

Regis University

ePublications at Regis University

Regis University Student Publications
(comprehensive collection)

Regis University Student Publications

Fall 2006

Clinical Data Entry & Protocol Tracking System

Neeta Pophali
Regis University

Follow this and additional works at: <https://epublications.regis.edu/theses>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Pophali, Neeta, "Clinical Data Entry & Protocol Tracking System" (2006). *Regis University Student Publications (comprehensive collection)*. 755.
<https://epublications.regis.edu/theses/755>

This Thesis - Open Access is brought to you for free and open access by the Regis University Student Publications at ePublications at Regis University. It has been accepted for inclusion in Regis University Student Publications (comprehensive collection) by an authorized administrator of ePublications at Regis University. For more information, please contact epublications@regis.edu.

Regis University
School for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Use of the materials available in the Regis University Thesis Collection ("Collection") is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the "fair use" standards of the U.S. copyright laws and regulations.

Regis University
School for Professional Studies Graduate Programs
MSCIT Program
Graduate Programs Final Project/Thesis
Certification of Authorship of Professional Project Work

Print Student's Name NEETA POPHALI

Telephone 781 752 0106 Email popha961@regis.edu

Date of Submission November 04, 2006 Degree Program MSCIT

Title of Submission CLINICAL DATA ENTRY & PROTOCOL TRACKING SYSTEM

Advisor/Faculty Name Donald Archer

Certification of Authorship:

I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the Master of Science in Computer Information Technology Degree Program.

Neeta Pophali nsppophali 11/14/2006

Student Signature

Date

Regis University
 School for Professional Studies Graduate Programs
 MSCIT Program
 Graduate Programs Final Project/Thesis
Advisor/Professional Project Faculty Approval Form

Student's Name: Neeta Pophali Program MSCIT

PLEASE PRINT

Professional Project Title: Clinical Data Entry & Protocol Tracking System

PLEASE PRINT

Advisor Name Donald E. Archer

PLEASE PRINT

Project Faculty Name Donald E. Archer

PLEASE PRINT

Advisor/Faculty Declaration:

I have advised this student through the Professional Project Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MSCIT Degree Program.

Project Advisor Approval:

Donald E. Archer

Digitally signed by Donald E. Archer
 DN: CN = Donald E. Archer, O = Regis University, OU =
 Master of Science in Computer Information Systems
 Date: 2006.11.13 14:43:20 -0700

November 13, 2006

Original Signature

Date

Degree Chair Approval if:

The student has received project approval from Faculty and has followed due process in the completion of the project and subsequent documentation.

Donald E. Archer

Digitally signed by Donald E. Archer
 DN: CN = Donald E. Archer, C = US, O = Regis University,
 OU = Master of Science in Computer Information Systems
 Date: 2006.11.13 14:43:48 -0700

November 13, 2006

Original Degree Chair/Designee Signature

Date

Regis University

School for Professional Studies Graduate Programs

MSCIT Program

Graduate Programs Final Project/Thesis

Authorization to Publish Student Work

I, NEETA POPHALI, the undersigned student, in the Master of Science in Computer Information Technology Degree Program hereby authorize Regis University to publish through a Regis University owned and maintained web server, the document described below ("Work"). I acknowledge and understand that the Work will be freely available to all users of the World Wide Web under the condition that it can only be used for legitimate, non-commercial academic research and study. I understand that this restriction on use will be contained in a header note on the Regis University web site but will not be otherwise policed or enforced. I understand and acknowledge that under the Family Educational Rights and Privacy Act I have no obligation to release the Work to any party for any purpose. I am authorizing the release of the Work as a voluntary act without any coercion or restraint. On behalf of myself, my heirs, personal representatives and beneficiaries, I do hereby release Regis University, its officers, employees and agents from any claims, causes, causes of action, law suits, claims for injury, defamation, or other damage to me or my family arising out of or resulting from good faith compliance with the provisions of this authorization. This authorization shall be valid and in force until rescinded in writing.

Print Title of Document(s) to be published: CLINICAL DATA ENTRY & PROTOCOL TRACKING SYSTEM

Neeta Pophali neetapophali
Student Signature

11/14/2006
Date

Check if applicable:

 The Work contains private or proprietary information of the following parties and their attached permission is required as well: _____

Authorized Personnel

Name of Organization and/or

CLINICAL DATA ENTRY & PROTOCOL TRACKING SYSTEM

CLINICAL DATA ENTRY & PROTOCOL TRACKING SYSTEM (CDEPT)

Neeta Pophali

Regis University

School for Professional Studies

Master of Science in Computer Information Technology

Regis University
School for Professional Studies Graduate Programs
MSCIT Program
Graduate Programs Final Project/Thesis
Certification of Authorship of Professional Project Work

Print Student's Name NEETA POPHALI

Telephone 781 752 0106 Email popha961@regis.edu

Date of Submission November 04 , 2006 Degree Program MSCIT

Title of Submission CLINICAL DATA ENTRY & PROTOCOL TRACKING SYSTEM

Advisor/Faculty Name Donald Archer

Certification of Authorship:

I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the Master of Science in Computer Information Technology Degree Program.

Neeta Pophali

11/14/2006

Student Signature

Date

Regis University
School for Professional Studies Graduate Programs
MSCIT Program
Graduate Programs Final Project/Thesis
Authorization to Publish Student Work

I, **NEETA POPHALI**, the undersigned student, in the Master of Science in Computer Information Technology Degree Program hereby authorize Regis University to publish through a Regis University owned and maintained web server, the document described below ("Work"). I acknowledge and understand that the Work will be freely available to all users of the World Wide Web under the condition that it can only be used for legitimate, non-commercial academic research and study. I understand that this restriction on use will be contained in a header note on the Regis University web site but will not be otherwise policed or enforced. I understand and acknowledge that under the Family Educational Rights and Privacy Act I have no obligation to release the Work to any party for any purpose. I am authorizing the release of the Work as a voluntary act without any coercion or restraint. On behalf of myself, my heirs, personal representatives and beneficiaries, I do hereby release Regis University, its officers, employees and agents from any claims, causes, causes of action, law suits, claims for injury, defamation, or other damage to me or my family arising out of or resulting from good faith compliance with the provisions of this authorization. This authorization shall be valid and in force until rescinded in writing.

Print Title of Document(s) to be published: **CLINICAL DATA ENTRY & PROTOCOL TRACKING SYSTEM**

Neeta Pophali
Student Signature

11/14/2006
Date

Check if applicable:

 The Work contains private or proprietary information of the following parties and their attached permission is required as well: _____

Authorized Personnel

Name of Organization and/or

Regis University
School for Professional Studies Graduate Programs
MSCIT Program
Graduate Programs Final Project/Thesis
Advisor/Professional Project Faculty Approval Form

Student's Name: Neeta Pophali Program MSCIT

PLEASE PRINT

Professional Project Title: Clinical Data Entry & Protocol Tracking System

PLEASE PRINT

Advisor Name Donald E. Archer

PLEASE PRINT

Project Faculty Name Donald E. Archer

PLEASE PRINT

Advisor/Faculty Declaration:

I have advised this student through the Professional Project Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MSCIT Degree Program.

Project Advisor Approval:

Original Signature

November 13, 2006

Date

Degree Chair Approval if:

The student has received project approval from Faculty and has followed due process in the completion of the project and subsequent documentation.

Original Degree Chair/Designee Signature

November 13, 2006

Date

Abstract

CLINICAL DATA ENTRY & PROTOCOL TRACKING SYSTEM (CDEPT)

is a software framework designed to provide the tools necessary to rapidly develop web based data entry and data management systems for clinical trails and medical research studies.

A software framework defines a model, approach, procedures and tools for creating new protocols for clinical studies. As a framework this software is able to gain efficiency by providing standard approaches and tools for commonly needed capabilities such as construction of data entry routines, validation of data, audit trails and monitoring the completeness and timeliness of data collection. However, as a framework this software provides far greater flexibility, expandability and customization than is typical in a turnkey or off the shelf application.

CDEPT is designed around a three-tiered software development model widely used on Internet. This model consists of browsers, web server and database servers. This software is mostly designed to run on two major browsers Internet Explorer and Netscape.

Table of Contents

Chapters

| | |
|---|----|
| Chapter 1: Introduction / Executive Summary | 7 |
| 1.1 Statement of the Problem to be Investigate and Goal to be achieved..... | 7 |
| 1.2 Relevance, significance or need for the project | 7 |
| 1.3 Barriers and/or issues | 8 |
| 1.4 Limitations/scope of the project..... | 9 |
| 1.5 Definition of terms..... | 10 |
| Chapter 2: Review of Literature / Research..... | 12 |
| 2.1 Literature and research that is specific/relevant to the project | 12 |
| 2.2 The contribution this project will make to the field..... | 12 |
| Chapter 3: Methodology | 13 |
| 3.1 Life-cycle models to be followed | 13 |
| 3.2 Resource requirements..... | 13 |
| 3.3 Formats for presenting results / deliverables | 15 |
| 3.4 Specific procedures and Review of deliverables | 15 |
| 3.4.1 Define initial Study Meta Data | 15 |
| 3.4.2 Define data entry routines for study forms | 44 |
| 3.4.3 Use Auto DBA tool to create Oracle tables | 46 |
| 3.4.4 Format HTML data entry screens to match paper forms | 46 |
| 3.4.5 Test the data entry routines to make sure everything works correct..... | 48 |
| 3.4.6 Define custom features on standard CDEPT pages. | 48 |
| Chapter 4: Project History..... | 52 |
| 4.1 How the project began. | 52 |
| 4.1.1 My role in project development..... | 53 |
| 4.2 How the project was managed. | 54 |
| 4.3 Significant events / Milestones in the project | 54 |
| 4.4 Changes to the project plan..... | 55 |
| 4.5 Evaluation of whether or not the project met project goals | 55 |
| 4.6 Discussion of what went right and what went wrong in the project..... | 56 |
| 4.7 Discussion of project variables and their impact on the project | 56 |
| Chapter 5: Lessons Learned and Next evolution of the project..... | 58 |
| 5.1 Lessons learned from the project experience..... | 58 |
| 5.2 Discussion of weather or not the project met initial project expectations | 58 |
| 5.3 Next stage of evolution for the project would be if it continued | 58 |
| Works Cited | 59 |

Chapter 1: Introduction / Executive Summary

1.1 Statement of the Problem to be Investigate and Goal to be achieved

“Finding Reality” is a clinical research organization which conducts medical studies and trails in all parts of the world. “Finding Reality” collects data for medical research in various fields or contracts with private drug companies to conduct clinical trails to prove their drug or finding. Usually any medical study can be done in one or multiple locations and are bound to certain rules and regulations as to what extent they can capture people’s data. Every study has its predefined "study protocol" which is approved when the study is funded. This "study protocol" has the details of the study purpose, eligibility criteria for the people participating in the study, study duration and locations. Every study is carried out as a series of actions that occur in a particular time or interval. Most of the medical studies are monitored by standard health care institutes or government to protect participant rights.

