DISTRIBUTION SHEET

To Distribution	From ISS - 62280	Page 1 of 1			
Project Title/Work Order QUEST Redesign (QUEST2)/R50211		Date 02/10/95			
		EDT No. 604720			
		ECN No.			
Name	MSIN	Text With All Attach.	Text Only	Attach./ Appendix Only	EDT/ECN Only
S. Y. Bian	B5-25		X		
F. D. Braaten	B5-25		X		
J. W. Hagan	B3-55		X		
M. L. Hermanson	A4-79		X		
M. S. Holowczak	B5-20		X		
W. J. Jones	B5-19		X		
S. A. Vega	L4-86		X		
Central Files	L8-077		X		
OSTI (2)	L8-074		X		

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

FEB 27 1995

ENGINEERING DATA TRANSMITTAL

Page 1 of 1

1. EDT 604720

2. To: (Receiving Organization) Distribution		3. From: (Originating Organization) LAN Systems Development		4. Related EDT No.: A/A	
5. Proj./Prog./Dept./Div.: WHC/CADS		6. Cog. Engr.: Mark Holowczak		7. Purchase Order No.: NA	
8. Originator Remarks: Original approval and release are requested.				9. Equip./Component No.: NA	
				10. System/Bldg./Facility: QUEST2/CADS	
11. Receiver Remarks:				12. Major Assn. Dwg. No.: NA	
				13. Permit/Permit Application No.: NA	
				14. Required Response Date: June 15th, 1994	

15. DATA TRANSMITTED					(F)	(G)	(H)	(I)
(A) Item No.	(B) Document/Drawing No.	(C) Sheet No.	(D) Rev. No.	(E) Title or Description of Data Transmitted	Impact Level	Reason for Transmittal	Originator Disposition	Receiver Disposition
1	SD WHC-CM-GN-PMP 30072	LA	0	QUEST2: SA/Release1 Supporting Documents Deliverable Set	QA	1		

16. KEY		
Impact Level (F)	Reason for Transmittal (G)	Disposition (H) & (I)
1, 2, 3, or 4 (see MRP 5.43)	1. Approval 2. Release 3. Information 4. Review 5. Post-Review 6. Dist. (Receipt Acknow. Required)	1. Approved 2. Approved w/comment 3. Disapproved w/comment 4. Reviewed no/comment 5. Reviewed w/comment 6. Receipt acknowledged

(G)	(H)	17. SIGNATURE/DISTRIBUTION (See Impact Level for required signatures)								(G)	(H)
Reason	Disp.	(J) Name	(K) Signature	(L) Date	(M) MSIN	(J) Name	(K) Signature	(L) Date	(M) MSIN	Reason	Disp.
1/2	1	Cog. Eng. Mark Holowczak	<i>[Signature]</i>	6/15/94	LA-84	B. L. Jones	<i>[Signature]</i>	6/15/94	BS-16	3	
1/2	1	Cog. Mgr. Samuel A. Yarbrough	<i>[Signature]</i>	6/15/94	SL-84	OSTI (2)	<i>[Signature]</i>	6/15/94	AB-04	1	1
1/2	1	QA Mark Hermonson	<i>[Signature]</i>	6/15/94	LA-84						
		Safety									
		Env.									
1/2	1	Sara Y. Blum	<i>[Signature]</i>	6/15/94	FS-25						
1/2	1	Floyd D. Green	<i>[Signature]</i>	6/15/94	BS-25						
18.		19.				20.				21. DOE APPROVAL (if required) Ltr. No.	
F. D. Green Signature of EDT Originator		Jeffrey Kinney for Sara Blum Authorized Representative Date 6/15/94				Samuel A. Yarbrough Cognizant/Project Engineer's Manager Date 6/15/94				☐ Approved ☐ Approved w/comments ☐ Disapproved w/comments	

RELEASE AUTHORIZATION

Document Number: WHC-SD-GN-CSWD-30072, Rev. 0

Document Title: QUEST2: Release 1: SA/Release 1 Supporting Documents Deliverable Set

Release Date: 01/25/95

This document was reviewed following the procedures described in WHC-CM-3-4 and is:

APPROVED FOR PUBLIC RELEASE

WHC Information Release Administration Specialist:

V. L. Birkland
V.L. Birkland

1/25/95

TRADEMARK DISCLAIMER. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors.

