override that default for individual tables as necessary.

To select the default table type from which to retrieve data for each panel, from the **File** menu, select **Preferences.** The Preferences window opens. On the Retrieve tab, set RTV_DEF_SOURCE to **UPDATE**, **DATA**, or **ALL** (to view results from both data tables and update tables). Changes to this setting will affect only those panels selected after the change is made; however, you can override individual panel settings.

Note: The default setting for RTV_DEF_SOURCE is DATA.

To override the default table type for individual panels, place your cursor anywhere in the table for which you want to select a type, then from the **Panel** menu, select **Source ALL**, **Source DATA**, or **Source UPDATE.** You must do this for every table in which you want to modify the default panel type.

Note: You can also select a table type by right clicking. Place your cursor within the panel title, then right click to make a selection from the resulting panel menu.

Selecting columns in tables

After you select panels from which you want to retrieve data, Retrieve displays the corresponding Oracle database tables. You can then select one or more columns within each Oracle database table, depending on the specific information you want to retrieve. To select all of the available columns, from the **Panel** menu, select **Select All**.

Note: The **Select All** command will select all available columns; however, system items do not automatically appear in the list of available columns. If you want to make system items available for your queries, from the **File** menu, select **Preferences** and set RTV_INCL_SYS to **Yes**.

Note: The default setting for RTV_INCL_SYS is No.

How the SQL Text box works

After you select one or more columns within a table, an SQL Text box opens, listing the SQL statement that corresponds to your selections.

Note: You may have to scroll down to view the SQL Text box. If you have selected fewer than five panels in your query, you can expand the SQL Text box upwards for better viewing. To do so, place your cursor on the top of the SQL Text box until it becomes a vertical two-edged arrow, then drag upwards.

You cannot modify the SQL statement directly in this box; however, you can modify the SQL statement by doing any of the following:

- Assigning group functions such as COUNT, AVG, and MAX to columns within a table by using the Function Browser. To display the Function Browser, from the **Edit** menu, select **Open Function Browser**.
- Adding more specific search criteria by using the Select Builder. To display the Select Builder, from the **Edit** menu, select **Open Select Builder**.
- Modifying the code directly in an SQL window. To open an SQL window, from the **Modify SQL** menu, select **Convert to SQL Window**.

Caution: Any changes you make to a query using SQL can only be displayed in the Query by SQL window. These changes will not display in QBP. Once you convert a QBP to SQL, you should not return to QBP to modify the query.

Assigning group functions in the Function Browser

Sometimes you may want to retrieve data from a table that does not relate specifically to what is contained in individual rows, but is instead a summary of the data in an entire column within a table. Group functions help you to retrieve such data. You assign group functions using the Function Browser.

Using group functions, you can summarize an entire column of values and return a single value. For example, if you assign the group function AVG (average) to a specific table and column, the value returned in the query will take all of the values within the specified column and return the average value.

Note: You can assign multiple group functions to a specific column. For example, if you assign a group function to determine the minimum age of the subjects in a study, then assign a group function to determine the maximum age of the subjects in a study, both group functions defined in the Function Browser are included in the SQL statement.

You also can assign an alias to the column. Use the alias to create a meaningful name for the column when the query results display. The alias cannot have any spaces between characters.

Example: AVG group function

Suppose you want to find the average age of a person in a clinical study. You can select the DMG panel from the Select Panels dialog box, and select the AGE column in the resulting Oracle database table. The following SQL statement appears in the SQL Text box:

```
SELECT MEDIKA_CLINICAL.DMG_UPDATE.AGE
   FROM MEDIKA_CLINICAL.DMG_UPDATE
```

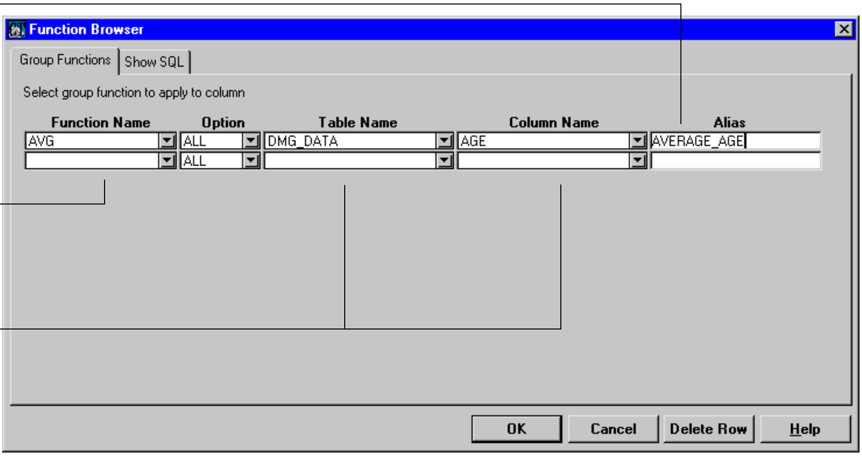
You then can assign a group function to the existing statement using the Function Browser. To display the Function Browser, from the **Edit** menu, select **Open Function Browser.** The Function Browser opens, with the Group Functions tab selected:

Name the column as you want it to appear in the query result.

Select the group function to include in your SQL statement.

Select the table and the column to which you want to assign the group function.

Delete any rows that you do not want to use.



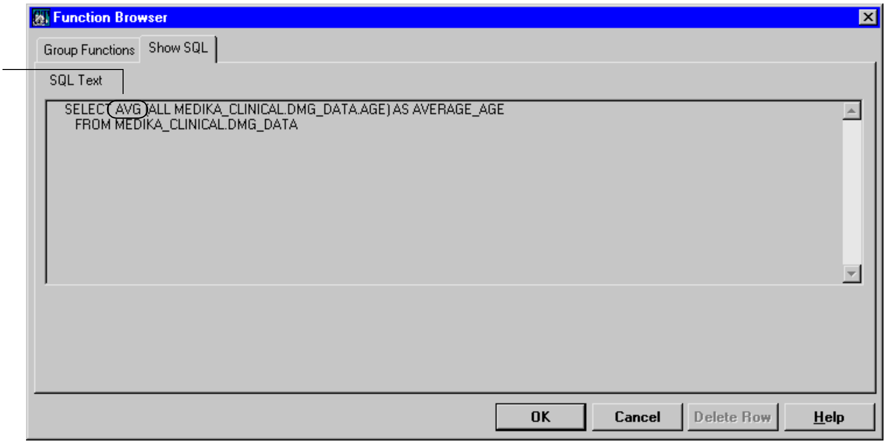
Select group function to apply to column

Function Name	Option	Table Name	Column Name	Alias
AVG	ALL	DMG_DATA	AGE	AVERAGE_AGE
	ALL			

Note: You can select either ALL or DISTINCT from the Option drop-down list. ALL retrieves all values in a column, including duplicate values. DISTINCT retrieves all unique values; it does not retrieve duplicate values.

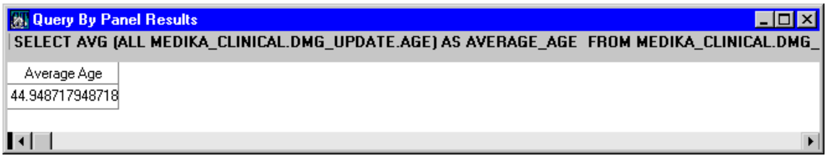
If you want to view the resulting SQL text before saving it, you can select the Show SQL tab, which is a preview of the completed SQL statement:

The resulting SQL statement displays in the SQL Text box, including the AVG group function.



```
SELECT AVG(ALL MEDIKA_CLINICAL.DMG_DATA.AGE) AS AVERAGE_AGE
  FROM MEDIKA_CLINICAL.DMG_DATA
```

After running the preceding query, the following results are returned, indicating that the average age is slightly less than 45 years old:



```
SELECT AVG (ALL MEDIKA_CLINICAL.DMG_UPDATE.AGE) AS AVERAGE_AGE  FROM MEDIKA_CLINICAL.DMG_
```

Average Age
44.948717948718

Note: To display the complete number, you may need to use your mouse to drag the column to the right.

You can use the Function Browser to assign any number of the following group functions to your SQL statement:

Group Function:	**Description:**
AVG	Returns the selected column’s average value, ignoring null values.
COUNT	Counts the number of rows in the specified column, ignoring null values.
MAX	Returns the maximum value in the selected column.
MIN	Returns the minimum value in the selected column.
STDDEV	Returns the standard deviation of the values in the selected column.
SUM	Returns the total of all values in the selected column.
VARIANCE	Returns the variance of the values in the selected column. Oracle uses a special formula to calculate the variance.

For more information on group functions, see the Oracle SQL documentation.

Using the Select Builder

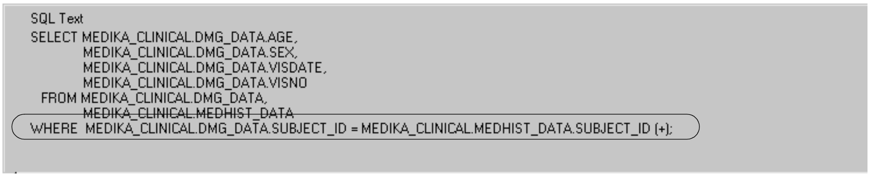
You can specify detailed search criteria for your query by using simple and outer joins, WHERE clauses, and other operators. You can also specify the order in which your results are returned. And, if you are using group functions in your query, you can specify how to group selected rows and return single rows of summary information. You enter all of this information on the corresponding tabs within the Select Builder. To display the Select Builder, from the **Edit** menu, select **Open Select Builder.**

What are joins?

Joins are used when a query requires information from two or more tables which have one variable in common. In QBP, you can retrieve information from multiple tables by using both simple joins and outer joins.

QBP's default joins

Whenever you select multiple tables for your query, your resulting SQL statement is created with a default join:



Default join

```
SELECT MEDIKA_CLINICAL.DMG_DATA.AGE,
       MEDIKA_CLINICAL.DMG_DATA.SEX,
       MEDIKA_CLINICAL.DMG_DATA.VISDATE,
       MEDIKA_CLINICAL.DMG_DATA.VISNO
  FROM MEDIKA_CLINICAL.DMG_DATA,
       MEDIKA_CLINICAL.MEDHIST_DATA
WHERE  MEDIKA_CLINICAL.DMG_DATA.SUBJECT_ID = MEDIKA_CLINICAL.MEDHIST_DATA.SUBJECT_ID (+);
```

To help you retrieve meaningful data, a default join is created for each QBP with multiple tables. This join can differ, depending on the panel type.

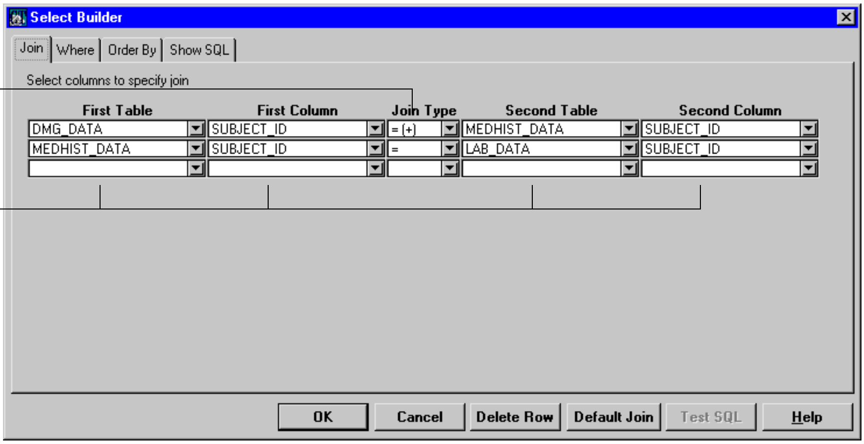
For panel types 1 to 4, QBP implements a join on the system item SUBJECT_ID.