People are recruited in the study based on the protocol requirements and asked to perform study evaluation or medical tests depending on the nature of the study. There can be multiple sources of research data for a single study.

- It can be on paper forms filled by the study participants at various study sites which can be either a hospital, doctor’s office, or study specific office.
- It can be results of some medical tests where data is downloaded from medical equipment (e.g. such as echocardiogram results)
- It can be on hand held devices like digital diary where people store their daily study information.

1.2 Relevance, significance or need for the project

“Finding Reality” is a medical research organization and conducts various medical studies at one time. All studies have some common elements in it as well as have custom needs specific to the protocol. Some of the common abstractions for any medical study are,

- People, places or things are studied
- The timing and location of interactions between study staff and the people, places or things are studied.
- The specific sets of data that are collected during the above interactions
- Protocol rules linking the above three things together

When any medical study is approved “Finding Reality” needs to create a new website accessible to all study participants and people managing study to enter the associated study data. Instead of investing in programming a new website for all future studies, the company needs to develop a generic website which can be used as the start of any new medical study in the company. The company requires a set of software tools for rapidly developing customized web-based data management systems which can easily implement the above abstractions.

The study data resides on paper forms or electronic lab instruments. The primary need of this software is to have a centralized database where data from all the different study sources can be stored so that it can be easily accessed by the research scientists and statisticians to prove the study purpose. Appropriate database design and development are fundamental to a successful research program. This software should be able to define, validate and save any study data as per the protocol rules. This software should be easy to use for any protocol but could also be customized for any protocol specific use.

1.3 Barriers and/or issues

For most of the medical research studies it is required that the personal identification data such as the patients name and address should be hidden from the project reports and not stored in any electronic database. This data is collected for patient contact use but stays in

the paper form only. It is needed that the patient should still be identified in the study not by its name but some other special character that can be used internally in the study.

1.4 Limitations/scope of the project

“Finding Reality” names this website as “CLINICAL DATA ENTRY & PROTOCOL TRACKING SYSTEM” often known as (CDEPT).

CDEPT Scope and Requirements:

- Develop generic website which can be easily implemented for any medical study
- This website should allow the user to easily define the protocol rules, individual actions in the protocol and their timings.
- The website should allow easy way to set up validation rules to monitor the integrity of the data
- The website should have an efficient data entry system to enter study data on daily basis.
- The website should have a user friendly interface to see all the activities for a given study participant
- The website should have easy way to add/ change menus within the existing website
- There should be a way to add custom programs designed for protocol specific use
- The database design of this software should be strong enough and provide appropriate auditing features for all the data stored
- The website should provide a standard set of reports to view the data stored
- The website needs to be secured enough to avoid inappropriate people from accessing it. The website should be password protected
- There should be a easy way to maintain all the users of the website

1.5 Definition of terms

“Finding Reality” uses the following terminology for defining items in CDEPT

1) Data perspective – It defines the people, places, or things to be studied. Mostly the company studies people but sometimes it can study school, towns, hospital or families. Case definition and unit of study are both terms sometimes used to express this concept.

2) Event / Event types – Event defines the timing and location of interactions between study staff and the people, places or things are studied. One or more kinds of study events are clinic visits. However other kinds of interactions can also be study events, such as telephone interviews, hospitalizations, and annual censuses of town. Events might be defined in terms of very discrete point in time like exact time of participant’s withdrawal from study or can be more general such as participants baseline visits data collection – which might actually span over a number or related visits.

3) Dated events – The events that occur on the specific dates and time are called dated events. For example the 5-week visit after hospitalization will happen on the specific date and these events need to be triggered during that time only.

4) Event Categories- Event categories allow events to be grouped together based on some common features. Event categories are then used within the CDEPT system to control what events are displayed on the screen for a given data perspective member. For example, a study might have study visits where data is collected on an annual basis. During these visits data might be collected about subject hospitalizations. If the study protocol calls for the collection of extensive hospital record abstractions, then it might be useful to create an Annual Visit event category (which groups the annual follow-up visits together) and a hospitalization event category to group hospitalization events together.

5) Form - Within CDEPT study, forms often correspond to physical paper forms such as interviews, physical exams forms etc. The vast majority of study data is collected through the development of study forms. However, forms can also correspond to laboratory data electronically transferred to the DMS, phone interviews, or administrative functions such as enrollment, deactivation, randomization etc.

6) Form Version - CDEPT supports multiple versions of the same study form. When there are changes to the form such as more fields are added/subtracted or the data type of any fields are changed a new form version can be created. Form versions are assigned dates and corresponding single alphabetic codes.

7) Study Protocol Triggers- The study protocols rules link the data perspective, event and form altogether at the database side. The CDEPT protocol triggers create a study specific oracle trigger based upon the triggering condition.

8) master view – This is an Oracle view which gets automatically created with the creation of the each data perspective. The variables in this view can be referenced in the client side of CDEPT code very easily by having a prefix of a. in front of the variable name.

9) instrument definition – a tool used in CDEPT to create the field structure in the form. This tool involves the screen where the data type, length, order and special comments about the field in the form can be created.

10) custStudyRef.js – this is a JavaScript file which gets created automatically by the CDEPT code for each individual study. This file decides the menu structures and form structure for the project and has to be updated if any of these changes needs to be shown on the study page.

Chapter 2: Review of Literature / Research

2.1 Literature and research that is specific/relevant to the project

Most of the information written in this thesis about the CDEPT website was found in the ADEPT manual produced by “Finding reality”. The other source of research was the Internet where some help about the Active Server Pages and JavaScript commands was used. The source of research for the customized project of CDEPT called Ventricular Volume Variability, which I worked on most of the time was the specification provided by the project director and data manager. This specification was prepared by the study staff as per the study rules and regulations.

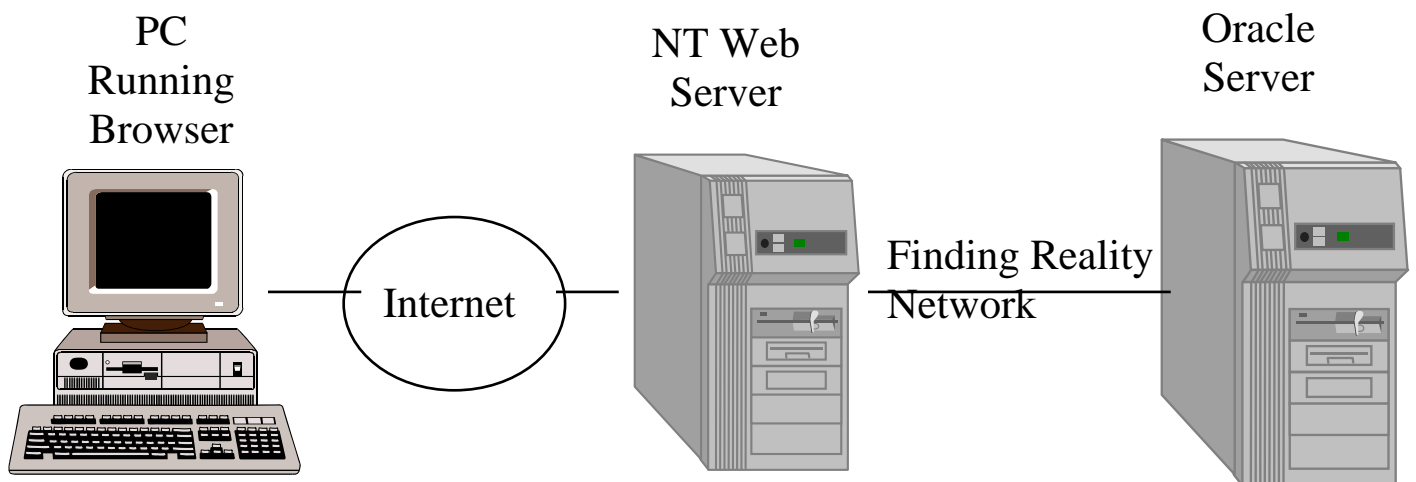
2.2 The contribution this project will make to the field

CDEPT structure is defined such as it can be used to store data for any medical research study. The customization of CDEPT for individual study use is very simple. This will be very helpful for the small medical research companies seeking for electronic data storage but do not want to invest time and money in creating the software. The use of this software is also easy which in turn increases customer satisfaction and need of less training.

Chapter 3: Methodology

3.1 Life-cycle models to be followed

CDEPT is designed around a three tiered software development model widely used on the Internet. This model consists of browsers, web servers and database servers. This approach consists of the client, or user interface, also known as the presentation layer; the data storage on the server end, or the database layer; and a middle layer that performs the processing and business logic that are the more time-consuming and intensive operations required during client-server interaction. The architecture separates presentation, processing, and data into distinct entities. The middle tier will normally use well-defined protocols to interact with the client and data server ends. (CDEPT manual)



3.2 Resource requirements

The following sections outline the system and software used by CDEPT.

Browsers

CDEPT is designed to run on the more recent versions of the two major browsers. Currently CDEPT requires version 4.5 or higher of Netscape's browser, or version 4.0 or higher of

Microsoft's Internet Explorer Browser. CDEPT requires these newer versions because of extensive use of JavaScript to enhance the performance the system and to control data entry of study forms. In addition, in order to work properly, the browsers must be configured correctly. Browsers must:

- Have JavaScript enabled
- Have cookies enabled
- Have caching set to check "every time" for updated files

Web Servers:

CDEPT uses a web server to manage communication between the user (using a browser) and data in the database. CDEPT uses Microsoft's Internet Information Server 4.0 web server software. Extensive use is made of Microsoft's server side scripting abilities. The majority of scripting is written in JScript – with a small amount of VB script. In addition other software including ActiveX and com objects are used on the server.

The following software is currently run on the web server:

- ABMailer – for E-mail generation
- Graphics Servers – for server-side graphics generation
- Active Data Objects ADO – for database communications
- Verandi Randomization software
- ODBC Drivers to access Oracle

Each study may have multiple web applications – one for development, one for production, and possibly a third used for demonstration purposes.

Oracle Servers:

CDEPT currently uses Oracle's relational database software version 8 to store study data.