This report has been reproduced from the best available copy. Available in paper copy and microfiche. Printed in the United States of America. Available to the U.S. Department of Energy and its contractors from:

U.S. Department of Energy
Office of Scientific and Technical Information (OSTI)
P.O. Box 62
Oak Ridge, TN 37831
Telephone: (615) 576-8401

Available to the public from:

U.S. Department of Commerce
National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161
Telephone: (703) 487-4650

SUPPORTING DOCUMENT		1. Total Pages 150
2. Title QUEST2: Release 1: SA/Release 1 Supporting Documents Deliverable Set		3. Number WHC-SD-GN-CSND-30072
5. Key Words QUEST2, CADs, Corrective Action, Initiating Document, Tracking, Records, prototype, data definition, functional standards, conversion, performance criteria, operating environment APPROVED FOR		4. Rev No. 0
6. Author Name: F. D. Braaten Signature: <i>[Signature]</i> Organization/Charge Code: 62280/R40650		MD23F 502H
7. Abstract PUBLIC RELEASE <i>V. Burkland 1/25/95</i> This Deliverable Set contains the documents which represent the final analysis activities in the System Architecture phase of the QUEST2 project. These deliverables directly support the start of the construction work for the first release of QUEST2.		
<p>B. PURPOSE AND USE OF DOCUMENT - This document was prepared for use within the U.S. Department of Energy and its contractors. It is to be used only to perform, direct, or integrate work under U.S. Department of Energy contracts. This document is not approved for public release until reviewed.</p> <p>PATENT RIGHTS - This document copy, since it is transmitted in advance of patent clearance, is made available in confidence solely for use in performance of work under contract with the U.S. Department of Energy. This document is not to be published nor its contents otherwise disseminated or used for purposes other than specified above before patent approval for such release or use has been secured, upon request, from the Patent Counsel, U.S. Department of Energy Field Office, Richland, WY.</p> <p>DISCLAIMER - This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use of the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service, trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.</p>		<p>10. RELEASE STAMP</p> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>OFFICIAL RELEASE BY WHC DATE FEB 27 1995 35 SECTION 21</p> </div>
9. Impact Level QA		

**QUEST2: SA/RELEASE 1 SUPPORTING
DOCUMENTS DELIVERABLE SET**

**Document
Release
Page
Effective Date**

**WHC-SD-GN-CSWD-30072
New
1 of 1
May 31, 1994**

TABLE OF CONTENTS

NUMBER	TITLE	REV	EFFECTIVE DATE
	QUEST2: SA/Release 1 Supporting Documents Deliverable Set Introduction	0	5/31/94
Appendices			
A	Definition of Data Elements (P+ Doc No 320)	0	5/31/94
B	Conversion Rules (P+ Doc No 340)	0	5/31/94
C	System Performance Criteria (P+ Doc No 350)	0	5/31/94
D	Functional Standards (P+ Doc No 360)	0	5/31/94
E	Operating Environment (P+ Doc No 370)	0	5/31/94
F	Design Standards (P+ Doc No 380)	0	5/31/94
G	Evolutionary Prototype (P+ Doc No 235)	0	5/31/94

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

35

WESTINGHOUSE HANFORD COMPANY

Document

WHC-SD-GN-CSWD-30072

Section

Introduction, REV 0

Page

1 of 2

QUEST2: SA/RELEASE 1 SUPPORTING
DOCUMENTS DELIVERABLE SET

Effective Date

May 31, 1994

Organization

CA Data Systems

TITLE:

Approved by

QUEST2: SA/RELEASE 1 SUPPORTING
DOCUMENTS DELIVERABLE SET

S. A. Vega, Manager
CA Data Systems

INTRODUCTION

1.0 PURPOSE

This document contains deliverables which reflect the last of the System Architecture phase analysis for the Quality, Environmental, Safety Tracking System redesign (QUEST2) project.