For panel types 3 and 4, the default join includes two context items for each particular protocol, the system item SUBJECT_ID and the block key item.

You can edit the default join. If you edit the default join, you can recreate it in the Select Builder's Join tab by clicking **Default Join**.

Adding your own joins

You also can specify which joins are added to the existing SQL statement by selecting tables, columns, and join types on the Join tab:



First Table	First Column	Join Type	Second Table	Second Column
DMG_DATA	SUBJECT_ID	= (+)	MEDHIST_DATA	SUBJECT_ID
MEDHIST_DATA	SUBJECT_ID	=	LAB_DATA	SUBJECT_ID

Select the type of join that joins each selected table and column.

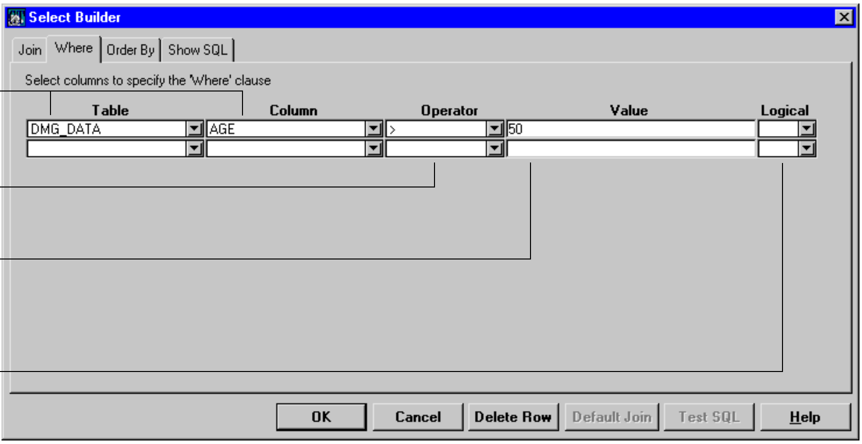
Select a table and column to join to a second table and column.

You can select any of the following Join types:

Join type:	**Description:**
= Simple Join	Returns all rows from both tables, where values for specific columns are equal.
(+) = Left Outer Join	Any unmatched rows from the left (first) table are preserved, but any unmatched rows from the right (second) table are discarded.
= (+) Right Outer Join	Any unmatched rows from the right (second) table are preserved, but any unmatched rows from the left (first) table are discarded.

Adding WHERE clauses

A WHERE clause specifies a condition that must be met in answer to your query. Using the Where tab of the Select Builder, you can build an expression that includes a table and column name, a relational operator, and a value:



Select columns to specify the 'Where' clause

Table	Column	Operator	Value	Logical
DMG_DATA	AGE	>	50	

Select tables and columns for the expression.

Select an operator for the expression.

Enter a value for which you want your expression to query.

If you are entering multiple rows, select AND or OR for your logical connector.

Note: An *expression* is a statement that consists of an item, a relational operator, and a value. For example, 'AGE > 50' is a simple expression. In QBP, expressions are joined using the logical operators AND and OR.

Note: The Value field is case sensitive. The default is lower case. If you want to query values that are in upper case, you must use upper case in the Value field.

Selecting operators

You select an operator for the expression by clicking an operator in the operator list.

The following table lists the relational operators that are available on the Where tab:

Relational operators:	**Expression retrieves records for which:**
=	The item value equals the specified value.
! and <>	The item value does not equal the specified value.

Retrieve

Relational operators:	Expression retrieves records for which:
>	The item value is greater than the specified value.
<	The item value is less than the specified value.
>=	The item value is greater than or equal to the specified value.
<=	The item value is less than or equal to the specified value.
IN	The item value is included within the specified set of values.
NOT IN	The item value is not included within the specified set of values.
BETWEEN	The item value is between or equal to the specified values. (Specify *x* AND *y* in the Value field.)
NOT BETWEEN	The item value is not equal to or between the specified values. (Specify *x* AND *y* in the Value field.)
EXISTS	The item value is TRUE if a subquery returns at least one row.
LIKE	The item value has the specified value pattern.
NOT LIKE	The item value does not have the specified value pattern.
IS NULL	No item value is present. You do not need to enter a value with this operator.
IS NOT NULL	Any item value is present. You do not need to enter a value with this operator.

Note: The relational operators ! and <> both return the same results; however, they may not be available on all platforms.

Grouping with the GROUP BY clause

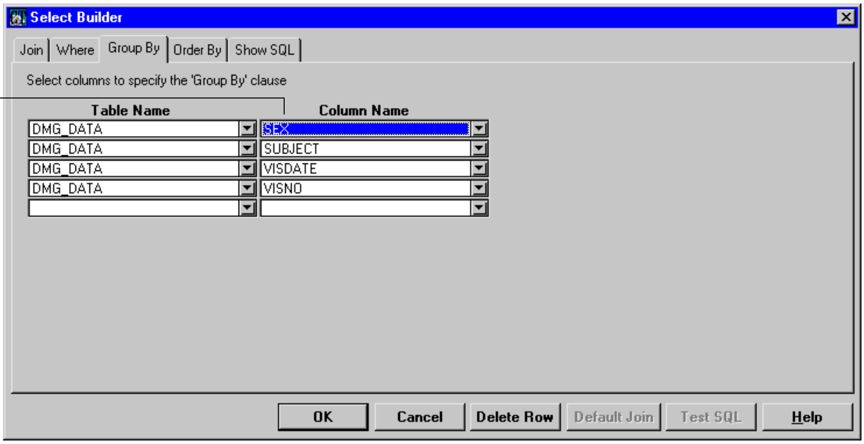
You can use the GROUP BY clause to group selected rows and return single rows of summary information. Oracle collects each group of rows based on the values of the expression(s) specified in the GROUP BY clause.

Note: The Group By tab only appears in the Select Builder if you have made selections in the Function Browser, and the values that appear in the Group By tab are based on whatever items are selected that were not included in the Function Browser.

Suppose you want to find the average age of individuals in a clinical study, grouped by male and female. In the Select Panels dialog box, you can select the DMG panel. From the resulting table, you select the AGE and SEX columns. Then, in the Function Browser, you can assign the AVG group function to the AGE column, while not applying any group function to the SEX column.

The Select Builder's Group By tab displays any remaining columns that you have selected in a table, but to which you have not assigned a group function:

Select the SEX column from the DMG table.



Select columns to specify the 'Group By' clause

Table Name	Column Name
DMG_DATA	SEX
DMG_DATA	SUBJECT
DMG_DATA	VISDATE
DMG_DATA	VISNO

In this case, the query will retrieve two average ages; one for men and one for women.

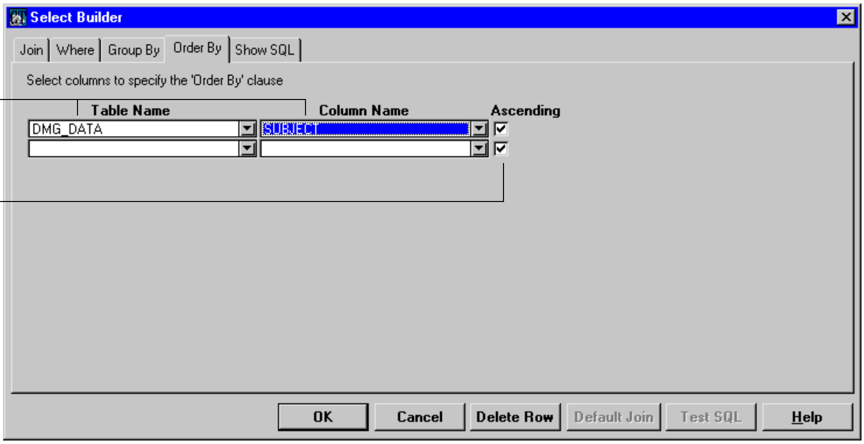
You can modify the columns on the Group By tab. However, to ensure that you are using the correct Oracle syntax, you should always test your changes using the **Test SQL** button on the Show SQL tab. If the syntax is incorrect, error messages will display.

Sorting rows with the ORDER BY clause

You can use the ORDER BY clause to sort the rows returned by a query. The ORDER BY clause specifies expressions in the select list of the statement. Oracle returns rows based on their values for these expressions.

You can specify multiple expressions using the ORDER BY clause. Oracle first sorts rows based on their values for the first expression. Rows with the same value for the first expression are then sorted based on their values for the second expression, and so on.

You can specify the order in which query results display for selected tables and columns using the Order By tab:



Select columns to specify the 'Order By' clause

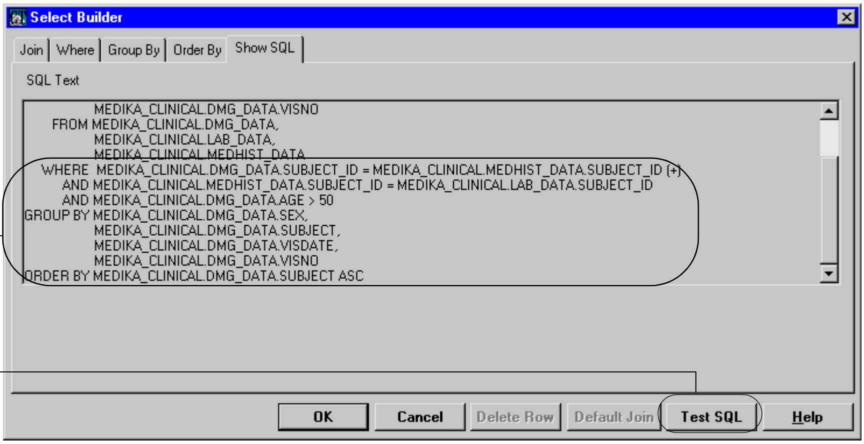
Table Name	Column Name	Ascending
DMG_DATA	SUBJECT	☑
		☑

Select the table and columns for which you want to set the order.

Select the order in which you want your query results for each row to display.

Viewing and testing SQL

After you make your selections on the various tabs of the Select Builder, you can view and test your SQL statement using the Show SQL tab:



```
            MEDIKA_CLINICAL.DMG_DATA.VISNO
     FROM MEDIKA_CLINICAL.DMG_DATA,
            MEDIKA_CLINICAL.LAB_DATA,
            MEDIKA_CLINICAL.MEDHIST_DATA
   WHERE  MEDIKA_CLINICAL.DMG_DATA.SUBJECT_ID = MEDIKA_CLINICAL.MEDHIST_DATA.SUBJECT_ID (+)
       AND MEDIKA_CLINICAL.MEDHIST_DATA.SUBJECT_ID = MEDIKA_CLINICAL.LAB_DATA.SUBJECT_ID
       AND MEDIKA_CLINICAL.DMG_DATA.AGE > 50
GROUP BY MEDIKA_CLINICAL.DMG_DATA.SEX,
            MEDIKA_CLINICAL.DMG_DATA.SUBJECT,
            MEDIKA_CLINICAL.DMG_DATA.VISDATE,
            MEDIKA_CLINICAL.DMG_DATA.VISNO
ORDER BY MEDIKA_CLINICAL.DMG_DATA.SUBJECT ASC
```

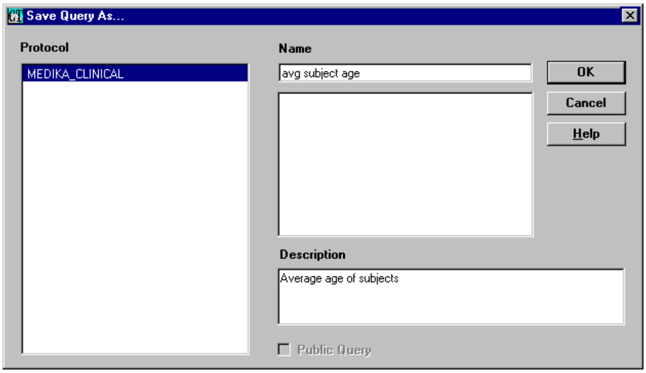
The SQL statement now contains the information specified on the various tabs.