The Oracle server runs on an NT 4.0 server. All routine access to the Oracle tables is performed via the Web server

3.3 Formats for presenting results / deliverables

The results of the project are presented in the following section "3.4 Specific procedures and Review of deliverables" in the form of screenshots from the user interface.

3.4 Specific procedures and Review of deliverables

When a new research study is funded the programmer assigned for the study meets with the project staff to understand the study protocol. The programmer then creates the customized CDEPT website by performing the following steps.

3.4.1 Define initial Study Meta Data

The study meta data is defined by defining the data perspective, events, forms, form version, protocol triggers within the study. The detail steps of defining the meta data is as follows.

These steps are written as described in the 'CDEPT manual'.

3.4.1.a Defining Data Perspectives:

Every study must define at least one data perspective. Many studies only need one data perspective. To define data perspectives use the Data Perspective menu under the Study Definition Menu.

Maintain Data Perspectives

Perspective Descriptions: Participant
Participant
---New---

Code: PART

Description: Participant

Length of ID #: 7

Check digit: ☒

Default Event Category: CORE

Allow initialization of new IDs: ☐

Update Client and Table Cancel

Figure - 3.4.1.A Data Perspective Maintenance

Use the pull down combo-box to either select an existing data perspective for editing, or select “---new---” to add a new perspective (see figure 3.4.1.A). Each of the fields that define the data perspective are explained below.

Code:

This is a 4-character code used to identify uniquely the data perspective. It is used internally by the system. This code is also appended to the string “MAST_” to define the Oracle table name that will hold one record for each person, place or thing studied in this data perspective. This code must be 4 characters long and it must be unique. All characters should be in upper case.

Description:

This 100-character string is used to store a description for what the data perspective represents. This description should define the people place or thing being studied. This should use a description that will be meaningful in the context of the study. This description will be used in many reports and displays within CDEPT.

Length of ID:

This field defines the length of the ID field used for the data perspective. It is required that all data perspectives will utilize a system of unique IDs.

Check digit:

This checkbox is used to indicate if the ID system used for this data perspective is going to use a check digit. The check-digit treats the last digit of the ID (and it must be a number) as a special digit to check that the ID is a valid id. When you define the length of the ID in the prior field you should take into account if you plan to use a check-digit. For example if you enter 5 in the Length of ID field and you check the check digit checkbox, the system will treat the fifth digit of the id as the check digit. It will not automatically append another digit to the length of ID field.

Check-digits are calculated using an algorithm that is based on the other digits in the ID. It is designed to reduce transpositional keystroke errors. If a user keys in two digits in the wrong order there is a very high probability that the check digit will not be correct – and hence the error will be detected.

Default Event Category:

This field holds a 4-character code that corresponds to an event category. This will define the category of events that will be displayed when you look up a new ID from this data perspective. Because you define data perspectives prior to event categories there is no

checking that the code you enter is valid - it's up to you to make sure any code you use here ultimately defines a valid event category. If a study only has one event category, then that code should be put here. All codes should be in upper case.

Allow initialization of new Ids:

This check box defines whether or not users of the system are allowed to initialize new ID numbers from within the data management system. Some studies will load IDs from an external source - from a randomization system, or perhaps bulk loaded from some other database at the start of the study. Others will recruit new subjects one at a time and will need to be allowed to add new subjects when they are recruited. Check this checkbox if the system should allow new IDs to be initialized.

Saving Changes:

After completing all the fields to define the data perspective click on Update Client and Table. For changes made in data perspectives to become active, run Update StudyRef.js under the Development Menu option.

Data Perspective definitions - Oracle and client side:

The data that define the data perspectives is stored in a single table in Oracle called Perspect.

3.4.1.b Defining Event Types:

Data perspectives and event categories must have been defined prior to defining event types.

To maintain event types select Event Types menu item from the Study Definition menu. Use the pull down combo-box to either select an existing event type for editing edit, or select “---new---” to add a new event type (see figure– 3.4.1.b1 & 3.4.1.b2). Each of the fields that define the event type are explained below.

Code:

This is a 4-character code used to identify uniquely the event type. It is used internally by the system. This code must be 4 characters long and it must be unique. All characters should be in upper case.

Description:

This 100-character string is used to store a description for what the event type represents. This description should define a specific time at which study data is generally collected. This should use a description that will be meaningful in the context of the study. For example, Baseline visit, 1st Annual Follow-up, Phone Interview, and Medical Record Abstraction.

Associated with which Event Type:

This pull-down combo-box displays a list of data perspectives. Each event type must be associated with one and only one event category. Select the appropriate event category from the pull down list.

Event Forms:

This field defines the relationship between the event and study forms. A study event has three different potential relationships with study forms. An event is typically associated with several different study forms. If under any circumstance it is possible for the protocol to require more than one study form for the event, and then select "Multiple forms". If the study event is always associated with one and only one study form, select "Singleton". Lastly, if the study event is never associated with the collection of a study form select "None".

ADEPT - Foss Tighe - Maintain Event Types - Netscape

Maintain Event Types

Event Type: Introductory Letter Mailed

Event Type ID: MAIL

Description: Introductory Letter Mailed

Associated with which Event Category:

Physician Data

Event Forms: No forms

Dated Event: ☒

Event Has Windows: ☒

Enforce Windows: ☐

Display Order: 1

Start Window Snippet:

a.mail_date+7

End Window Snippet:

a.mail_date+21

Figure 3.4.1.b1: Event Type Maintenance Screen

ADEPT - Foss Tighe - Maintain Event Types - Netscape

HTML Event Display Snippet:

```
<td width=200><FONT COLOR=#CDCDCD SIZE=2
TYPE=arial>Description:</font><br>" + g.eventtypeList[eventtype_id].descript + "</td>
<td colspan=2><FONT COLOR=#CDCDCD SIZE=2 TYPE=arial>Letter
Mailed:</font><br>" + event_date + " </td>"
```

Update Client and Table Cancel

Figure 3.4.1.b2: Event Type Maintenance Screen

Dated Event:

This checkbox indicates if initiation the event requires recording a specific date. Most study events are associated with a specific date. In a few circumstances it does not make sense to require the recording of a date with the event. A dated event will prompt the user for a date when the event is initialized (selected for the first time). If the event is also a windowed event, system will validate that the date entered falls between the window start and window end dates.

Event has Windows:

This field defines whether the event has an event window. The event window serves as a constraint on valid event dates, and it can also be used in scheduling and reporting systems. The actual logic of how the event window is defined is stored elsewhere.

Enforce Windows:

This field should only be used with Windowed and dated events. This field controls whether or not the system requires that the event date fall within the event windows. In most CDEPT systems the event date is entered by the user when the event is initiated. The system prompts the user for a date. If the Enforce Windows option is checked then the system will completely refuse to accept a date that does not fall within the window. If this option is not checked, the system will warn the user if the date does not fall within the windows but it will allow the user to proceed anyway.

Display Order:

This field controls the order in which lists of events are displayed in the event summary screen. Any numeric value up to three digits can be entered. Display order is relative to the

display order of other event types. It is also used to sort event types in some other CDEPT reports.

Start Window Snippet:

This field is with windowed event types and protocol triggers to define how event windows are calculated when events are added. If used this snippet needs to define an Oracle date field within the context of a select statement. The snippet can use any internal Oracle functions and can reference any fields in the master view. Fields from the master view must be prefixed with "a". For example if a study had a field in the master view (MAST_PATI) called rand_date that corresponded to the date a subject was randomized. The following snippet would define the start window as 30 days after the randomization date:

```
a.rand_date+30
```

However simple logic using the system date can still be used such as:

```
sysdate+30
```

End Window Snippet:

This field is same as the window start date except that it calculates the window end date

HTML Event Display Snippet:

This 1000 character text area field is used to define a snippet of HTML code that controls how the event is displayed when listed on a screen as part of a list of events for a specific person, place or thing. More specifically, it defines part of a single row of an HTML table. CDEPT displays lists of events in an HTML table. The first column of the table is pre-defined by CDEPT and includes buttons that control what actions can be performed, while the remaining columns are defined here.

Saving:

After completing all the fields to an event category click on Update Client and Table. For these changes to become active run update studyref.js option on the Development Tools Menu.

3.4.1c Defining Study Forms:

Forms represent the various sets of data that are collected as a part of the study. This level of definition merely names the study form and some basic characteristics of the form. This step does not involve defining the data entry routines for the form. That is performed elsewhere and after the initial study meta data has been defined.

Maintain Form Table

Form:

Form ID:

Description:

Detail Expression:

Completed Date Field Name:

Completed By Field Name:

Estimate Number Records:

HTML Form Display Snippet:

```
"<td><font size=1 face=arial
color=gray>Description:</font><br>" + g.formList[form_id].descript + "</td><td><font
size=1 face=arial color=gray>Filled By:</font><br>" + comp_by + "</td><td><font
size=1 face=arial color=gray>Filled on:</font><br>" + comp_d + "</td><td><font
size=1 face=arial color=gray>Entered by:</font><br>" + add_by + "</td><td><font
```

Temporary Save Interval:

Default Version:

Before Load:

After Save:

Encrypt in Oracle:

Figure 3.4.1c1

Data perspectives, event categories and event types must be defined prior to defining study forms. To define forms, select the Forms menu item under the Study Definition menu. Use

the pull down combo-box to either select an existing form to edit, or select “---new---” to add a new form (see above figure 3.4.1c). Each of the fields that define the form are explained below.

Form ID:

This is a 4-character code used to identify uniquely the form. It is used internally by the system. It is also used to define the Oracle table name that will store data related to this form. This code must be 4 characters long and it must be unique. All characters should be in upper case.

Description:

This 80-character string is used to store a description for the form. This description should match the form name used by those conducting the study. This description will be used in many reports and displays within CDEPT.

Detail Expression

This is an optional field that cannot be defined until the field names associated with form data entry have been determined. A JavaScript expression can be stored here which will be evaluated whenever the form is saved – the result will then be stored in the appropriate row of the formstatus table in the detail column. For example, is a protocol required multiple blood draws during a single study event it would be highly desirable to be able to display some unique identifiable information about the different blood draws on any summary screen. Placing the following in the detail expression field could do this:

“Draw Time: “+draw_time.value

This assumes there is a field called draw_time in the blood draw form. The resulting string i.e. “Draw Time: 10:45” would be then stored in the detail column of the formstatus table.

This column is available when defining the HTML Form Display Snippet below, which defines how a form appears on the screen in a list of forms on file. Any valid JavaScript expression

can be used – the expression is evaluated in the context of the HTML data entry form. The resulting string can be up to 50 characters long.