2.0 SCOPE

These deliverables are focused on the final insights required to start functional design of the first QUEST2 release. They include the data definitions, conversion rules, standards for design and user interface, performance criteria, and rules to be followed during the prototyping activity described in the Project Management Plan.

3.0 FORMAT CONVENTIONS FOR THIS DOCUMENT

This project plan is composed of seven (7) Appendices. Each appendix addresses a specific aspect of analysis. These include:

Appendix	P+ DEL NO	Title and Content
Appendix A	320	Definition of Data Elements: The definitions for each QUEST2 data element, as contained in the CASE tool repository used by the project team.
Appendix B	340	Conversion Rules: The mapping of the current QUEST and CICS data elements into the new QUEST2 data base structure. Includes the general responsibilities of the respective system owners during conversion.
Appendix C	350	System Performance Criteria: The response times, event scheduling, and other such expectations from the finished system.
Appendix D	360	Functional Standards: The user expectations for system functional support and the system features that will implement those expectations.

**QUEST2: SA/RELEASE 1 SUPPORTING
DOCUMENTS DELIVERABLE SET**

**Document
Section
Revision Number
Page
Effective Date**

**WMC-SD-GN-CSWD-30072
Introduction
0
2 of 2
May 31, 1994**

INTRODUCTION

Appendix E	370	Operating Environment: Description of the hardware and software required to develop the system and to use the system when it is implemented.
Appendix F	380	Design Standards: The system construction standards that the development team will follow in constructing the system.
Appendix G	235	Evolutionary Prototype: The rules to be followed by the development team and the users during the prototyping activity described in the project plan.

1.0 PURPOSE

This document provides the Definition of Data Elements (320) for the QUEST2 system. These data element definitions are consistent with the System Architecture. They will be updated again during the Functional Design Phase of the project after the data model has been normalized and optimized. (See the Project Management Plan, WHC-SD-GN-PMP-20004 for definition of terms pertaining to the methodology used for QUEST2). The Definition of Data Elements (320) related in this document are sufficient to permit table design structures and to begin writing functional specifications.

2.0 FUNCTIONAL STANDARDS

This document includes only those data elements perceivable by the users. Data elements required to run and manage QUEST2 are not included at this time, but will be added when this 320 is updated during system construction.

The name of each data element conforms to the Hanford Site Data Naming Convention. The first one or two words of the data element name is the same as the name of a business entity defined in the Conceptual Data Model Diagram and the Definition of Entities (180). The last word of each data element name is a class word. The class word describes what the data element is for and how it might be used. Code, Name, Title and Text are examples of class words.

In addition to a text definition, the following information is included with each data element:

Integrity Rules: This describes if and when a data element is required to be entered into QUEST2. The conditions that must exist for data entry and update are also included.

Format: The data type and the input and display formats

Length: The size of each data element

Validation: Rules enforceable by the computer system needed to ensure correct and accurate data entry

Permitted Values: A list of all the appropriate values, if known.

The alternate name of each data element is an abbreviation of the data element name and also conforms to the Hanford Site Data Naming Conventions. The alternate name is used to define database table structures and indexes.

The following definitions are a report from the CASE tool used by the QUEST2 project and are included in this document as an attachment.