Click to test your SQL statement for errors.

Saving a Query By Panel

You can save any QBP to the Query Library, so that you or other users can use the query.

To save a query, from the **File** menu, select **Save Query As**. The Save Query As dialog box opens:



Protocol: MEDIKA_CLINICAL

Name: avg subject age

Description: Average age of subjects

□ Public Query

In this dialog box, you specify the following:

- A name for the query

  The name can be up to 20 characters and is displayed in the Query Library.
- A description of the query

  The description can be up to 70 characters and is displayed in the Query Library.
- Whether to save the query as a public query or as a private query

After you save a QBP, you or another user with the appropriate access rights can modify and run it.

For more information on saved queries and the Query Library, see Chapter 24.

Running a Query By Panel

Once you create the query, you can run it to obtain the query results.

To run the query, from the **File** menu, select **Run Query**. Retrieve runs the query and sends the results to the specified results destination.

For more information on results destinations and running queries, see Chapter 30.

28 *Using Ad Hoc Query*

Overview

What is Ad Hoc Query?

Ad Hoc Query is a query method that allows you to use a graphical query builder to create lists of subjects, then browse through subject data.

Using Ad Hoc Query, you can:

- Specify selection criteria to retrieve a list of subject records from the database.
- Save the selection criteria for later use.
- Display a list of records for subjects that meet your selection criteria.
- Display detail data for subjects.
- Update the subject record list by redefining the selection criteria.
- Save the detail data and subject records for later use.

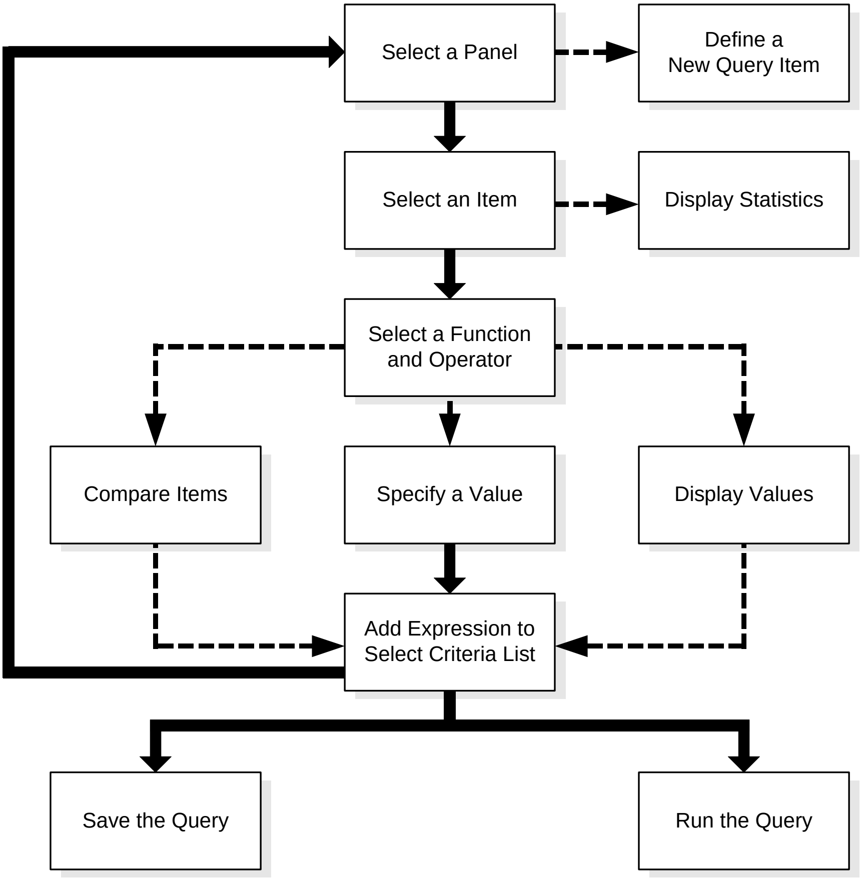
Who should use Ad Hoc Query?

Ad Hoc Query is designed for users who may not be familiar with SQL or with the online study pages, or for users who want to browse through Clintrial software data.

Creating an Ad Hoc Query

Selection Criteria workflow

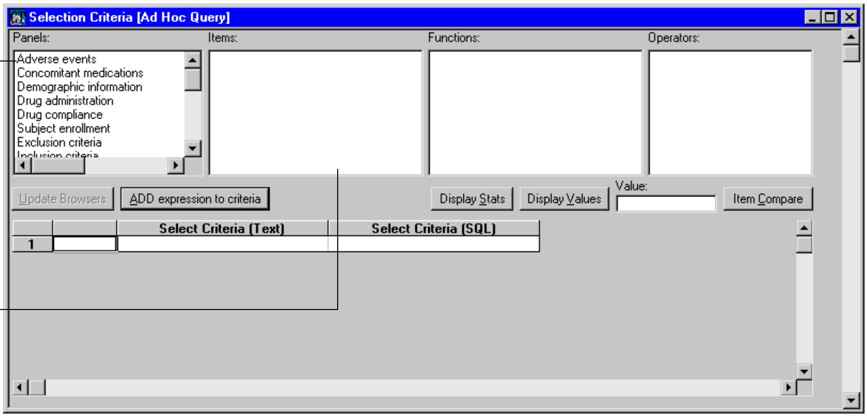
The following figure shows the Ad Hoc Query workflow. Dotted lines indicate optional tasks.



Overview

You create an Ad Hoc Query by building expressions, then adding the expressions to the Select Criteria list. An *expression* is a statement that consists of an item, a relational operator, and a value. For example, ‘AGE > 50’ is a simple expression. Expressions are joined using the logical operators AND, OR, and ANDSelect.

You create expressions in the Selection Criteria window. To open the Selection Criteria window, from the **File** menu, select **New Query >> Ad Hoc.** The Selection Criteria [Ad Hoc Query] window opens:



Panels list
(The panel that you select determines which items appear in the Items list.)

Items list

Selecting a panel

You select a panel for the expression from the Panel list, which contains all panels defined in the current protocol. The panel you select determines which items appear in the Items list.

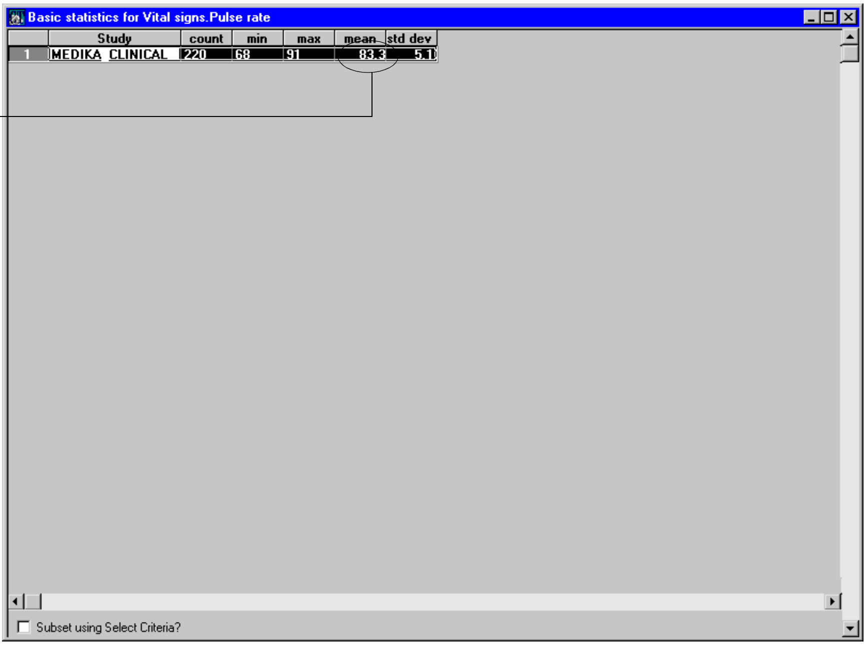
Selecting an item

You select an item for the expression by clicking an item name in the Items list. The item you select determines which functions, operators, and values you can use in the expression.

Displaying value statistics

You also can display value statistics for an item. Displaying statistics allows you to determine the maximum and minimum values allowed for an item, as well as an item's mean and standard deviation. The statistics for an item can help you determine which value you want to include in your expression.

For example, you may want to create an expression that retrieves all subjects whose pulse is greater than the mean value for the group. To determine the mean value of the pulse values stored in the database, you select Pulse, then click **Display Stats**. The Basic Statistics window for the selected item opens:



	Study	count	min	max	mean	std dev
1	MEDIKA CLINICAL	220	68	91	83.3	5.1

□ Subset using Select Criteria?

The mean value for the item Pulse is 83.3.

After reviewing the statistics, you can enter the mean value in the Selection Criteria window's Value field.

The Basic Statistics window displays the statistics based on all values for the item that exist in the database. You also can display a statistics subset that is based only on the values for the item in records that meet your current Select Criteria. To display a subset of the Basic Statistics list, select **Subset using Select Criteria**.

Selecting a function and operator

Functions determine whether the Select Criteria will retrieve records for which the item value matches the actual value you specify in the expression, or whether it matches some derived form of the value. The data type of the selected items determines the functions available for the expression.

Ad Hoc Query assigns the most likely function for a particular data type to the expression as a default. For example, the default function for Text items is actual value. You can use this default function, or select another function from the list.

The following table lists the functions that are available in Ad Hoc Query for the various data types:

Function:	**Description:**	**Available for:**
Actual Value	The value you enter represents the actual value of the item.	All data types
Decoded Value	The value you enter is the decoded value of the item. This function is available only if the item has an associated codelist.	TEXT NUMERIC
Calendar Year	The value you enter refers to the year portion of the DATE item.	DATE
Calendar Month	The value you enter refers to the month portion of the DATE item.	DATE
Calendar Day	The value you enter refers to the day portion of the DATE item.	DATE
Day of Week	The value you enter references the day of the week on which the DATE item value occurred.	DATE

You select an operator for the expression by clicking an operator in the operator list. The operators available for the expression are determined by the data type of the selected item. For example, operators such as “contains” and “begins with” are available for TEXT items only.

Note: You can use the wildcard characters % (for multiple characters) and _ (for single characters) with items of data type TEXT.

The following table lists the operators that are available in Ad Hoc Query for the various data types:

Relational operator:	**Expression retrieves records for which:**	**Available for:**
=	The item value equals the specified value.	All data types
<>	The item value does not equal the specified value.	All data types
<	The item value is less than the specified value.	All data types
>	The item value is greater than the specified value.	All data types
<=	The item value is less than or equal to the specified value.	All data types
>=	The item value is greater than or equal to the specified value.	All data types
is missing (NULL)	No item value is present. You do not need to enter a value with this operator.	All data types
is not missing (not NULL)	Any item value is present. You do not need to enter a value with this operator.	All data types
contains	The item value contains the specified text string.	TEXT
doesn’t contain	The item value does not contain the specified text string.	TEXT
begins with	The item value begins with the specified text string.	TEXT
ends with	The item value ends with the specified text string.	TEXT

Note: You cannot use the relational operators IN and BETWEEN in Ad Hoc Query.