Completed Date Field Name:

This is an optional field that also cannot be defined until the field names associated with form data entry have been determined. This field stores the Oracle column name that corresponds to the date that the study form was completed. Like detail expression, this field is evaluated every time a form is saved, and the results are stored in the comp_d column of the formstatus table.

This field (if available) is used in the form history feature to calculate time between data collection and data entry.

Completed By Field Name:

This is an optional field that cannot be defined until the field names associated with form data entry have been determined. This field stores the Oracle column name that corresponds to the field that identifies who completed the study instrument. Like detail expression, this field is evaluated every time a form is saved, and the results are stored in the comp_by column of the formstatus table.

Estimated Number of Records:

This field is used internally by the system when allocating space for Oracle tables. Enter an estimate of the total number of records that this table will have. Multiply the number of anticipated subjects by number of times the form will be completed per subject. This estimate does not have to be perfect. Do not over-inflate this estimate, because this will waste disk space. This estimate will not limit the number of records that can be entered.

HTML Form Display Snippet:

This 1000 character text area field is used to define a snippet of HTML code that controls how the form is displayed when listed on a screen as part of a list of forms for a specific person, place or thing. More specifically, it defines part of a single row of an HTML table. CDEPT displays lists of forms as an HTML table. The first column of the table is defined by CDEPT and includes buttons that control what actions can be performed, while the remaining columns are defined here.

3.4.1d Defining Study Form Versions:

CDEPT supports multiple versions of the same study form. The first version should always be assigned version A, the second version - with the second version date should be assigned version B, etc. Each form must have at least one form version defined prior to developing data entry routines. The actual version date associated with this initial version can be changed later if needed.

Forms must be defined prior to defining versions specific to that form. To define a form version select Form Versions from the Study Definition menu. Two pull-down combo-boxes are used to select a form version for editing, or to add a new version (see figure below 3.4.1d1). First select the desired form from the pull-down combo-box. Then select the desired version to edit or select “---new---” to add a new version to the selected form. Each of the fields and links on the form version screen are explained below.

Maintain Form Versions Table

Form: Form 2 Hospitalization & Discharge Form

Version: F 12/07/1998

Version ID: F

Version Date: 12/07/1998 | [View Code Book](#) | [Create HTML Form](#) |

[Update Client and Table](#) [Cancel](#)

Figure 3.4.1d1

Version ID:

This one letter code of the version. All forms have as the first version "A". Enter the desired version letter.

Version Date:

Enter the date of the version. All study forms should be printed with a clear version date somewhere on the paper. In the case of electronic file transfer the version ID and version date correspond to the version of the file specification agreed upon. If the file specification needs to be changed a new version should be created.

View Codebook:

This link on the form versions screen will produce a code book for the study form. This code book will list all the individual form fields and their attributes.

Create HTML form:

This link relates to a function within the Form Wizard. When using the Form Wizard one of the last steps prior to creating the Oracle data table is to automatically generate a crude HTML form for data entry based on the instrument definition. Once the Oracle table has been

created – it is no longer possible to run the form Wizard on that form. This link provides a method for re-creating a crude HTML form even after the Oracle table has been created.

Saving:

After completing all the fields to an event category click on Update Client and Table. For these changes to become active run the Update StudyRef.js option on the Development Tools Menu.

3.4.1e Define protocol triggers

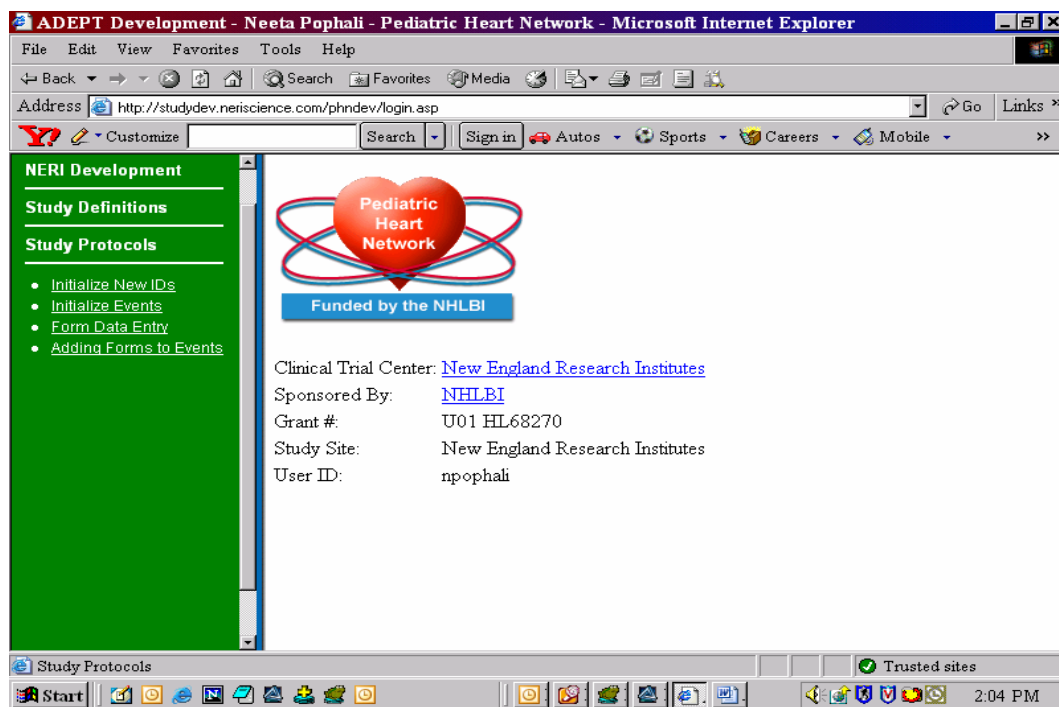
The CDEPT system is designed to support four different types of study protocol triggers. In addition it is possible to work with a project DBA to define custom triggers. This make it easier for multiple staff to understand and maintain the study, it reduces study costs and enables certain CDEPT reporting features to accurately reflect the study protocol.

The four types of triggers are briefly explained below:

| Trigger Type | Description | When Defined | Oracle Table |
|----------------------|---|------------------------------------|--------------|
| New ID | Triggers that execute when a new ID is added to the data perspective master table. These triggers expect study events. | After definition of core meta data | dp_protocol |
| Event Initialization | Triggers that execute when an event is initialized. These triggers expect specific study forms to be expected as part of the event. | After definition of core meta data | ie_protocol |

| | | | |
|-----------------|---|---|-------------|
| Form Data Entry | Triggers that execute when a form is added or edited. These triggers can expect additional study events or they can expect additional forms for the current study events. | After definition and testing of data entry routines | de_protocol |
| Optional Forms | These are not true triggers. They control which – if any – study forms should be “addable” to a study event. | After definition of core meta data | of_protocol |

All of these study protocol triggers can be maintained under the Protocols menu option off of the study definition menu as shown below. Each type of protocol trigger is explained in detail in the following sections.



3.4.1e i. Initialize New ID Protocol Triggers:

The first type of protocol trigger that CDEPT is designed to support relates to the adding of new Ids to a study. These triggers control what study events are to be expected for an ID. Depending how the data perspective is set up and how the study is designed this may occur when a user creates an ID using the CDEPT application or it may occur when a batch of Ids are added to the system by a programmer analyst of automated batch update.

The adding of expected events can be conditional on any information stored in the data prospective master view. Events can be added as “expected” or can be automatically initialized as soon as they are expected. This feature is sometimes useful in very simple studies where the study event model is not really needed.

Maintaining New ID Triggers:

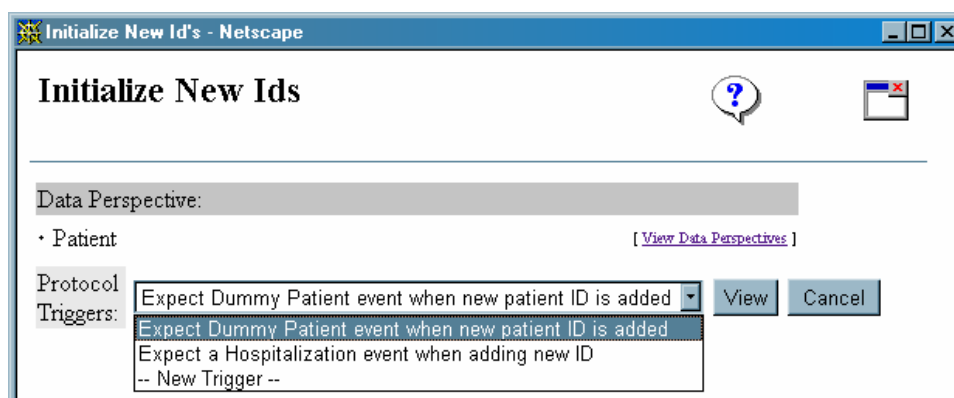
The New IDs Triggers can be maintained from the Study Protocols menu which can be found under the Study Definition menu. When first selected the system will prompt for the selection of a data perspective for which a New ID trigger will be defined. A screen example



appears below:

Figure: 1A

Choose the desired data perspective from the pull down list and click on View. A list of existing New ID triggers associated with the selected data perspective will be displayed.



Choose an exiting trigger, or use the last item on the list to select to add a new trigger

Figure: 1B

Attributes of an Optional Forms Triggers:

Once you have chosen to edit an existing trigger or to add a new trigger a trigger detail screen similar to that shown in figure 1C. Each of the attributes of a New ID triggers are explained in the following sections.

Description:

This is a 100 character text field where you enter a description of the trigger. Make it something meaningful because this is the description you use to find the protocol trigger for later editing.

Event Type to Expect:

This pull down includes a list of events types. The list will only include event types that are appropriate for the selected data perspective. Select the event type that should be added whenever a new ID for this data perspective is added.

Initialize New Ids

Data Perspective:

- Patient [\[View Data Perspectives \]](#)

Event Triggers:

- Expect Dummy Patient event when new patient ID is added [\[View Event Triggers \]](#)

Description: Expect Dummy Patient event when new patient ID is

Eventtype: Patient Data

Initiated: ☐

Conditional Test:

```
a.site_id<12
```

Save Cancel Delete

Figure: 1C

Initiated:

This option controls if the event type to be added should be automatically initiated. If an event is automatically initiated any forms that should be expected for that event will be immediately expected. Study events that are to occur in the future generally are not automatically initiated. Un-initiated events appear on the event summary screen with a yellow "Initiate" icon on the screen. It is generally not a good idea to automatically initiate an event where the event type is "dated".