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 1
Excelerator

Name	Description	Alternate Name
Action Abstract Text	An abstract taken from the action description text. Coding and Edit Rules: Format: Free form text Length: 250 characters	Act_Abstract_Txt
Action Comments Date	The date and time the action comments were written. (Key) Coding and Edit Rules: Format: Date Time Length: 15 Validation: entered by the system	Act_Comments_DtT
Action Comments Text	A narrative of the progress made while working an action. Integrity Rules: Text cannot be modified or deleted. Coding and Edit Rules: Format: alphanumeric Length: 250	Act_Comments_Txt
Action Completed Date	The date the action work was actually completed. Coding and Edit Rules: Format: Date Length: 10 Validation: Not before the Action Due Date	Act_Completed_Dt
Action Description Text	A text narrative of the work to be done. Integrity Rules: Required for initial entry. Coding and Edit Rules: Format: Text Length: 2k minimum	Act_Description_Txt
Action Due Date	The date the action is scheduled for completion. Integrity Rules: Required before getting an "Active" status. Coding and Edit Rules: Format: Date mm/dd/yyyy Length: 10 Validation: Not before condition response due, or initiated dates.	Act_Due_Dt
Action Identifier	A system generated unique key for an action record. Coding and Edit Rules: Format: alphanumeric Length: 6 Validation: system created to guarantee uniqueness	Act_Id
Action Ref Type	A word used to describe other documents or alias numbers. (key) Coding and Edit Rules: Format: alphanumeric Length: 10	Act_Ref_Tp
Action Ref Type Alias Indicator	Some reference numbers are an alias for the associated action. Integrity Rules: Required for each action reference type. Coding and Edit Rules: Format: Alpha	Act_Ref_Type_Alias_Ind

A-A-1

320DATA.V?

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WHC-SD-6N-CSMD-30072
REVISION NUMBER 0

DATE: 11-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 2
Excalerator

Name	Description	Alternate Name
	Length: 1 Permitted Values: Y or N Default: N	
Action Ref Type Title	A text description of a reference type. Integrity Rules: Required for each action reference type. Coding and Edit Rules: Format: alphanumeric Length: 70	Act_Ref_Type_Ttl
Action Reference Identifier	A system generated unique key for condition reference number. Coding and Edit Rules: Format: The standard for system generated keys. Length:	Act_Ref_Id
Action Reference Number	The number used to reference another document or an alias. Coding and Edit Rules: Format: alphanumeric Length: 25	Act_Ref_No
Action Rejected Date	The date the completed action was rejected. Coding and Edit Rules: Format: Date Length: 10 Validation: Not before the completed date	Act_Rejected_Dt
Action Revised Due Date Count	The number of times the action due date is changed. Coding and Edit Rules: Format: integer Length: 2 Validation: Start at zero, the first change will be count one.	Act_Rev_Due_Date_Cnt
Action Title	A short text description for the action. Integrity Rules: Required at initial entry. Coding and Edit Rules: Format: alphanumeric Length: 70	Act_Ttl
Assessment Abstract Text	An abstract description for an initiating document. Coding and Edit Rules: Format: free form text Length: 250 characters	Ass_Abstract_Txt
Assessment Comments Date	The date and time the text was entered. {key} Coding and Edit Rules: Format: datetime Length: 15 Validation: This will be the system datetime, entered by the system.	Ass_Comments_DtT
Assessment Comments Text	This is used to add progress reports during the life of an initiating document. Coding and Edit Rules: Format: alphanumeric	Ass_Comments_Txt

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WMC-SD-GN-CSMD-30072
REVISION NUMBER 0

A-A-2

320DATA.V1

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 3
Excelerator

Name	Description	Alternate Name
	Length: 250	
Assessment Conducted Date	The date the assessment was conducted or performed. Coding and Edit Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: Required for came items. Permitted Values: Nothing before 1945.	Ass_Conducted_Dt
Assessment Discovery Date	The date an occurrence was discovered. Coding and Edit Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: Included for PNC items.	Ass_Discovery_Dt
Assessment Identifier	System generated unique key for an Initiating Document. This will be visible to the user for queries. Coding and Edit Rules: Format: YYMM99999 YY is last two digits of the year entered into the system. MM is the month entered into the system. 99999 a count of items as they are entered, beginning with 00001. Length: 9 Validation: System created unique id for initiating documents.	Ass_Id
Assessment Initiated Date	The date the initiating document takes effect. This is the date on the transmittal letter. Coding and Edit Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: Required. Permitted Values: Nothing before 1945.	Ass_Initiated_Dt
Assessment Introduction Text	A narrative text introduction for an initiating document. Entered from a word processor this may have codes displayable only by the word processor. Coding and Edit Rules: Format: text Length: 2K minimum Validation: Required.	Ass_Introduction_Txt
Assessment Scheduled Date	The quarter the assessment was scheduled. Coding and Edit Rules: Format: Date, MM/YYYY Length: 7 Validation: Not Required.	Ass_Scheduled_Dt
Assessment Summary Text	Narrative text concerning the summary for an initiating document. Entered from a word processor this may have codes displayable only by the word processor. Coding and Edit Rules: Format: text Length: 2K minimum Validation: Not Required.	Ass_Summary_Txt