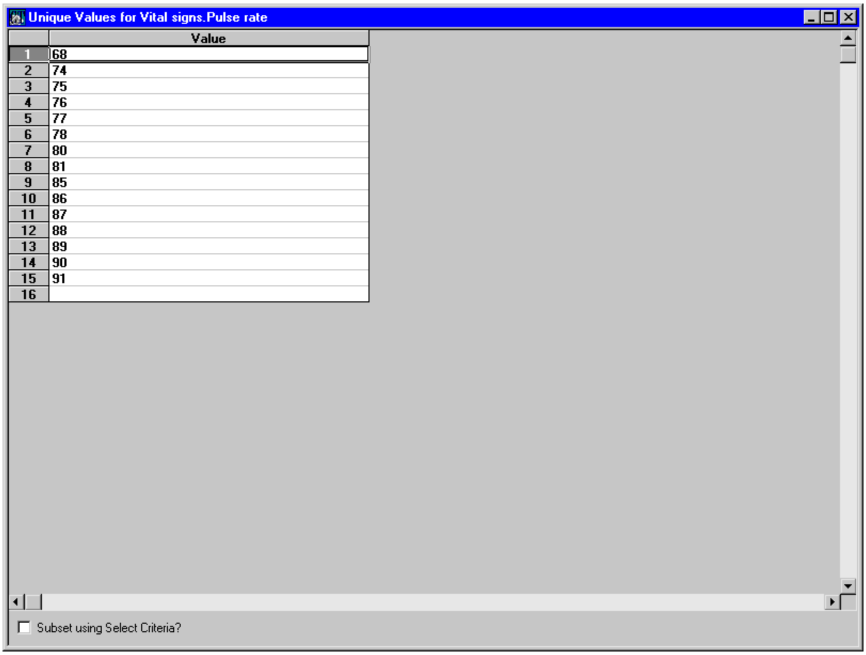
Specifying and displaying values

In the Selection Criteria window, you select values for the expression in one of two ways:

- Enter a value in the Value field.
- Select from a list of existing values.

Note: You also can create an expression in which you compare one item to another item, as opposed to a specific value. For more information, see "Comparing items" on page 387.

To display a list of values for the selected item, in the Selection Criteria window, click **Display Values**. The Unique Values window opens:



	Value
1	68
2	74
3	75
4	76
5	77
6	78
7	80
8	81
9	85
10	86
11	87
12	88
13	89
14	90
15	91
16	

□ Subset using Select Criteria?

The Unique Values window displays a list of all values that exist in the database for the item.

You also can display a subset of this list that includes only those values that exist for the items in records that meet your current Select Criteria. To display a subset of the Unique Values list, select **Subset using Select Criteria**.

Note: You must run the query before you can use this option. For more information, see "Running the query" on page 392.

To select a value from the Unique Values list to add to your expression, do one of the following:

- Click the value that you want. Then, to redisplay the Selection Criteria window, from the **Window** menu, select **Selection Criteria [Ad Hoc Query]**. When you click **Add Expression to Criteria**, Retrieve places the expression in the Select Criteria list.
- Double-click the value that you want. The expression is added automatically to the Select Criteria list.

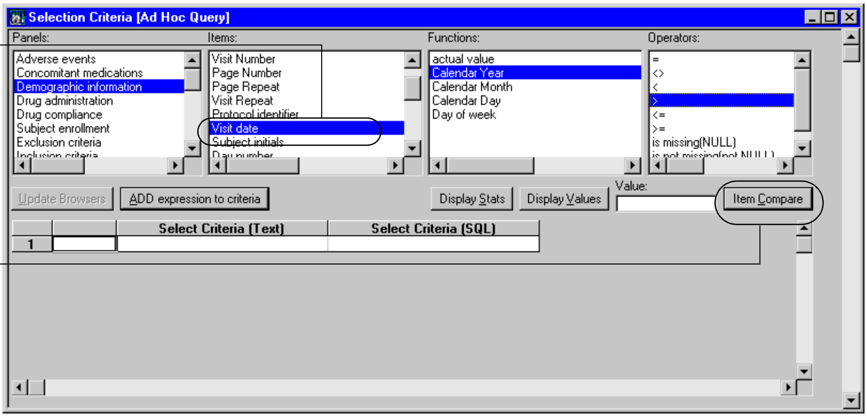
Note: If you have selected the operator "contains" or "doesn't contain", you can select multiple values from the Unique Values list at one time. All the values you select are then added to the expression.

To delete an expression from the selection criteria, select the row containing the expression. Then, from the **Edit** menu, select **Cut.**

Comparing items

You use Item Compare to create expressions in which you can compare items to one another, rather than to specific values. For example, you can create an expression that allows you to compare two items, and can retrieve results such as standingbp > sittingbp, rather than an expression that is dependent on a specific value, such as standingbp > 80.

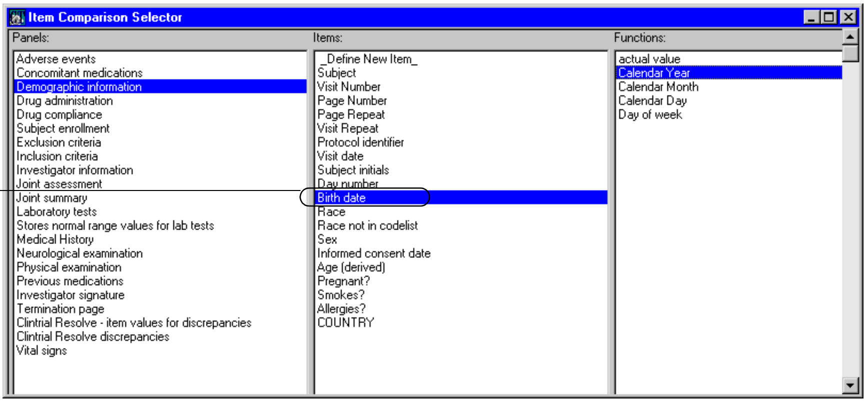
You specify the items for the first part of your expression using the Selection Criteria [Ad Hoc Query] window:



Select the item for the first part of your expression.

Click here to create the second part of your expression.

You specify the second part of your expression using the Item Comparison Selector window. In the Selection Criteria window, click **Item Compare.** The Item Comparison Selector window opens:



Item Comparison Selector

Panels:
- Adverse events
- Concomitant medications
- Demographic information
- Drug administration
- Drug compliance
- Subject enrollment
- Exclusion criteria
- Inclusion criteria
- Investigator information
- Joint assessment
- Joint summary
- Laboratory tests
- Stores normal range values for lab tests
- Medical History
- Neurological examination
- Physical examination
- Previous medications
- Investigator signature
- Termination page
- Clintrial Resolve - item values for discrepancies
- Clintrial Resolve discrepancies
- Vital signs

Items:
- _Define New Item_
- Subject
- Visit Number
- Page Number
- Page Repeat
- Visit Repeat
- Protocol identifier
- Visit date
- Subject initials
- Day number
- Birth date
- Race
- Race not in codelist
- Sex
- Informed consent date
- Age (derived)
- Pregnant?
- Smokes?
- Allergies?
- COUNTRY

Functions:
- actual value
- Calendar Year
- Calendar Month
- Calendar Day
- Day of week

Select the item for the second part of your expression.

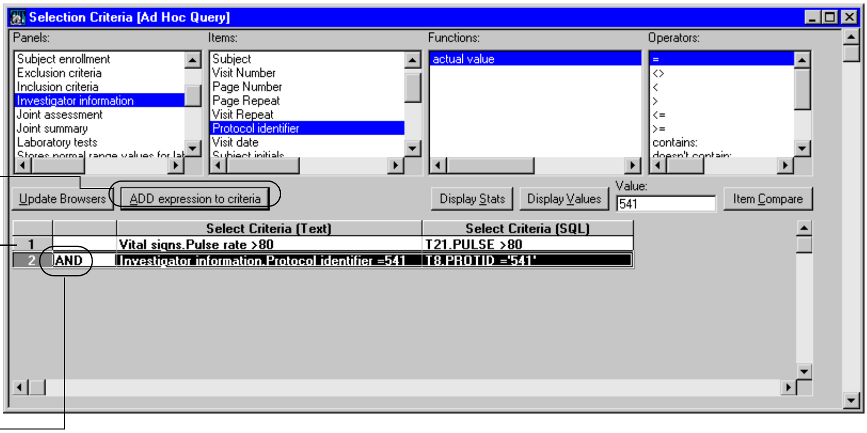
To add your selection to the Select Criteria list, you must first redisplay the Selection Criteria window. From the **Window** menu, select **Selection Criteria [Ad Hoc Query]**. When you click **ADD expression to criteria**, Retrieve places the expression in the Select Criteria list.

Note: Once you make a selection in the Item Comparison Selector window, the item you specify is compared with any subsequent selections you make in the Selection Criteria window. If you would like to continue your SQL statement with expressions containing specific values, you must first close the Item Comparison Selector window.

You cannot specify both a specific value and an item comparison within the same expression. If you specify an item comparison and you also specify a value in the Value field, the specific value overrides any selections that you make in the Item Comparison Selector window.

Adding an expression

After you specify an item, function, operator, and value, you add the expression to the Select Criteria list. To add an expression, click **ADD expression to criteria**. Ad Hoc Query then places the expression you created in the next available line of the Select Criteria list:



Value: 541

		Select Criteria (Text)	Select Criteria (SQL)
1		Vital signs.Pulse rate >80	T21.PULSE >80
2	AND	Investigator information.Protocol identifier =541	T8.PROTID ='541'

After you select the panel, item, function, operator, and value, click this button.

The expression is then added to the Select criteria list.

When you add multiple expressions, a logical operator field appears.

If you add more than one expression to the Select Criteria list, Ad Hoc Query automatically uses AND as a logical operator between the expressions. You can change this operator to OR or to ANDSelect.

The following logical operators determine how Retrieve selects subjects from the database:

- AND

  Use AND as a logical operator when you want to retrieve subject records that meet both the first expression criteria and the second expression criteria.

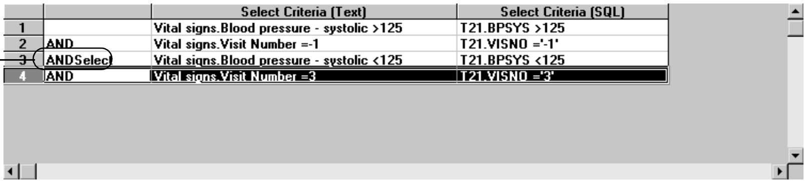
- OR

  Use OR as a logical operator when you want to retrieve subject records that meet either the first expression criteria or the second expression criteria. The records are not required to meet both expression criteria.

- ANDSelect

  Use ANDSelect as a logical operator when you want to retrieve subject records that meet criteria that may be mutually exclusive. ANDSelect creates selection segments that are evaluated in succession, so that one segment’s criteria do not exclude records that meet another segment’s criteria.