Conditional Test:

Here you can define conditional logic that controls if the specified event type should be added when the ID is added to the data perspective. This conditional test is executed as a portion of a SQL where clause. Oracle syntax should be used here not JavaScript. Any

columns in the master view for the selected data perspective can be included. An CDEPT convention assigns the alias “a” to the master view. For example:

a.site_id=10

This example would mean the event would only be added for the new ID if the ID was from site 10.

Additional Implementation details:

If an initialize new ID trigger adds a new event where the event type is “windowed” then the logic that controls the start and end window dates come from the event type meta data.

1.4.1e ii Event Initialization Protocol Triggers:

The second type of protocol trigger that CDEPT is designed to support relates to the initialization of new study events. These triggers control what study forms are to be expected for a study event when the event is initialized. Events are generally initialized in one of two ways. Events can be initialized when the user clicks on the yellow event initialization icon on the events summary screen. Events can also be automatically initialized when the event is added via an Initialize new ID protocol trigger. Regardless of how the initialization occurs, CDEPT will scan the Event Initialization Protocol Triggers and add expected forms for that event.

The adding of expected forms for an event can be conditional on any information stored in the data perspective master view and events table.

Maintaining Event Initialization Triggers:

The Event Initialization Triggers can be maintained from the Study Protocols menu which can be found under the Study Definition menu. When first selected the system will prompt for the selection of an event type. A screen example appears below:

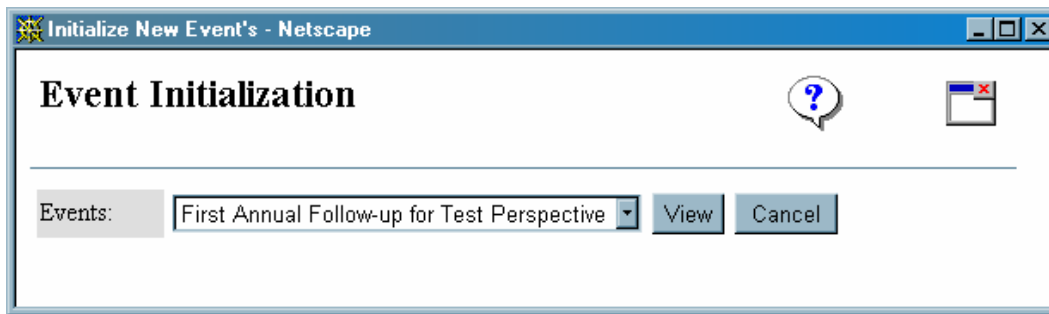


Figure: 1.4.1e iiA

Choose the desired event type from the pull down list and click on View.

A list of existing triggers associated with the selected event type will be displayed. Choose an existing trigger, or use the last item on the list to select to add a new trigger.

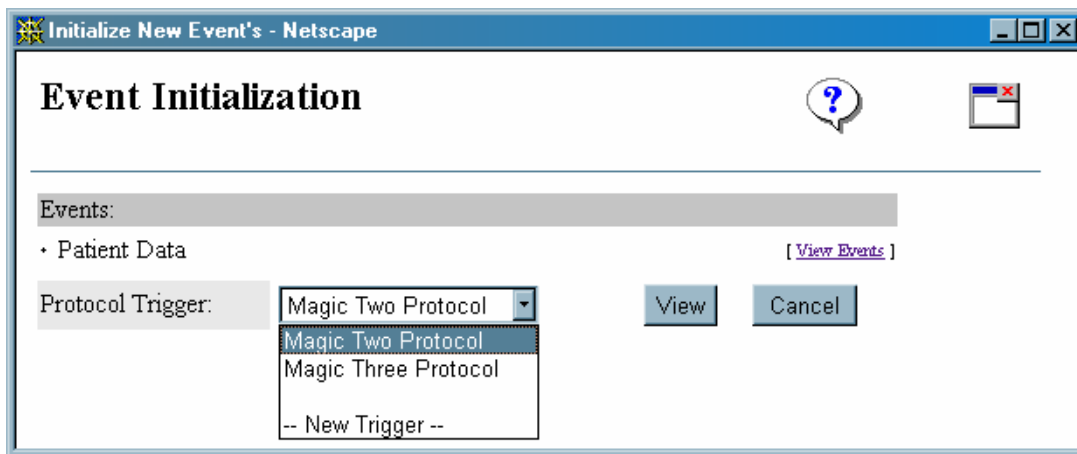


Figure: 1.4.1e iiB

Attributes of an Event Initialization Trigger:

Once you have chosen to edit an existing trigger or to add a new trigger a trigger detail screen is displayed. The figure 2.2.9.2C displays an example screen. Each of the attributes on the detail screen will be explained in the following sections.

Description

This is a 100 character text field where you enter a description of the trigger. Make it something meaningful because this is the description you use to find the protocol trigger for later editing.

Form to Expect:

This pull down includes a list the forms defined in the CDEPT project. Select the form that should be expected the selected event type is initiated.

Conditional Test:

Here you can define conditional logic that controls if the specified form should be added as expected when the event is initialized. This conditional test is executed as a portion of a SQL where clause. Oracle syntax should be used here not JavaScript. Any columns in the master view or events table can be included. An CDEPT convention assigns the alias "a" to the master view the letter "b" to the events table. For example:

a.site_id=10 and b.event_date>to_date('01/01/2000','MM/DD/YYYY')

This example would mean the form would only become expected as part of the newly initialized event if the ID was from site 10 and the event_date for the event was after 01/01/2000.

Event Initialization

Events:

- Patient Data [\[View Events \]](#)

Event Triggers:

- Magic Two Protocol [\[View Event Triggers \]](#)

Description: Magic Two Protocol

Forms: Form 2 Hospitalization & Discharge Form

Conditional Test:

Protocol Status: Required

Save Cancel Delete

Figure 1.4.1e iiC

Protocol Status:

This setting controls a protocol status flag in the formstatus table. There are three settings: Required, Protocol and Optional. Optional forms will no longer appear as expected if the user chose to delete the form. Both Required and Protocol will remain expected should the user choose to delete the form. At this point in time there is no difference between Required and Protocol.

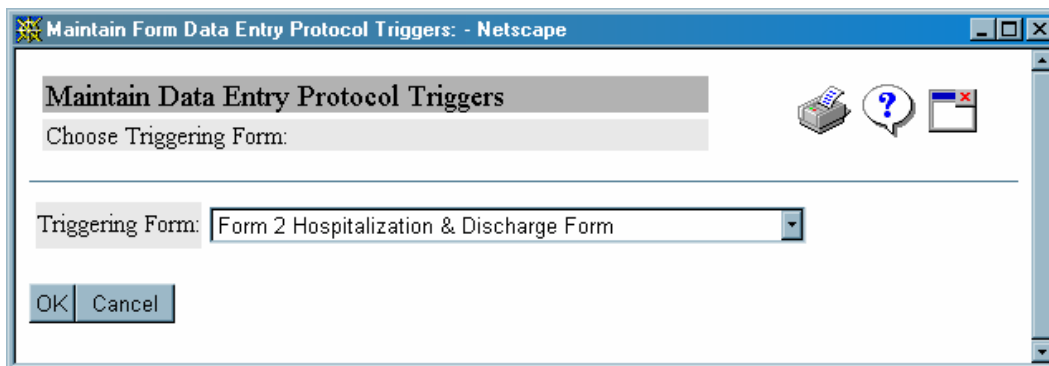
1.4.1e iii Form Data Entry Protocol Triggers:

The third type of protocol trigger that CDEPT is designed to support relates to the entry of specific study forms containing specific data. These triggers can be used to expect

additional study forms or to expect additional study events based on the entry of a specific form or entry of specific data within a form. The other triggers are generally “one way” triggers in that they add expectations to the system - and they cannot remove those expectations. The Form Data Entry triggers are “two way” triggers. This means that if the conditional test is true, the form or event becomes expected. If the condition is false, the system will attempt to remove the expected event or form. However, CDEPT has build in safe guards that will prevent it from removing an event or form if any data has been entered associated with the event or form.

Maintaining Event Initialization Triggers:

The Form Data Entry Triggers can be maintained from the Study Protocols menu which can be found under the Study Definition menu. When first selected the system will prompt for the select a form for which a form data entry trigger will be defined. A screen example



appears below:

Figure 1.4.1e iiiA

Choose the desired form the pull down list and click on OK. A list of existing triggers associated with the selected form will be displayed. Choose an existing trigger, or use the last item on the list to select to add a new trigger.

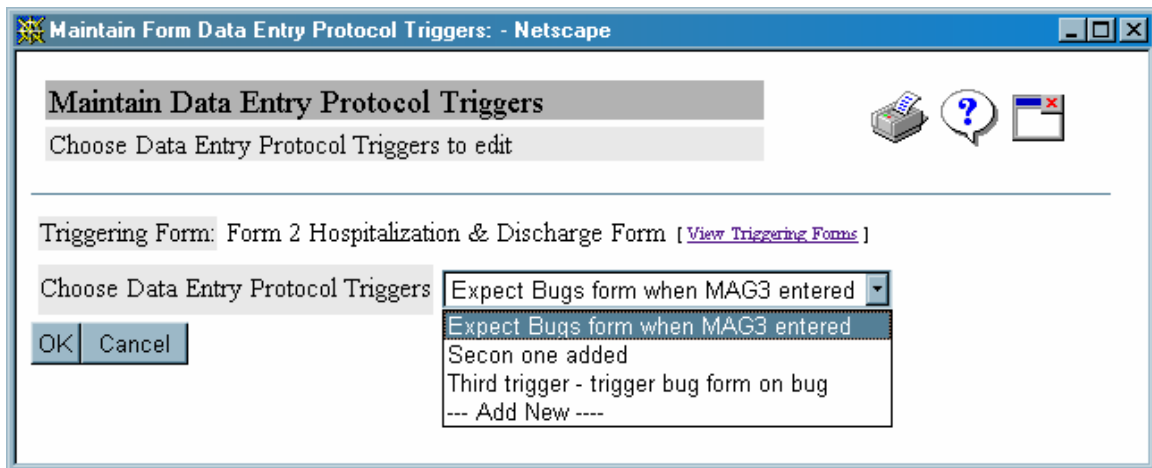


Figure 1.4.1e iii B

Attributes of an Event Initialization Trigger:

Once you have chosen to edit an existing trigger or to add a new trigger a trigger detail screen is displayed. The figure 3C displays an example screen.

Each of the attributes on the detail screen will be explained in the following sections.

Trigger Description

This is a 100-character text field where you enter a description of the trigger. Make it something meaningful because this is the description you use to find the protocol trigger for later editing.

Data Perspective:

Forms within CDEPT are not specifically linked to a data perspective. However for the trigger to operate it is necessary that the data perspective be known. Select the appropriate data perspective from the pull down list of data perspectives.