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

MRC-SD-GN-CSMD-30072
REVISION NUMBER 0

A-A-3

320DATA.V1

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 4
Excelerator

Name	Description	Alternate Name
Assessment Title	The title of the assessment or initiating document. Coding and Edit Rules: Format: free form text, alphanumeric Length: 70 Validation: Required.	Ass_Ttl
Cond Revised Completion Dt Cnt	The number of times the planned completion date is changed. Coding and editing rules: Format: integer Length: 2 Validation: initialize to zero.	Cond_Revised_Comp_Dt_Cnt
Condition Abstract Text	A short abstract description of the condition. Coding and Edit Rules: Format alphanumeric Length: 250	Cond_Abstract_Text
Condition Actions Completed Date	The date the final action was completed. Coding and Editing Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: The date is taken from the last action completed date.	Cond_Actions_Comp_Dt
Condition Area Code	A code describing a Hanford defined area. Coding and Edit Rules: defined by FacilityCORE.	Cond_Area_Cd
Condition Closed Date	The date the condition has been recognized as corrected. Coding and editing Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: All actions must be closed. Permitted values: The date can't be earlier than other dates concerning the condition, like the completed date.	Cond_Closed_Dt
Condition Comments Date	The date and time the comments were entered. (key) Coding and Edit Rules: Format: datetime mm/dd/yyyy hh:mm Length: 15 Validation: This is the system datetime, entered by the system.	Cond_Coments_DtT
Condition Comments Text	A free form text field for comments about condition progress Coding and Edit Rules: Format: alphanumeric Length: 250 characters	Cond_Comments_Txt
Condition Description Text	A narrative describing or explaining the condition. Integrity Rules: Required before status of "draft pending acceptance" is allowed. Coding and Editing Rules: Format: Text Length: 2K minimum Validation: After Active status it cannot be modified.	Cond_Description_Txt
Condition Equipment Description	A text field used to list various types of equipment.	Cond_Equipment_Desc

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

MHC-SD-6N-CSMD-30072
REVISION NUMBER 0

A-A-4

320DATA.VT

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 5
Excelerator

Name	Description	Alternate Name
	Integrity Rules: A free form text field. Coding Edit Rules: Format: alphanumeric Length: 70 Validation: None	
Condition First Tag Code	A text field to enter keyword codes for searches and trending. Integrity Rules: Not required. No validation. Coding and Edit Rules: Format: alphanumeric Length: 10	Cond_First_Tag_Cd
Condition Fourth Tag Code	A text field to enter keyword codes for searches and trending. Integrity Rules: Not required. No validation. Coding and Edit Rules: Format: alphanumeric Length: 10	Cond_Fourth_Tag_Cd
Condition Identifier	The system generated unique key used to access a condition. Coding and Editing Rules: Format: alphanumeric, X999 Length: 4 Validation: system created	Cond_Id
Condition Location Description	A text field used to list or describe locations listed in no other category, ie contractor, facility, building, room or equipment. It may be map coordinates, railroad crossings, lamp posts, fire hydrants etc. Integrity Rules: A free form text field. Coding Edit Rules: Format: alphanumeric Length: 70 Validation: None	Cond_Location_Desc
Condition Memo Text	Notes to inputters concerning data updates and changes. Coding and Editing Rules: Format: alphanumeric Length: 250	Cond_Memo_Txt
Condition Planned Completion Dt	The date the last action will be completed. Coding and Editing Rules: Format: Date in the form MM/DD/YYYY Length: 10 Validation: The default will the latest of the action due dates or later.	Cond_Planned_Comp_Dt
Condition Ref Type	A descriptive word for other types of documents. (key id) Coding and Edit Rules: Format: alphanumeric Length: 10	Cond_Ref_Tp
Condition Ref Type Alias Ind	Set if the reference number is an alias for the condition.	Cond_Ref_Type_Alias_Ind