  For example, you may want to retrieve records for subjects whose systolic blood pressure was greater than 125 on the initial visit, but was less than 125 on the final visit. You can use ANDSelect to create an expression to retrieve those records:



ANDSelect logical operator

		Select Criteria (Text)	Select Criteria (SQL)
1		Vital signs.Blood pressure - systolic >125	T21.BPSYS >125
2	AND	Vital signs.Visit Number =-1	T21.VISNO ='-1'
3	ANDSelect	Vital signs.Blood pressure - systolic <125	T21.BPSYS <125
4	AND	Vital signs.Visit Number =3	T21.VISNO ='3'

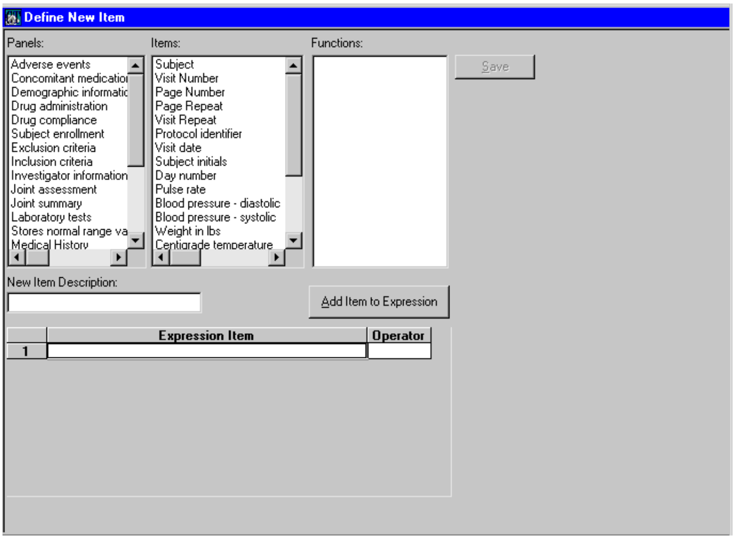
To create complex queries, you can add parentheses to the Select Criteria by entering the parentheses directly in the SQL column.

Defining new query items

You can define new query items whose values are derived from existing items. The items you use to derive the new query item can be in different panels. These derived values are stored in a file on your client and are available to your user account any time you use Ad Hoc Query, until you delete them. You also can edit any new query items you define.

For example, you may want to retrieve records for subjects that are over the age of 50. However, the protocol you are using does not store this value. You can define a new item in the DMG panel called Age, which derives its value from the items Birth Date and Visit Date. Then, you can use the new item to create your expression.

To define a new item, in the Selection Criteria window, select a panel in which to store the reference to the new item. Then, from the Items list, select **Define New Item**. The Define New Item dialog box opens:



Define your new item and specify a new item description, then click **Save**. The new item is automatically added to the end of the Items list in the Selection Criteria window.

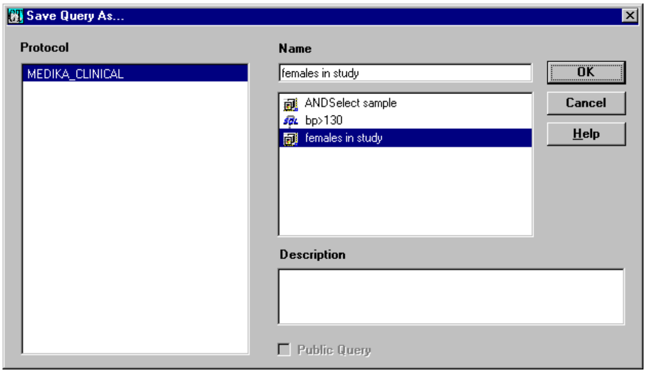
You also can edit any new query items you define. To edit a new item, in the Selection Criteria window, select it from the Items list, then from the **Edit** menu, select **New Item**.

Note: The query items you define also display in the detail data window when you select the associated panel. For more information, see "Displaying detail data for a subject" on page 395.

Saving the query

You can save any Ad Hoc Query to the Query Library, so that you or other users can use the query.

To save a query, from the **File** menu, select **Save Query As**. The Save Query As dialog box opens:



Protocol: MEDIKA_CLINICAL

Name: females in study

ANDSelect sample
bp>130
females in study

Description:

☐ Public Query

In this dialog box, you specify the following:

- A name for the query

  The name can be up to 20 characters and is displayed in the Query Library.
- A description of the query

  The description can be up to 70 characters and is displayed in the Query Library.
- Whether to save the query as a public query or as a private query

  For more information on public and private queries, see "Types of security access" on page 319.

Once you save an Ad Hoc Query, you or another user with the appropriate access rights can modify and run it.

For more information on saved queries and the Query Library, see Chapter 24.

Running the query

After you create the query, you can run it to obtain the query results.

To run the query, from the **File** menu, select **Run Query**. Retrieve runs the query and sends the results to the specified results destination.

Note: If the Data Browser window is already open, you can run the query by clicking **Update Browsers**.

If you send the results to a window, the Data Browser window opens and displays a list of subject records that meet the query criteria.

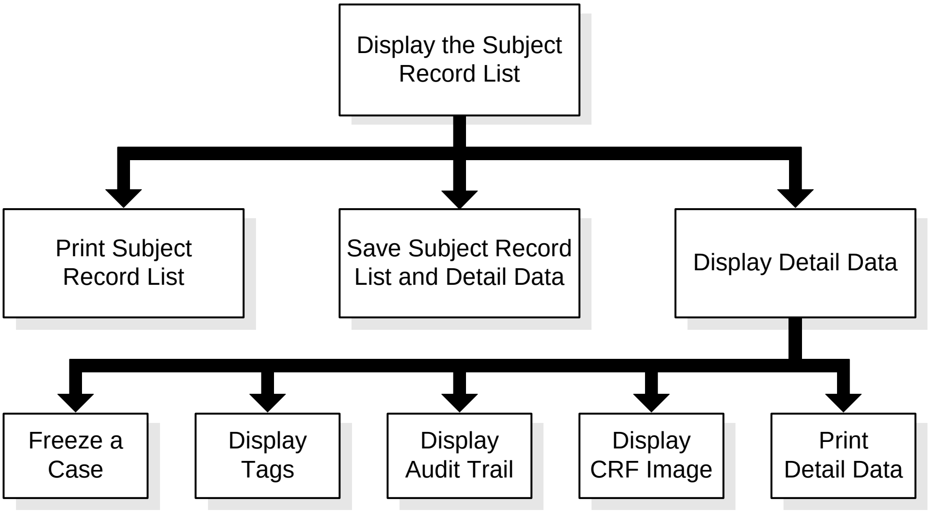
For more information on the Data Browser window, see the next section. For more information on results destinations and running queries, see Chapter 30.

Working with the Data Browser

Data Browser workflow

The Data Browser window displays subject records that meet the query criteria. You can use this subject record list to browse, print, and save subject data.

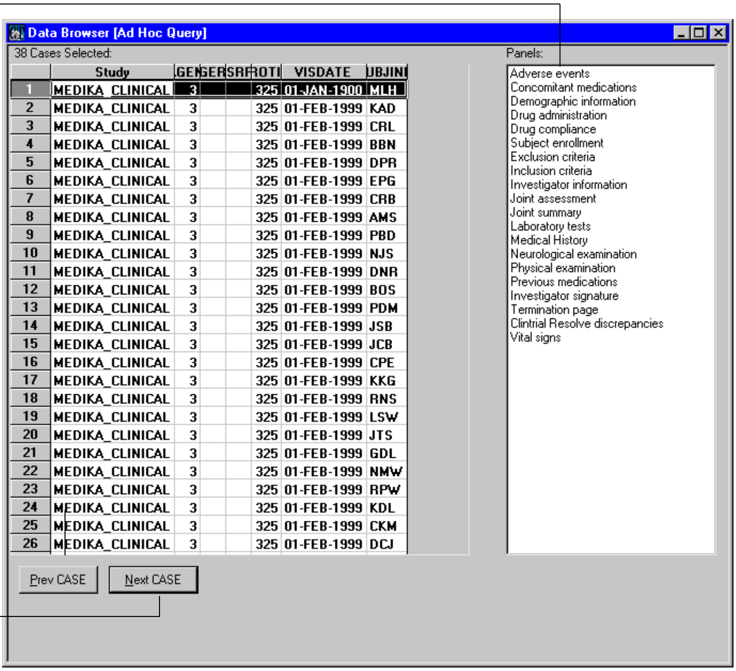
The following figure shows the Data Browser workflow:



Note: All of the tasks illustrated in this workflow are optional.

Displaying the subject record list

When you run an Ad Hoc Query and send the results to a window, the Data Browser window opens and displays a list of subject records that meet the query selection criteria:



You select panels from this list to display detail data for a case.

Subject records are also referred to as cases.

The subject record list displays a list of subject records that meet the query criteria.

Click to display data from a different case.

Data Browser [Ad Hoc Query]

38 Cases Selected:

	Study	GEN	GER	SRF	ROTI	VISDATE	UBJINI
1	MEDIKA_CLINICAL	3			325	01-JAN-1900	MLH
2	MEDIKA_CLINICAL	3			325	01-FEB-1999	KAD
3	MEDIKA_CLINICAL	3			325	01-FEB-1999	CRL
4	MEDIKA_CLINICAL	3			325	01-FEB-1999	BBN
5	MEDIKA_CLINICAL	3			325	01-FEB-1999	DPR
6	MEDIKA_CLINICAL	3			325	01-FEB-1999	EPG
7	MEDIKA_CLINICAL	3			325	01-FEB-1999	CRB
8	MEDIKA_CLINICAL	3			325	01-FEB-1999	AMS
9	MEDIKA_CLINICAL	3			325	01-FEB-1999	PBD
10	MEDIKA_CLINICAL	3			325	01-FEB-1999	NJS
11	MEDIKA_CLINICAL	3			325	01-FEB-1999	DNR
12	MEDIKA_CLINICAL	3			325	01-FEB-1999	BOS
13	MEDIKA_CLINICAL	3			325	01-FEB-1999	PDM
14	MEDIKA_CLINICAL	3			325	01-FEB-1999	JSB
15	MEDIKA_CLINICAL	3			325	01-FEB-1999	JCB
16	MEDIKA_CLINICAL	3			325	01-FEB-1999	CPE
17	MEDIKA_CLINICAL	3			325	01-FEB-1999	KKG
18	MEDIKA_CLINICAL	3			325	01-FEB-1999	RNS
19	MEDIKA_CLINICAL	3			325	01-FEB-1999	LSW
20	MEDIKA_CLINICAL	3			325	01-FEB-1999	JTS
21	MEDIKA_CLINICAL	3			325	01-FEB-1999	GDL
22	MEDIKA_CLINICAL	3			325	01-FEB-1999	NMW
23	MEDIKA_CLINICAL	3			325	01-FEB-1999	RPW
24	MEDIKA_CLINICAL	3			325	01-FEB-1999	KDL
25	MEDIKA_CLINICAL	3			325	01-FEB-1999	CKM
26	MEDIKA_CLINICAL	3			325	01-FEB-1999	DCJ

Panels:

Adverse events
Concomitant medications
Demographic information
Drug administration
Drug compliance
Subject enrollment
Exclusion criteria
Inclusion criteria
Investigator information
Joint assessment
Joint summary
Laboratory tests
Medical History
Neurological examination
Physical examination
Previous medications
Investigator signature
Termination page
Clintrial Resolve discrepancies
Vital signs

The subject record list contains key clinical data for each subject. The type of information that displays is determined by your system administrator. For more information, see *Admin and Design*.

Each column in the subject record list is sized automatically to the column length specified in the database. You can expand or contract any of the columns in the list.

Each time you modify the selection criteria in the Ad Hoc Query window, and then click **Update Browsers**, the Data Browser updates the subject record list to include subjects that meet the new selection criteria.

You can display detail data for a subject by selecting a panel, which displays item values from that panel for the selected subject. For more information, see "Displaying detail data" on page 395.

Printing the subject record list

To print the list of subjects that meets the current selection criteria, you click the title of the subject record list, and then from the **File** menu, select **Print**.

For more information on printing query results, see Chapter 30.