Type of Expectation:

Form Data Entry triggers come in two forms. Triggers that expect additional events and triggers that expect additional forms for the current event. Use the radio button to select the

desired type. Several of the attributes that follow are only relevant to one type or the other.

Do not worry about setting values for those attributes that are not needed.

Event to Expect:

This attribute only needs to be set if the Type of expectation is set to Event. A pull down list of all the event types in the system will be displayed. Select the appropriate event.

Maintain Data Entry Protocol Triggers - Netscape

Maintain Data Entry Protocol Triggers

Editing

Triggering Form: Form 2 Hospitalization & Discharge Form [[View Triggering Forms](#)]

Trigger Description: Expect Bugs form when MAG3 entered [[View Triggers](#)]

Data Perspective: Patient

Type of Expectation: ☒ Event expectation ☐ Form Expectation

Event to Expect: Patient Data

Auto Initiate: ☒

Form to Expect: Form 2 Hospitalization & Discharge Form

Protocol Status: Form Required

Trigger Conditional Test:

a. site_id=11

Event Triggers only

Event Triggers only

Form Triggers only

Form Triggers only

Figure 1.4.1e iii C

Auto Initiate Event:

This attribute only needs to be set if the type of expectation is set to event. This check box controls if the event should be automatically initiated when it is added. See section 2.2.9.1.5 for details.

Form to Expect:

This attribute only needs to be set if the type of expectation is set to form. This pull down will list all the study forms in the project. Select the study form that should be expected as part of this trigger.

Protocol Status

This attribute only needs to be set if the type of expectation is set to form. This controls the setting of a flag in the formstatus table. The three options are Required, Protocol and Optional. The Optional setting will allow a form expectation to be deleted completely should the user choose to delete data entered for that form. The other two settings will leave a form as expected when a user deleted data. At this point in time there is no difference between Required and Protocol settings.

Conditional Test:

Here you can define conditional logic that controls if the specified form or event should be added as expected. This conditional test is executed as a portion of a SQL where clause. Oracle syntax should be used here not JavaScript. Any columns in the master view, events, formstatus or the form data table can be used in the conditional test. An CDEPT convention assigns the alias "a" to the master view the letter "b" to the events table, c to the formstatus table and d to the data table. For example:

a.site_id=10 and d.gender=1

This example would mean the form would only become expected as part of the newly initialized event if the ID was from site 10 and the form initiating the trigger has a field called gender and its value is 1.

3.4.1e iv Optional forms Protocol Triggers:

The fourth type of protocol trigger that CDEPT is designed to support relates to optional study forms that can be added to a study event. These type of triggers are different from the other three in that they are not strict rules enforced totally within the database. Instead their

operation is in part dependent on the end user of the system. However, the protocol system allows the developer to control which forms can be added to which study events. Full implementation of the optional forms protocol triggers also requires adding a custom button to the form summary screen. This button is used to call up the add optional forms dialog

Maintaining Optional Forms Triggers:

The optional form Protocol Triggers can be maintained from the Study Protocols menu, which can be found under the Study Definition menu. When first selected the system will prompt for the selection of a study event for which an optional form trigger will be defined. See figure below.

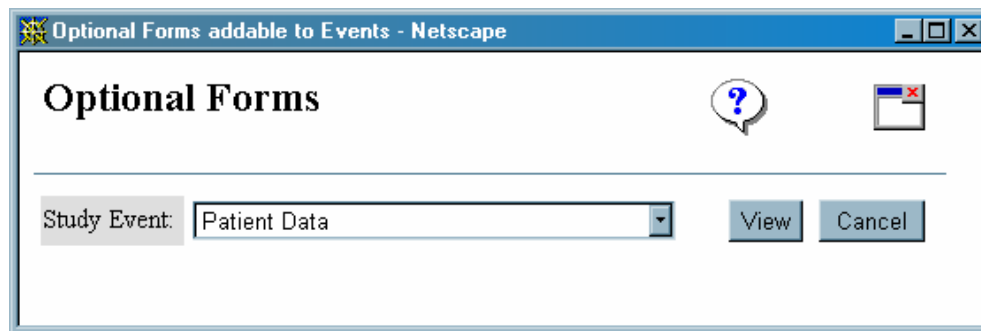


Figure 3.4.1e ivA

Choose the desired study event from the pull down list and click on View.

A list of existing optional form triggers associated with the selected event will be displayed.

Choose an exiting trigger, or use the last item on the list to select to add a new trigger.

Figure 3.4.1e ivB

Optional Forms addable to Events - Netscape

Optional Forms

Study Event:

• Patient Data [View Events]

Protocol Trigger:

Test of adding MAG3

Test of adding MAG3 no dups

Allow adding bug form

-- New Trigger --

View Cancel

Attributes of an Optional Forms Trigger:

Once you have chosen to edit an existing or add a new trigger a trigger detail screen will be displayed. The figure 3.4.1e iv C displays an example screen. Each of the attributes is explained in the following sections.

Description

This is a 100-character text field where you enter a description of the trigger. Make it something meaningful because this is the description you use to find the protocol trigger for later editing.

Optional Form:

This pull down includes a list of all the study forms defined in the CDEPT project. Select the form that should be addable to the selected study event.

Conditional Test:

Here you can define conditional logic that controls if the specified form should be addable to the selected study event. This conditional expression is evaluated in JavaScript and not as part of a SQL where clause. You can reference two special JavaScript objects: EVENTS and MASTER. The EVENTS object will have an attribute for each field in the EVENTS table.

During execution of the trigger - the

EVENTS object will be populated with information corresponding to the study event on the screen when the Add Optional Forms command is invoked. Likewise the MASTER object will have attributes for all the fields in the master view for the ID on screen when the Add Optional Forms command is invoked. For example:

MASTER.site_id==10 &&

EVENTS.event_date,substr(5,4)>2000

This example would only make the selected form addable if the current ID was from site 10 and the event date was after the year 2000. The attributes of the EVENTS and MASTER object should always be in lower case.

Optional Forms addable to Events - Netscape

Optional Forms

Study Event:

- Patient Data [\[View Events \]](#)

Optional Form Triggers:

- Allow adding bug form [\[View Event Triggers \]](#)

Description:

Forms:

Conditional Test:

Multiple Forms Per event? ☒

Display Order:

Figure 3.4.1e iv C

Multiple Forms:

Check this box if the same study form can be added to the same study event multiple times.

If this box is not checked the system will not allow the user to add a second form of the same kind to study event.

Display Order:

This option allows you to control the order in which optional forms appear in the add form dialog box. This is only relevant if there are more than one form that might be added to a study event.

Additional Implementation details:

In order to make the optional forms system fully functional it is necessary to add a hook to the form summary screen that calls the program that will display the allowable optional forms.

Hooks are added to the custStudyRef.js file. Below is an example that will add a button labeled "Opt Forms" to the form summary screen - that will call the optForms.asp program.

```
hooks[0]=new Hook(4,"1","Opt Forms','optForms.asp','2,'de');
```

Hooks is an array, use the next highest index number if your custStudyRef.js already has some Hooks defined. The above example assumes this is the first custom Hook to be defined.

After adding this hook and updating the studyRef.js file the form summary screen will include a button in the bottom frame labeled Opt Forms. Pressing this button will display a pull down list of optional forms that the user can add to the currently displayed event.

3.4.2 Define data entry routines for study forms

Define the form details as specified in the paper forms. The parameter inspector tool is used to define the details of all the fields in the form. The field details

include the field length, type, validation associated during data entering in this field. The following images show the details of the parameter inspector where the field master_id is defined in the form V100 version A.

Pediatric Heart Network
Parameter Inspector - Form: V100 Version: A

Fields ☒ Defaults ☐ Save

Field: **A1 Study Identification Number**

Add Insert Delete Re-Order

Question: A1 Help

Desc: Study Identification Number Help

Field Name: master_id Help

Field Type: Character Help

Field Length: 8 Help

Decimals: Help

Data Table: V100 Help

Validation: master_id.value==df.mast.master_id Help

Rigidity: Cannot Proceed Help

Target: Help

Documentation: Help

Stat. Type: Open Text Help

Codes: Help

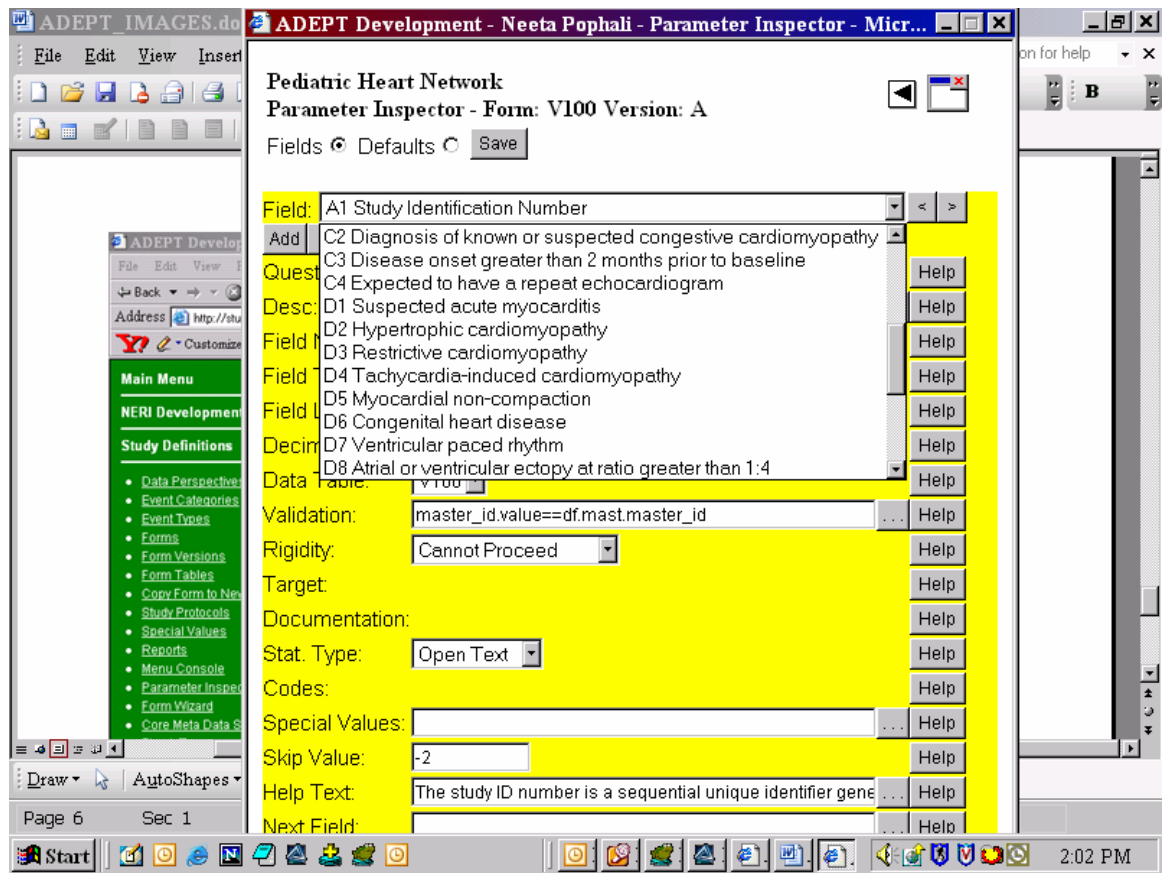
Special Values: Help

Skip Value: -2 Help

Help Text: The study ID number is a sequential unique identifier gene Help

Next Field: Help

Start 2:01 PM



3.4.3 Use Auto DBA tool to create Oracle tables

After defining all the forms details for the study create their respective Oracle tables using the Auto DBA tool. This tool creates the Oracle table with the field's size and structure defined in the parameter inspector.