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

MHC-SD-GM-CSMD-30072
REVISION NUMBER 0

A-A-5

320DATA.V3

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 6
Excelerator

Name	Description	Alternate Name
	Integrity Rules: Required for each condition reference type. Coding and Edit Rules: Format: alpha Length: 1 Permitted values: Y or N	
Condition Ref Type Description	A description of another type of document number. Integrity Rules: Required for each document reference type. Coding and Edit Rules: Format: alphanumeric Length: 70	Cond_Ref_Type_Desc
Condition Reference Identifier	A system generated unique key for condition reference number. (key) Integrity Rules: Not visible to the user. The system will assign one to first reference number and count one for each ref number added to the table. Coding and Edit Rules: Format: Integer Length: Permitted Values: 1 to maxint	Cond_Ref_Id
Condition Reference Number	A number referencing another condition or an alias number. Coding and Edit Rules: Format: alphanumeric Length: 25	Cond_Ref_No
Condition Revised Completion Dt	The new or modified planned completion date. Coding and editing Rules: Format: date, MM/DD/YYYY length: 10 Validation: It cannot be before the planned completion date of the latest action.	Cond_Revised_Comp_Dt
Condition Room Number	A room number. Coding and Edit Rules: Format: Alphanumeric Length: 10	Cond_Room_No
Condition Second Tag Code	A text field to enter keyword codes for searches and trending. Integrity Rules: Not required. No validation. Coding and Edit Rules: Format: alphanumeric Length: 10	Cond_Second_Tag_Cd
Condition Third Tag Code	A text field to enter keyword codes for searches and trending. Integrity Rules: Not required. No validation. Coding and Edit Rules: Format: alphanumeric Length: 10	Cond_Third_Tag_Cd
Condition Title	A short free form text description of the condition.	Cond_Ttl

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

NHC-SD-GM-CSHD-30072
REVISION NUMBER 0

A-A-6

320DATA.V1

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WHC-SD-GN-CSMD-30072
REVISION NUMBER 0

PAGE 7
Excelerator

QUEST2 Definition of Data Elements (320)

DATE: 13-JUN-94
TIME: 08:56

Name	Description	Alternate Name
Condition Type	<p>A word used to categorize conditions. (key id)</p> <p>Integrity Rules: Required Coding and Editing Rules: Format: Alphanumeric Length: 70</p> <p>Permitted Values: Finding, Observation, Concern, Deficiency, Recommendation or Issue.</p>	Cond_Tp
Condition Type Description	<p>A text description of a condition type.</p> <p>Integrity Rules: Required for Initial entry. Coding and Edit Rules: Format: alphanumeric Length: 70</p>	Cond_Type_Desc
Condition Verified Date	<p>Once work is completed, the date all actions are reviewed.</p> <p>Coding and editing Rules: Format: date, MM/DD/YYYY Length: 10 Validation: all actions must be closed. Not required. Permitted values: can't be earlier than the last closed date.</p>	Cond_Verified Dt
Contractor Identifier	<p>Unique contractor id.</p> <p>Integrity Rules: Included in a download from PeopleCore Coding and Edit Rules: Format: alpha Length: 3 Permitted Values: Defined by the PeopleCore system.</p>	Contractor_id
Distribution List Identifier	<p>A system generated id for a distribution list.</p> <p>Coding and Edit Rules: Format: The standard for system generated id's. Length:</p>	Dist_List_Id
Document Class Description	<p>A text description name for each class type. example occurrence</p> <p>Integrity Rules: Required for each document class type. Coding and Edit Rules: Format: alphanumeric Length: 25</p>	Doc_Class_Desc
Document Class Type	<p>The document class is used to identify the type of initiating document.</p> <p>Required fields may be different for each document class. (key)</p> <p>Integrity Rules: Required for Initial entry of an initiating document. Coding and Edit Rules: Format: alphanumeric</p>	Doc_Class_Tp