Saving the subject record list

You can save the list of subject records to a file, to an Oracle database table, or to a SAS Data file so that you can use the data in other applications.

For information on saving query results, see "Saving the query" on page 391.

Displaying detail data

Displaying detail data for a subject

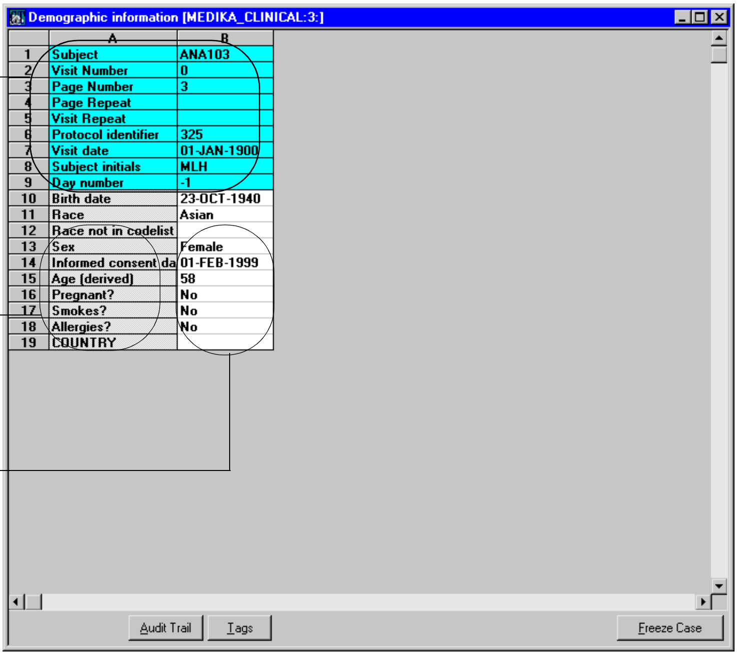
The subject record list displays only the key clinical data items specified by your system administrator. You also can display detailed data for a subject by selecting a subject, and then selecting a panel. Retrieve opens a window that displays the values of the panel's items for the selected subject.

Note: The Detail Data window also displays any query items you have created for the associated panel. For more information, see "Defining new query items" on page 390.

To select a subject, in the Detail Data window, click anywhere in the row for that subject. You also can use the **Prev CASE** and **Next CASE** buttons in the Data Browser window to view data for the previous or next subject in the list.

To select a panel, click one of the panels displayed in the Panel list. A new window opens, displaying a spreadsheet containing the descriptions and the actual data for the selected subject.

For example, the following Detail Data window shows the data from the Vital Signs panel for a subject:



Demographic information [MEDIKA_CLINICAL:3:]

	A	B
1	Subject	ANA103
2	Visit Number	0
3	Page Number	3
4	Page Repeat	
5	Visit Repeat	
6	Protocol identifier	325
7	Visit date	01-JAN-1900
8	Subject initials	MLH
9	Day number	-1
10	Birth date	23-OCT-1940
11	Race	Asian
12	Race not in codelist	
13	Sex	Female
14	Informed consent da	01-FEB-1999
15	Age (derived)	58
16	Pregnant?	No
17	Smokes?	No
18	Allergies?	No
19	COUNTRY	

The cells in the detail data table are highlighted to help you differentiate the data. For example, these cells contain context item data.

These cells contain the description of items.

These cells contain the clinical data for the panel.

If the selected panel contains multiple observations, as shown in the preceding example, all observations for the panel and subject appear in the panel view spreadsheet, ordered by visit date. You can expand or contract any of the columns in the spreadsheet.

Freezing a case

As you browse subject data in the Data Browser, the Detail Data windows are updated to reflect data for the currently selected subject. You also can browse data in multisubject mode, which allows you to review data for different subjects simultaneously.

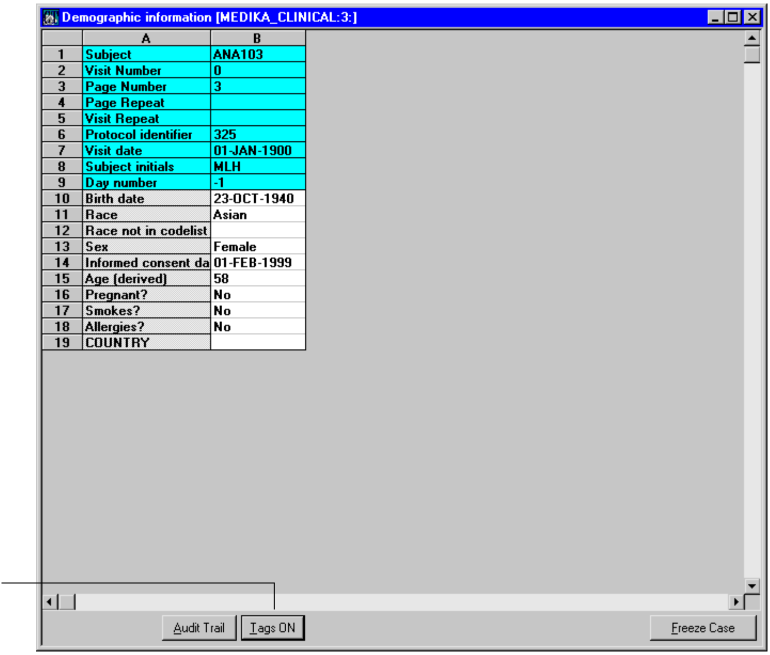
Each Detail Data window has a **Freeze Case** button. When you click **Freeze Case**, the button label changes to **Case Frozen**. At this point, the Detail Data window is frozen, and does not update when you select another subject. You can then open an additional Detail Data window for the panel to display data for another subject, while continuing to display the first subject’s detail data in the frozen Detail Data window.

Freezing a case allows you to display detail data from the same panel for many subjects simultaneously, providing you with a useful comparative clinical studies tool.

Displaying tags

You can display tag (flags and notes) information associated with a subject and a panel by turning on tag display.

To turn on tag display, in the Detail Data window, click **Tags**. The **Tags** button switches to **Tags ON**, and the Detail Data window displays tag information for any subject that you select until you turn off tag display:



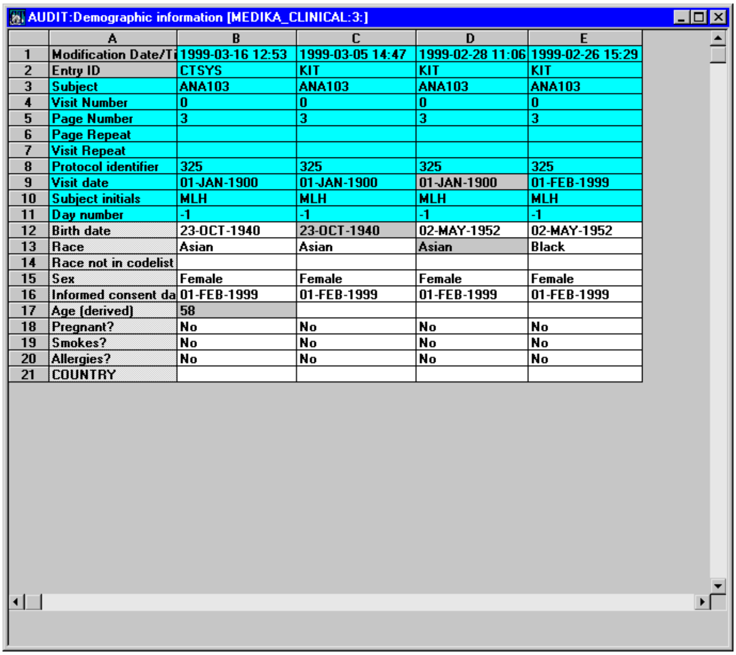
	A	B
1	Subject	ANA103
2	Visit Number	0
3	Page Number	3
4	Page Repeat	
5	Visit Repeat	
6	Protocol identifier	325
7	Visit date	01-JAN-1900
8	Subject initials	MLH
9	Day number	-1
10	Birth date	23-OCT-1940
11	Race	Asian
12	Race not in codelist	
13	Sex	Female
14	Informed consent da	01-FEB-1999
15	Age (derived)	58
16	Pregnant?	No
17	Smokes?	No
18	Allergies?	No
19	COUNTRY	

The Tags button switches to Tags ON.

Displaying the audit trail

You also can use the Data Browser to display the audit trail for a panel. The audit trail documents the dates on which an item was modified, and the value of the item before the modification.

To display the audit trail, select a column from the Detail Data window, then click **Audit Trail**. The Audit Trail window displays the data chronologically, with the original data in the rightmost column:



AUDIT:Demographic information [MEDIKA_CLINICAL:3:]

	A	B	C	D	E
1	Modification Date/Ti	1999-03-16 12:53	1999-03-05 14:47	1999-02-28 11:06	1999-02-26 15:29
2	Entry ID	CTSYS	KIT	KIT	KIT
3	Subject	ANA103	ANA103	ANA103	ANA103
4	Visit Number	0	0	0	0
5	Page Number	3	3	3	3
6	Page Repeat				
7	Visit Repeat				
8	Protocol identifier	325	325	325	325
9	Visit date	01-JAN-1900	01-JAN-1900	01-JAN-1900	01-FEB-1999
10	Subject initials	MLH	MLH	MLH	MLH
11	Day number	-1	-1	-1	-1
12	Birth date	23-OCT-1940	23-OCT-1940	02-MAY-1952	02-MAY-1952
13	Race	Asian	Asian	Asian	Black
14	Race not in codelist				
15	Sex	Female	Female	Female	Female
16	Informed consent da	01-FEB-1999	01-FEB-1999	01-FEB-1999	01-FEB-1999
17	Age (derived)	58			
18	Pregnant?	No	No	No	No
19	Smokes?	No	No	No	No
20	Allergies?	No	No	No	No
21	COUNTRY				

Printing detail data

To print the data from a particular subject’s Detail Data window, from the **File** menu, select **Print**. A dialog box opens, displaying a list of all open windows in the Data Browser.

You can print the data from as many windows as you want. Each window’s data prints on a separate sheet of paper.

For more information on printing query results, see "Printing from the Query Results window" on page 408.

29 Retrieve and Multisite

Queries in distributed protocols

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

- Distributing queries with protocols
- Creating queries in distributed protocols

Distributing queries with protocols

Public queries are distributed with a protocol. Private queries are not distributed with a protocol.

Creating queries in distributed protocols

If a protocol has been distributed, you can create a private query in the protocol at the Distribution Master site or any Distribution Subordinate site.

You can never create a public query at a Distribution Subordinate site. You can create a public query at the Distribution Master site only when the protocol is open for revision. You can view changes to a public query in the Comparison Details report.

If you want to create a new public query in a distributed protocol:

1. Open the protocol for revision at the Distribution Master site.
2. Create the new public query.
3. Close the protocol for revision.
4. Distribute the revision of the protocol to all Destination sites.
5. Accept the revision of the protocol at all Destination sites.

Clintrial software-supplied MedDRA queries

In a Multisite environment, if you use MedDRA, you need to distribute the CT_MEDDRA protocol to all participating replication sites, and also run the ct_meddra_util.create_meddra_functions at each site. When you execute the create_meddra_functions, you create functions used in Clintrial software-supplied Retrieve queries to return a higher level term code, as well as the code for the lower level term.

These functions can also be used in SELECT statements in SQL Tools or in derivations.

For more information on these functions, see the MedDRA-related functions in the *Reference Guide*.

30 *Working with Query Results*

Overview

Running a query

To retrieve data from the database, you run queries that you either have just created or that you or another user have saved to the Query Library.