3.4.4 Format HTML data entry screens to match paper forms

The HTML text can now be created using the 'maintain form version' screen. The create HTML file creates the basic HTML text with the fields defined in the parameter inspector. Modify this HTML file to match with the formatting in the paper forms. This completes all the form creation process. The following screenshot shows the form V100.

ADEPT Development - Neeta Pophali - Data Entry Session: ID Number 13050034 - Microsoft Internet Exp...

Pediatric Heart Network V100: VVV Study Initial Screening and Eligibility Form

SECTION A: KEY IDENTIFYING INFORMATION

A1. Study Identification Number

A2. Date of eligibility screening

A3. Date of echocardiogram

A4. Date of form completion

A5. Initials of Person completing form

SECTION B: PATIENT INFORMATION

B1. Patient date of birth

B2. Patient gender

B3. Race

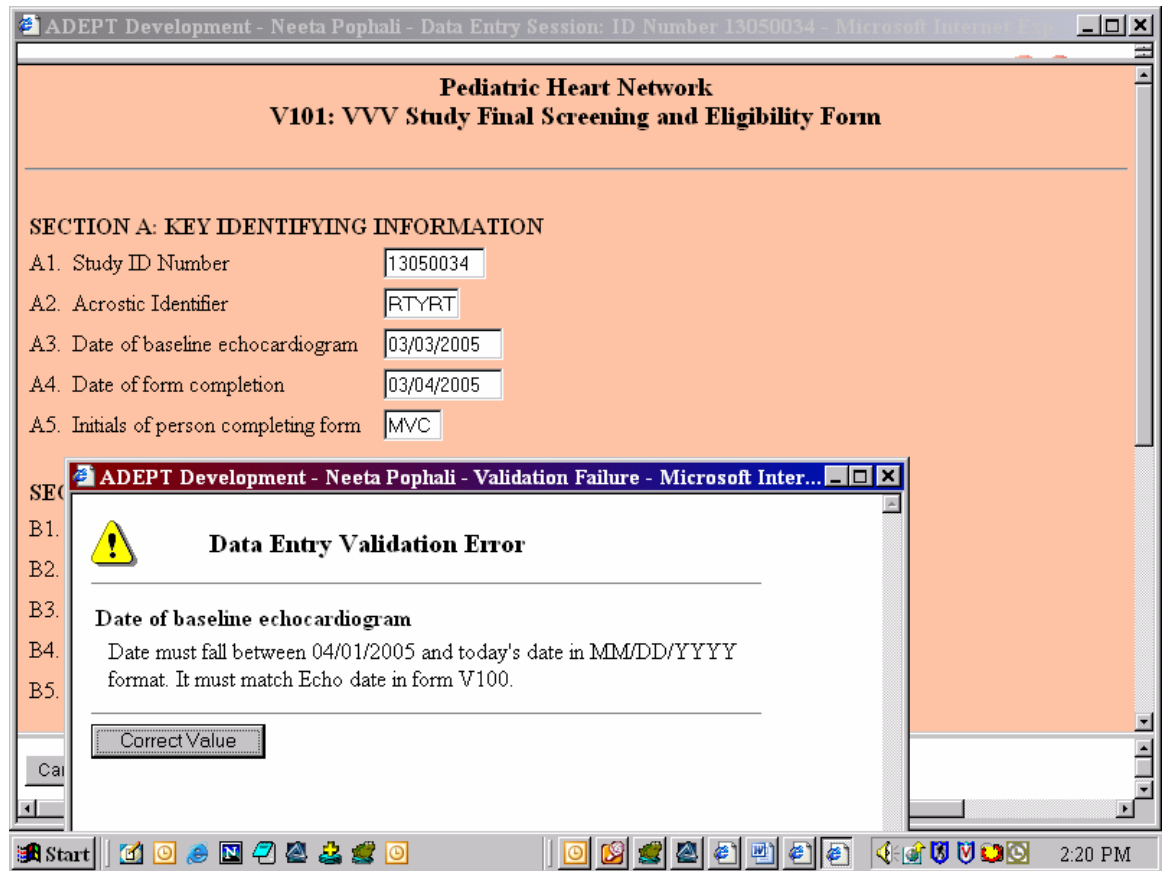
a. More than one race: specify

b. Other: specify other race

B4. Is patient of Hispanic or Latino origin?

Cancel ? [Icons] Other Commands

Start [Icons] 2:18 PM

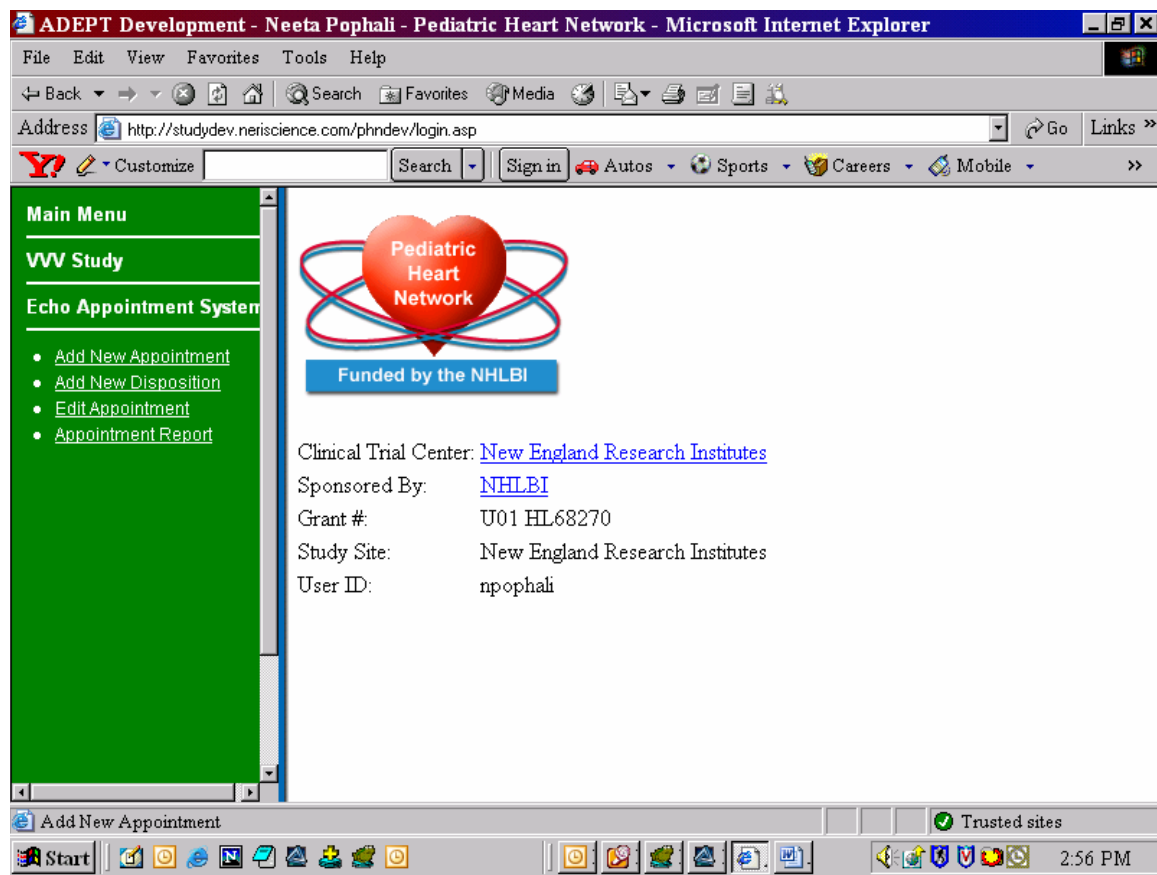


3.4.5 Test the data entry routines to make sure everything works correct.

3.4.6 Define custom features on standard CDEPT pages.

This step includes to add any additional feature needed by the study which is not included in the standard CDEPT website

Custom features added for a Pediatric Heart Study



Add new Appointment - Microsoft Internet Explorer

Pediatric Heart Network: Ventricular Volume Variability in Dilated Cardiomyopathy Study

Add New Appointment

| SITE | APPOINTMENT TYPE |
|---|--|
| 110 - Hospital for Sick Children 120 - Children's Hospital Boston 130 - Children's Hospital of New York 140 - Children's Hospital of Philadelphia 150 - Duke University Medical Center 152 - East Carolina University 153 - Wake Forest University Baptist Medical Center 160 - Medical University of South Carolina 170 - Primary Children's Medical Center 990 - Test Site - NERI Only | Baseline Follow-Up 1 Follow-Up 2 Follow-Up 3 Follow-Up 4 |
| | ID |
| | <input type="text"/> |
| | APPOINTMENT ECHO DATE (MM/DD/YYYY) |
| | <input type="text"/> |


OK Cancel

Start [Taskbar Icons] 2:56 PM

Test the whole system to make sure everything works correct. The following screenshot show the standard screens for the event summary screen which shows the details of the events defined and the form summary screen which shows the details of the forms defined in the system.

ADEPT Development - Neeta Pophali - Data Entry Session: ID Number 13050034 - Microsoft Internet Exp...