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 8
Excelerator

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

Name	Description	Alternate Name
	Length: 3 Permitted Values: Include AUD - Audit AST - Assessment SAT - Self-Assessment MAT - Management Assessment SUR - Surveillance APP - Appraisal INV - Investigation EMC - Employee Concern TTA - Tiger Team Assessment PAT - Performance Assessment Team OCC - Occurrence NOM - Non-CAPS PAT - Performance Assessment Team	
Document Ref Type	Used to describe where the document reference number originated, who owns that reference number. (key) Integrity Rules: Not required. Coding and Edit Rules: Format: alphanumeric Length: 5	Doc_Ref_Tp
Document Ref Type Alias Ind	Some reference numbers will be a number used to refer to the initiating document. Correspondence Control and external agencies assign their own numbers to initiating documents. The user will be able to retrieve documents based on these numbers. Integrity Rules: Required for each document reference type. Coding and Edit Rules: Format: alphanumeric Length: 1 Permitted Values: Y - Yes N - No	Doc_Ref_Type_Alias_Ind
Document Ref Type Description	A text description of the document reference type. Integrity Rules: Required for each document reference type. Coding and Edit Rules: Format: alphanumeric Length: 25	Doc_Ref_Type_Desc
Document Reference Identifier	Used as a unique number for documents outside the system. It is system generated to ensure uniqueness. (key) Coding and Edit Rules: Format: numeric Length: 6 Permitted Values: This will be a count of document reference numbers beginning with one. The row number.	Doc_Ref_Id
Document Reference Number	A number used by an external system used to identify a document that has some bearing on an initiating document. Coding and Edit Rules:	Doc_Ref_No

A-A-8

330DATA.V1

MHC-SD-GM-CSMD-30072
REVISION NUMBER 0

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 9
Excellerator

Name	Description	Alternate Name
	Format: alphanumeric Length: 25	
Good Practice Description	The text narrative explaining the good practice. Integrity Rules: Required for each good practice. Coding and Edit Rules: Format: text Length: 2k minimum	Good_Practice_Desc
Good Practice Identifier	The unique system generated identifier for a good practice. (key) Coding and Edit Rules: Format: The standard format for system generated id's. Length:	Good_Practice_Id
Good Practice Title	The title description for a good practice. Integrity Rules: Required for each good practice. Coding and Edit Rules: Format: alphanumeric Length: 70	Good_Practice_Ttl
Help Text	The text information the user reads for help. Integrity Rules: Required. Coding and Edit Rules: Format: text Length: 2k minimum	Help_Txt
Help Text Identifier	The system generated unique id for help text. (key) Coding and Edit Rules: Format: Integer Length: Permitted Values: 1 to maxint	Help_Text_Id
Help Text Type	The type of help text. This could be the user guide, a manual or some other category. Integrity Rules: Required for each help text. Coding and Edit Rules: Format: alphanumeric Length: 20	Help_Text_Tp
Intangible Code	This is a code representing a specific program or activity that has been assessed. (key) Coding and Edit Rules: Format: alphanumeric Length: 5 Permitted Values: to be determined.	Intang_Cd
Intangible Description	A textual description of a specific program or activity. Integrity Rules: Required for each intangible code. Coding and Edit Rules: Format: alphanumeric	Intang_Desc

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WHC-SD-GN-C5MD-30072
REVISION NUMBER 0

A-A-9

320DATA.V1

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WHC-SD-GN-CSWD-30072
REVISION NUMBER 0