When you run a query, Retrieve uses the query criteria to retrieve the corresponding data from the Oracle database tables. Once you have retrieved the data by running the query, you can print the results, or save them in a variety of formats.

You can run the query from the Query Library or from a query window. To run the query, from the **File** menu, select **Run Query**, or on the toolbar, click **Run**.

Specifying the results destination

Before you run the query, you must specify one of the following destinations for the query results:

- A window on your monitor

  If you are running a Query By Form, Query By Panel, or Query By SQL, this destination sends results to a Query Results window.

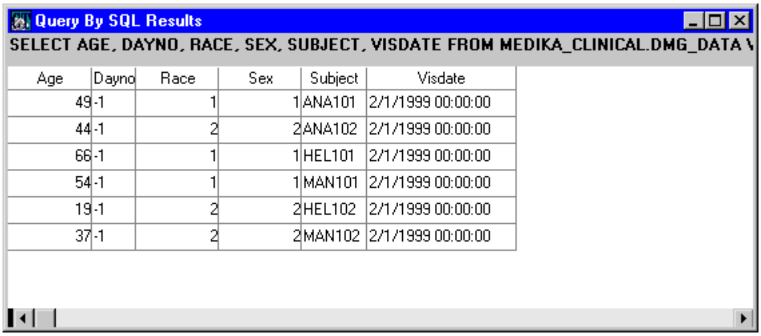
  If you are running an Ad Hoc Query, this destination sends results to a Data Browser window.
- An Oracle database table
- A SAS Data file

You select the destination from the **Results** menu. A check mark is displayed next to the selected destination. This destination remains selected for all queries run in the Retrieve session until you select a new destination. (When you begin a Retrieve session, the results destination defaults to **To Window**.)

Sending results to a Query Results window

You should direct your query results to a window on your monitor if you want to immediately display the results of your Query By Form, Query By Panel, or Query By SQL; print the query results; or, save the query results in other formats.

When the results destination for a Query By Form, Query By Panel, or Query By SQL is **To Window**, and you select **Run Query** from the **File** menu or click **Run** on the toolbar, a Query Results window opens, displaying the results of the query:



SELECT AGE, DAYNO, RACE, SEX, SUBJECT, VISDATE FROM MEDIKA_CLINICAL.DMG_DATA \

Age	Dayno	Race	Sex	Subject	Visdate
49	-1	1	1	ANA101	2/1/1999 00:00:00
44	-1	2	2	ANA102	2/1/1999 00:00:00
66	-1	1	1	HEL101	2/1/1999 00:00:00
54	-1	1	1	MAN101	2/1/1999 00:00:00
19	-1	2	2	HEL102	2/1/1999 00:00:00
37	-1	2	2	MAN102	2/1/1999 00:00:00

In the Query Results window, the query results are displayed in a table. You can customize the results table by resizing, filtering, and sorting the columns before printing or saving the results.

Resizing and moving the columns

When the Clintrial software sends the results to your monitor, each column appears in a default column size, determined by the length specified in the item definition. You can resize the columns by clicking on the column border, then dragging the column border to the size you want.

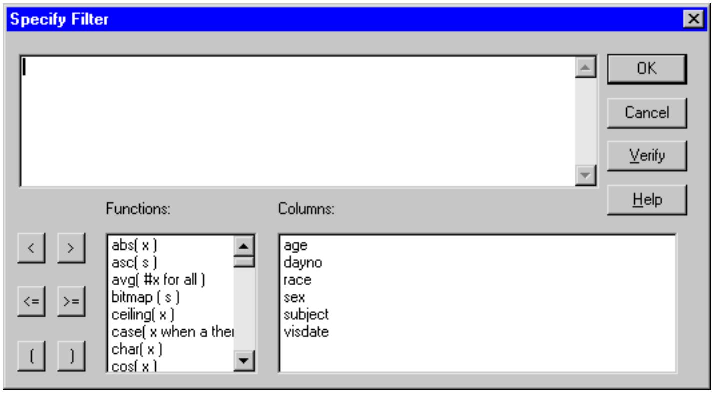
To change the order of the columns, click in the column header for an item and drag the column to a different location.

Filtering the query results

You can filter the results that appear in the results table to display different subsets of data.

Caution: Any filter that you specify applies only to the subset of results that are displayed on your monitor. However, when you save the results, all the results are saved.

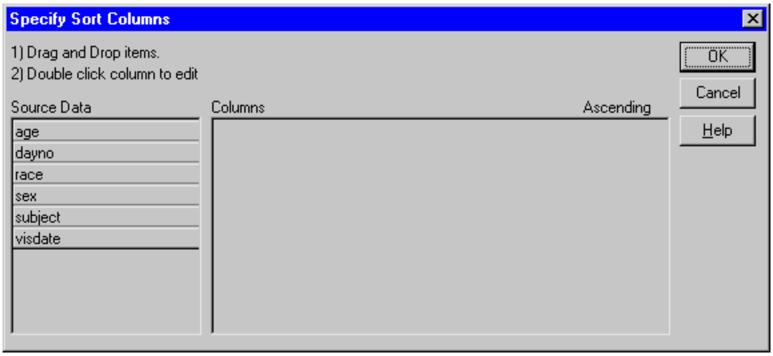
To filter the results, from the **View** menu, select **Filter**. The Specify Filter dialog box opens:



For more information on how to use the filter dialog box, see the Help.

Sorting the query results

You can sort the clinical data in ascending or descending order, by item. From the **View** menu, select **Sort**. The Specify Sort Columns dialog box opens:



Drag an item from the Source Data list to the Columns list. The first item you select is the primary sort item for all rows meeting the query criteria. If you select a second item, the data is sorted by the values in that item, grouped by the first sorting.

Caution: The sort order you specify is not in effect when you save the query results.

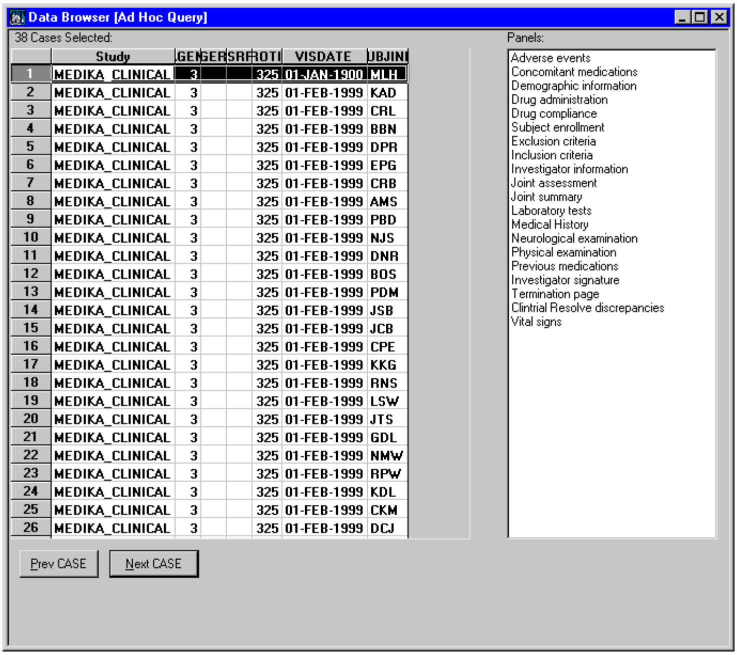
For more information on the sort order in a Query By Form, see "Selecting fields for retrieval" on page 339.

For information on specifying a sort order in a Query By SQL, see your SQL documentation.

Sending results to a Data Browser window

If you want to immediately display the results of an Ad Hoc Query, print the query results, or save the query results in other formats, you should direct the query results to a window on your monitor.

When the results destination for an Ad Hoc Query is **To Window**, and you select **Run Query** from the **File** menu or click **Run** on the toolbar, a Data Browser window opens, displaying records for subjects that meet the query criteria in a subject record list:



38 Cases Selected:

	Study	GEN	GER	SRF	ROTI	VISDATE	UBJINI
1	MEDIKA_CLINICAL	3			325	01-JAN-1900	MLH
2	MEDIKA_CLINICAL	3			325	01-FEB-1999	KAD
3	MEDIKA_CLINICAL	3			325	01-FEB-1999	CRL
4	MEDIKA_CLINICAL	3			325	01-FEB-1999	BBN
5	MEDIKA_CLINICAL	3			325	01-FEB-1999	DPR
6	MEDIKA_CLINICAL	3			325	01-FEB-1999	EPG
7	MEDIKA_CLINICAL	3			325	01-FEB-1999	CRB
8	MEDIKA_CLINICAL	3			325	01-FEB-1999	AMS
9	MEDIKA_CLINICAL	3			325	01-FEB-1999	PBD
10	MEDIKA_CLINICAL	3			325	01-FEB-1999	NJS
11	MEDIKA_CLINICAL	3			325	01-FEB-1999	DNR
12	MEDIKA_CLINICAL	3			325	01-FEB-1999	BOS
13	MEDIKA_CLINICAL	3			325	01-FEB-1999	PDM
14	MEDIKA_CLINICAL	3			325	01-FEB-1999	JSB
15	MEDIKA_CLINICAL	3			325	01-FEB-1999	JCB
16	MEDIKA_CLINICAL	3			325	01-FEB-1999	CPE
17	MEDIKA_CLINICAL	3			325	01-FEB-1999	KKG
18	MEDIKA_CLINICAL	3			325	01-FEB-1999	RNS
19	MEDIKA_CLINICAL	3			325	01-FEB-1999	LSW
20	MEDIKA_CLINICAL	3			325	01-FEB-1999	JTS
21	MEDIKA_CLINICAL	3			325	01-FEB-1999	GDL
22	MEDIKA_CLINICAL	3			325	01-FEB-1999	NMW
23	MEDIKA_CLINICAL	3			325	01-FEB-1999	RPW
24	MEDIKA_CLINICAL	3			325	01-FEB-1999	KDL
25	MEDIKA_CLINICAL	3			325	01-FEB-1999	CKM
26	MEDIKA_CLINICAL	3			325	01-FEB-1999	DCJ

Panels:

Adverse events
Concomitant medications
Demographic information
Drug administration
Drug compliance
Subject enrollment
Exclusion criteria
Inclusion criteria
Investigator information
Joint assessment
Joint summary
Laboratory tests
Medical History
Neurological examination
Physical examination
Previous medications
Investigator signature
Termination page
Clintrial Resolve discrepancies
Vital signs

The *subject record list* contains a spreadsheet of key clinical data for each subject, which is determined by your system administrator. You can resize the columns in the subject record list before saving or printing the results.

Resizing the columns

Each column in the subject record list appears in a default column size, determined by the length specified in the item definition. You can resize the columns by clicking on the column border, then dragging the column border to the size you want.

Changing the selection criteria

Each time you modify the selection criteria in the Selection Criteria window, then click **Update Browsers**, the Data Browser instantaneously provides a listing of the subject records that meets the new selection criteria.

Displaying detail data for a subject

You can display detail data for a subject by selecting a subject record, then selecting a panel. Retrieve opens a window that displays the values of the selected panel’s items for the selected subject.

For more information, see "Displaying detail data for a subject" on page 395.

Printing query results

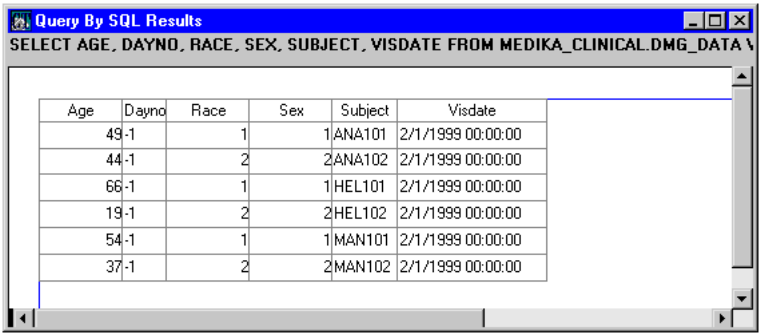
Printing from the Query Results window

After you have customized the display, you can print the query results from the Query Results window.