Study ID: **13050034** Eligible for Baseline Echo: **Yes**
 Acroscopic: **FLYBI** Consented to Baseline Echo: **Yes**
 DOB: **02/02/2002** Eligible for Follow-up Echo: **Yes**
 Gender: **Male** Date of Baseline Echo: **03/03/2005**


 Funded by the NHLBI

Summary of VVV Study Events


Initialize Event View Forms Delete Event Missing Event

| Description: | Window: | Date | Forms | Expected | Partial | Complete | Missing |
|------------------------------------|-------------------------|------|-------|----------|---------|----------|---------|
| VVV Study Screening and Enrollment | 04/01/2005 - 05/31/2006 | - | 5 | 1 | 2 | 2 | 0 |
| VVV Study Follow-Up Visit 1 | 06/01/2005 - 04/04/2006 | - | 4 | 1 | 1 | 2 | 0 |
| VVV Study Follow-Up Visit 2 | 01/15/2006 - 04/04/2006 | - | 4 | 1 | 2 | 1 | 0 |
| VVV Study Follow-Up Visit 3 | 11/10/2005 - 05/09/2006 | - | 4 | 1 | 2 | 1 | 0 |

Close Category of Data: VVV Study Events Ventricular Volume Variability Study ID Search

Event summary screen

ADEPT Development - Neeta Pophali - Data Entry Session: ID Number 13050034 - Microsoft Internet Exp...

| | | |
|---------------------------|--|--|
| Study ID: 13050034 | Eligible for Baseline Echo: Yes |  Funded by the NHLBI |
| Acroscopic: FLYBI | Consented to Baseline Echo: Yes | |
| DOB: 02/02/2002 | Eligible for Follow-up Echo: Yes | |
| Gender: Male | Date of Baseline Echo: 03/03/2005 | |

Summary of Forms for VVV Study Screening and Enrollment

| Form Version | Form Name | Status | Entered |
|----------------------------|--|-----------------------|---------------------|
| Form Version: A | Form V100: Initial Screening and Eligibility Form | Status: Complete | Entered: 03/04/2005 |
| Form Version: A | Form V101: Final Screening and Eligibility Form | Status: Complete | Entered: 03/04/2005 |
| Form Version: B | Form V102: Baseline Clinical Assessment and Medical Record Review Form | Status: Pending edits | Entered: 10/17/2005 |
| Form Version: B | Form V300: Local Echocardiography Form - Part 1 | Status: Pending edits | Entered: 10/17/2005 |
| Form Version: C 11/01/2005 | Form V301: Local Echocardiography Form - Part 2 | Status: Expected | Entered: |

Close Category of Data: VVV Study Events Add a Form Ventricular Volume Variability Study ID Search

form summary screen.

Chapter 4: Project History

4.1 How the project began.

"Finding reality" was already conducting medical research studies and wanted to have a generalized software framework to save the data, which can then be modified for specific project use. The research was conducted in the study centers all over the United States and the paper forms filled by the research participants at the centers were mailed by post to the main office of Finding reality. The data from the paper forms was then manually transferred on papers and then analyzed for specific finding. The reports about the project progress were manually written based on the project finding done manually. There was always a possibility of errors with all the manual work done. Also there was a significant time lag between data

collection and data analysis. While the data was analyzed and the reports were made there were a lot of data collected at the centers which was not counted towards for these reports because of the lack of proper system. To reduce these errors and make the system more readily accessible for everyone the president and the principle analyst on the projects in “Finding Reality” decided to develop CDEPT. The programming staff was appointed which included programming manager, Lead and test programmers. The entire website was done during 6-8 months of period. Then the individual research studies started to use this structure for their customized use.

I was involved in developing the software framework for customized study of Pediatric Heart Network (PHN) named VVV. VVV is a Trial of Variability of Echocardiographic Left Ventricular Mass, Volume and Ejection Fraction in Pediatric Patients with Congestive Cardiomyopathy. The objectives of the PHN are to accelerate research in the diagnosis and management of congenital and acquired pediatric heart disease, to standardize existing treatments, and to evaluate new therapies. The study was approved to be done in January of 2005. The project staff was appointed which included Project director, statistician, Data manager, Programming Lead and data collectors.

4.1.1 My role in project development.

My active role was involved in developing the customized CDEPT structure for the project VVV for PHN. This involved creating the basic metadata like data perspectives, forms, events and menu structures for the study. I was also responsible for the customized reports creation and appointment calendar. The routine data management meeting minutes creation and corresponding communication was done by me.

4.2 How the project was managed.

The study was expected to start enrolling patients by May 2005 which was the project Launch date. The study staff would meet once a week starting the study approval to go over the work to be done. The project director and the data manager would give details to the Programming Lead about the system requirements which included the patient's appointment system, study forms and events. The programming Lead would design the system and get it to the data manager for approval. After the design was approved the Lead developed the system using the basic model of CDEPT. The data manager was responsible for all the testing purposes. The study was launched into the production environment by the beginning of May 2005.

4.3 Significant events / Milestones in the project

This milestone report covers the project for the period from 12/01/2004 to 05/01/2005.

- 1) Project initiating and planning** was scheduled for 01/01/2005
 - a. **Description** – This milestone is achieved on the completion of project staff selection, project scope definition, cost estimate, staff estimate, duration estimate, communication plan definition
 - b. **Responsible** – Project Director, Programming manager

- 2) Complete project requirement study** was scheduled for 01/31/2005
 - a. **Description** - meet with project staff and study user requirement, system requirement
 - b. **Responsible** - Project Director, Programming Lead, data manager and data collectors.

- 3) Complete website front end and internal design** was scheduled for 02/25/2005
 - a. **Description** – complete the website front end design, website menus, reports as per the requirement and approve it from stakeholders
 - b. **Responsible** - Programming Lead, Project Director, data manager

- 4) Complete major website development** was scheduled for 04/01/2005
 - a. **Description** – complete programming, integrating the system and basic testing

b. **Responsible** - Programming Lead, data manager

5) **Complete major website testing** was scheduled for 04/15/2005

a. **Description** – complete all user testing and bug fixes

b. **Responsible** - Programming Lead, data manager and data collectors

6) **Site Rollout** to production was scheduled for 05/01/2005

a. **Description** – implement final changes suggested while testing and user testing and make the product ready for rollout

b. **Responsible** - Information Systems Lead, data manager

4.4 Changes to the project plan

While there were not much changes in the project plan with respect to launch date and the interim milestone achievement dates, there were some changes to the specification of the study forms and the appointment calendar design. Some of the changes suggested in the forms were significant and time consuming. Like the form V100 and V101 had a lot of changes in the order of the questions, which were redone just before the Site rollout date. Other than that there were not any major changes to the project's design.

4.5 Evaluation of whether or not the project met project goals

The customized study of Pediatric Heart Network named VVV met the project goals in most of the project creation aspects. Most of the major components were completed programming and testing before the site rollout with no major issues found in the production system after the project went live. Lots of users are using the system and have given positive feedback about the project usage. Most of the reports created showed accurate output of the study development and were useful for the study staff.

4.6 Discussion of what went right and what went wrong in the project

The project went right with respect to the timeline and Site rollout date. The front end of CDEPT looked great and was well appreciated. The system was easy to use for the first time users and did not need much user training.

Some of the project reports were slow and took more time to execute. There was some work that had to be redone because of specification changes.

4.7 Discussion of project variables and their impact on the project

When a user logs into CDEPT system a number of variables are created on both the client and server side of the application. On the client side the variables are defined in the main application window and can be accessed from other CDEPT windows using the window pointer variable g. All CDEPT windows should define this variable. On the server side the global variables are stored in the IIS session object (CDEPT manual).

The table below outlines some of these variables and is taken from the CDEPT manual. It provides the variable name used on the client side, the variable name used on the server side, and a brief description.

| Client Side Variable | Server Side Variable | Type client/ server | Description |
|-----------------------------|-----------------------------|----------------------------|--|
| sessionId | Session('SessionId') | String/ Number | Unique ID assigned by IIS to keep track of interactions between browser and server. CDEPT often uses this ID as part of temporary file names stored in \jsdata\ directory. |

| | | | |
|-------------------|------------------------------|--------------------|---|
| userId | Session('userId') | String | The User's Oracle user name. |
| studyName | Application('studyName') | String | Name of the study. |
| devStatus | Application('devStatus') | String/ Boolean | Whether CDEPT is running in development mode. Client side this is a string "true" or "false" |
| styleSheet | Application('styleSheet') | String | Name off ccs stylesheet used by project. |
| secureLevel | Session('secureLevel') | String | Security level of the user 1-9. |
| siteName | Session('siteName') | String | Name of the study site the user is associated with. |
| studyLogo | Application('studyLogo') | String | File name of study logo. |
| siteId | Session('siteId') | String/ Number | Site ID of the site the user is associated with. |
| webServer Path | Application('webServerPath') | String | On client this is the protocol, servername and directory of the project. On the server this is just the directory name. |
| webServer Root | --- | String | Equivalent to the server side webServerPath |
| CDEPTVersion | Application('CDEPTVersion') | String | Version of CDEPT |
| --- | Session('userPassword') | String | Password used by the user during login |
| --- | Application('ODBCDriver') | String | Name of ODBC driver used on the Server to access Oracle. |

Chapter 5: Lessons Learned and Next evolution of the project

5.1 Lessons learned from the project experience

Some of the lessons learned from the project experience are as follows

- In depth knowledge of client server architecture.
- Increased exposure to Active Server Pages and JavaScript, which was mainly used to code the software.
- Working on medical research software enhanced knowledge about the challenges in writing really complex reports used to do project calculations.
- Learned writing program specification and test plans.
- Got a good responsibility sense because of being the IS lead on the VVV project

5.2 Discussion of weather or not the project met initial project expectations

The project met the initial project expectations with respect to the specifications given. The user satisfaction was there after the website was launched. The project was also well coordinated and finished in time. The results from the project reports and the appointment colander was found to satisfactory and useful.

5.3 Next stage of evolution for the project would be if it continued

The next evolution of the project if it continued would have a better user maintenance system. Currently the user maintenance system does not satisfy all the FDA rules. The next evolution of CDEPT would fix all the loopholes the current system has. For example the FDA requires all the key staff persons in the project should be automatically notified when a new user is added or a existing user is deleted. Also the user password maintenance should be more effectively handled. The new version of CDEPT will handle internet popup more efficiently.

Works Cited

- 1) Finding Reality. "New England Research Institute". (October, 2006), 10 October 2006,
<www.neriscience.com>
- 2) CDEPT manual. "ADEPT systems developer manual", Jan 2005
- 3) PHN. "Pediatric Heart Network". (October, 2006), 10 October 2006,
<<http://www.pediatricheartnetwork.com>>
- 4) Elena Garderman (Feb 2004). "Ticketing System for a Three-Tiered Architecture". (Feb 2004), 10 October 2006,