To specify the printer, from the **File** menu, select **Print Setup**.

To print the results, from the **File** menu, select **Print**. Then, in the Print Options dialog box, specify the print options and click **OK**.

Retrieve prints the query results on the specified printer, just as they appear in the table in the Query Results window:



SELECT AGE, DAYNO, RACE, SEX, SUBJECT, VISDATE FROM MEDIKA_CLINICAL.DMG_DATA \

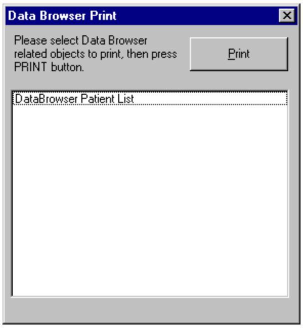
Age	Dayno	Race	Sex	Subject	Visdate
49	-1	1	1	ANA101	2/1/1999 00:00:00
44	-1	2	2	ANA102	2/1/1999 00:00:00
66	-1	1	1	HEL101	2/1/1999 00:00:00
19	-1	2	2	HEL102	2/1/1999 00:00:00
54	-1	1	1	MAN101	2/1/1999 00:00:00
37	-1	2	2	MAN102	2/1/1999 00:00:00

Printing from the Data Browser window

You can print the subject record list and detail data for the list of subjects from the Data Browser window.

To specify the printer, from the **File** menu, select **Print Setup**.

To print the results, from the **File** menu, select **Print**. The Data Browser Print dialog box opens:



Please select Data Browser related objects to print, then press PRINT button.

DataBrowser Patient List

Select the panel whose data you want to print, then click **Print**. Retrieve prints the subject record list on one page, and the items from the selected panels for all subjects in the subject record list on another page.

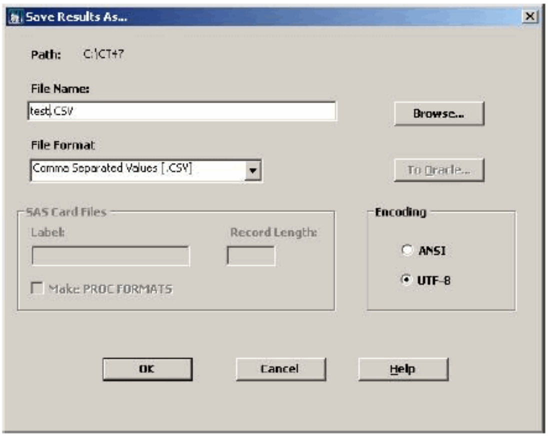
Saving query results

Saving results to a file

You can save the query results in a variety of formats, then use these results in other applications to analyze the data or create reports.

To save the results, from the **File** menu, select **Save Results As**, or on the toolbar, click **Save**.

- If the query results are in a Data Browser window, the Results Panel Selection dialog box opens. Select the panel for which you want to save data, then click **OK**, which opens the Save Results As dialog box.
- If the query results are in a Query Results window, the Save Results As dialog box opens:



Save Results As...

Path: C:\CT47

File Name: test.CSV

File Format: Comma Separated Values (.CSV)

SAS Card Files — Label: Record Length: □ Make PROC FORMATS

Encoding: ○ ANSI ● UTF-8

In this dialog box, you specify the file name and file format you want. If you are saving the results to an Oracle database table or to a SAS Data file, you also specify additional information.

You can save CSV file outputs to either a Unicode or a non-Unicode format on the Japanese operating system. Under Encoding, the are radio buttons to allow the choice of UTF-8 (Unicode) or ANSI (non-Unicode) output to SAS. The default value is UTF-8. The ANSI option is available to the Japanese operating system only, and is grayed out in the English operating system. The phrase ENCODING='ANSI' is added at the end of the INFILE statement when the ANSI button is selected.

⚠ ***Caution:*** If you send the query results to the To Window destination, be sure that all records are returned before you save the results. You can save the results before all the records are displayed, in which case the file may not contain all the records that are returned.

The sort order you specify is not in effect when you save the query results.

The file name you specify automatically uses the three-character extension corresponding to the file format you select. The file is stored by default on your C drive. You can change this default location by clicking **Browse**, then selecting the location you want.

⚠ ***Caution:*** If you override the default file extension, the Clintrial software still saves the results in the specified format. You cannot change the format by changing the file extension.

The following table lists the formats supported by Retrieve. You can also save the results to an Oracle database table.

File Extension:	**File Format:**
.CSV	Comma separated values
.DBF	dBase2 or dBase3
.DIF	Data Interchange Format
.XLS	Microsoft Excel Format
.PSR	InfoMaker Report
.SAS	SAS Data file ***Note***: When you choose this file format, you also need to specify additional SAS Data file information. For more information, see "Saving results to a SAS Data file" on page 413.
.SQL	SQL Syntax
.TXT	Text file
.WMF	Windows Meta File

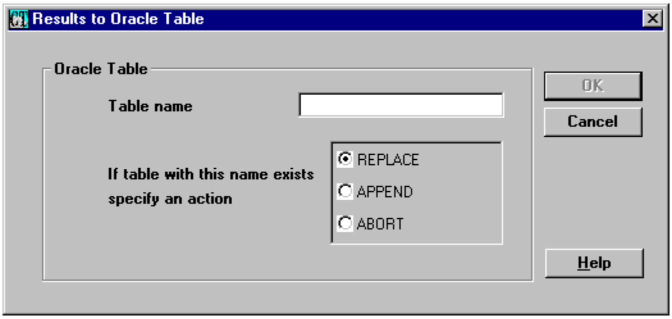
Saving results to an Oracle database table

You can send the results of a query directly to an Oracle database table as follows:

1. From the **Results** menu, select **To Oracle Table**.

   Note: To access the Oracle database table you create from within the Clintrial software, you can use an SQL statement in Query By SQL to retrieve records from the table.

2. When the results destination is **To Oracle Table**, and you select **Run Query** from the **File** menu or click **Run** on the toolbar, the Results to Oracle Table dialog box opens:



If table with this name exists specify an action

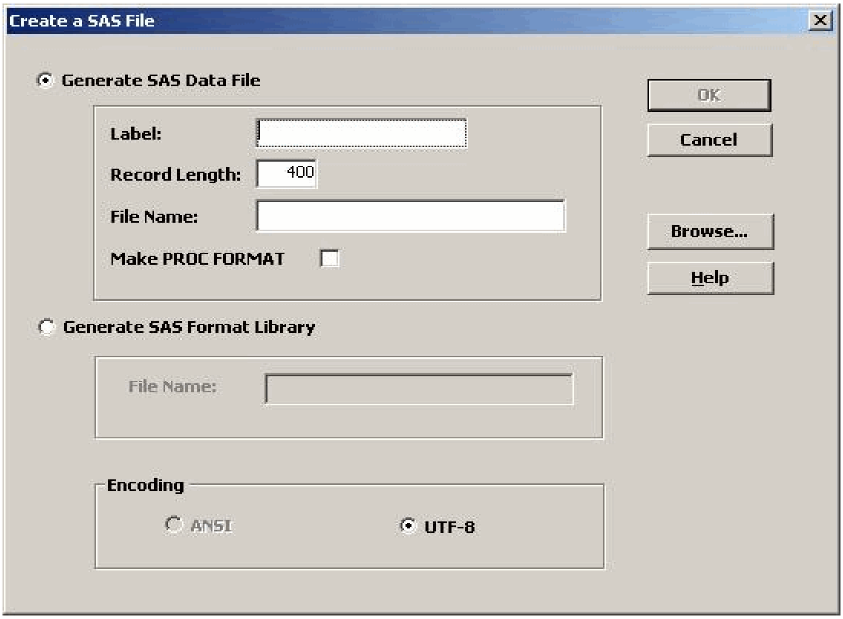
● REPLACE
○ APPEND
○ ABORT

In this dialog box, you specify the table name, and the action Retrieve should take if a table with this name already exists in your user account directory. Retrieve stores the Oracle database table in your Oracle user account using your default tablespace.

Saving results to a SAS Data file

You can send the results of a query directly to a SAS Data file, so that you can use SAS to analyze the retrieved data:

1. From the **Results** menu, select **To SAS Data file**.
2. When the results destination is **To SAS Data file**, and you select **Run Query** from the **File** menu or click **Run** on the toolbar, the Create a SAS File dialog box opens:



Create a SAS File

● Generate SAS Data File

Label:

Record Length: 400

File Name:

Make PROC FORMAT □

○ Generate SAS Format Library

File Name:

Encoding

○ ANSI ● UTF-8

In this dialog box, you specify the label, record length, and file name for the data file.

The user preference RTV_SAS_RECLEN determines the default record length for SAS Data file retrievals. You can override this default by entering a new value in the **Record Length** field. If you choose the ANSI option in the **Encoding** field, the length for items with the VARCHAR2 datatype will be doubled. For the UTF-8 option, the length will be five times the original item length. The increase of the length is for Unicode databases when the NLS_LENGTH_SEMANTICS is CHAR. The **Record Length** field may be increased if appropriate.

Retrieve uses the file name you specify to create two SAS files with different extensions: an ASCII raw data file (file extension .dat) if the ANSI option is selected, and a command file (file extension .sas). For example, if you specify ENROLL as the file name, Retrieve creates the files ENROLL.DAT and ENROLL.SAS.

The SAS files are stored by default in your Clintrial software directory on your client computer. You can store them in a different directory, or to a network, by clicking **Browse**, and then selecting a new drive and directory.

In addition to creating SAS command and data files, you can create proc formats to use with the data files in the SAS application. You can create two types of proc formats:

- Proc format specific to the data file retrieval

  If you want to create a proc format specific to the data file you are creating, you check **Make PROC FORMAT**. The format file is stored in the same directory as the data file, using the file name you specified and the file extension .FMT (for example, ENROLL.FMT). This file is stored with the other two SAS files.

  Caution: If the proc format file has any statements in addition to the PROC FORMAT statement, it lists the codelists that are referred to by the exported items. However, each codelist name is appended with a letter, indicating how the code is being decoded:

  – V — Decodes with the value in the Value column.

  – S — Decodes with the short label.

  – L — Decodes with the long label (up to 40 characters).

  You must edit the file to remove the V, S, or L, so that the format in SAS has the same name as in the Clintrial software database.

- Format library for the entire database

  If you want to create a format library for the entire database, you select **Create Format Library**, then specify the directory and file name for the format library.

  The user preference SAS_LIBNAME determines the default file the Clintrial software uses for storing the format library file. You can change this file at the time you create the format library. Retrieve adds the file extension .FMT to the file name you specify when it creates the file.

For more information on user preferences, see "Setting your user preferences" on page 315.

In the Japanese operating system, you can set the **Encoding** for the SAS output file to either ANSI or UTF-8 at the time you create the format library. ANSI is disabled in English operating systems.

Click on the encoding method you would like the SAS file saved in. Select either ANSI or UTF-8.

If data are in multiple languages, such as in a combination of Japanese, Chinese and English, then the UTF-8 option will not work correctly. You must edit the SAS file directly to use variable lengths and delimiters. The data file may use comma separated output first, then be changed globally to a desired delimiter.

Index

Clintrial 4.7.1
Enter, Resolve, and Retrieve
Part Number: E27575-01