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 10
Excelerator

Name	Description	Alternate Name
Intangible Type	<p>The type of intangible item assessed. (key id)</p> <p>Coding and Edit Rules: Format: alpha Length: 1</p> <p>Permitted Values: P - Program A - Activity</p>	Intang_Tp
Intangible Type Description	<p>A textual description of a intangible item type is. Activity.</p> <p>Integrity Rules: Required for each Intangible type.</p> <p>Coding and Edit Rules: Format: alphanumeric Length: 10</p>	Intang_Type_Desc
Job Frequency Code	<p>How often each production job will run.</p> <p>Coding and Edit Rules: Format: alphanumeric Length: 3</p> <p>Permitted Values: D - Daily Dnn - Every nn Days W - Weekly Wnn - Every nn weeks M - Monthly Mnn - Every nn Months</p>	Job_Frequency_Cd
Job Identifier	<p>The system generated unique id for each quest2 batch job.</p> <p>Coding and Edit Rules: Format: integer Length: Permitted Values: 1 to maxint</p>	Job_Id
Job Run Date	<p>The next date and time the production job will run.</p> <p>Integrity Rules: By adding the Job Frequency to the current Job Run Date the next Job Run Date will be determined.</p> <p>Coding and Edit Rules: Format: Datetime Length: 15</p> <p>Validation: This will be generated by the system but can be overridden.</p>	Job_Run_Dt
Job Title	<p>The title for the production batch job.</p> <p>Integrity Rules: Required for each production job.</p> <p>Coding and Edit Rules: Format: alphanumeric Length: 20</p>	Job_Ttl
Message Identifier	<p>The system generated unique id for each system message.</p> <p>Coding and Edit Rules: Format: numeric Length: Permitted Values: 1 to 999999</p>	Message_Id

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 11
Exceleator

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

Name	Description	Alternate Name
Message Text	The actual text of the system message. Integrity Rules: Required for each message. Coding and Edit Rules: Format: alphanumeric Length: 80	Message_Txt
Message Type	QUEST2 system messages can be of several types. These are context help, Message_Tp error or informational messages to online users. Integrity Rules: Required for each message. Coding and Edit Rules: Format: alpha Length: 1 Permitted Values: E - error H - Help I - Information W - warning	Message_Tp
Notice Criteria Description	The criteria including initiating documents, conditions and actions in a Notice_Criteria_Desc cc mail message. This will probably be SQL statements. Integrity Rules: Required for each notice Coding and Edit Rules: Format: alphanumeric Length: 250	
Notice Text	The actual text of a ccmall message. Integrity Rules: Required for each notice. Coding and Edit Rules: Format: text Length: 2k minimum	Notice_Txt
Notice Title	A description of the type of cc-mail message. Integrity Rules: Required for each notice. Coding and Edit Rules: Format: alphanumeric Length: 30	Notice_Ttl
Notice Type	The category the cc mail notice will cover. (key) These are the events that will trigger a system generated CC Mail message. However, to reduce the number of ccmall messages sent and received a ccmall message may include one or more notice types. Coding and Edit Rules: Format: alpha Length: 3 Permitted Values: AC - Active Condition, AI - Active Initiating Document, AA - Active Action, DC - Draft Condition, DI - Draft Initiating Document, RI - Response to an Initiating Document,	Notice_Tp

A-A-11

320DATA.V1

MHC-SD-GN-CSMD-30072
REVISION NUMBER 0

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 12
Excelerator

Name	Description	Alternate Name
	CD - Coming Due in 10 working Days, AC - Actions Complete	
PPG As Found Indicator	A flag set when this score is the "as found" ppg. Integrity Rules: Required for each ppg. Coding and Edit Rules: Format: alpha Length: 1 Permitted values: Y or N	PPG_As_Found_ind
PPG Assumption Text	The assumptions made in determining the risk of a condition. Coding and Edit Rules: Format: text Length: 2k minimum	PPG_Assumption_Txt
PPG DOE Mission Prob. Value	The relative risk value for DOE mission impact. Coding and Edit Rules: Format: numeric 9999.9999 Length: Permitted Values: The permitted range for discrete PPG factors is defined in WMC-CM-1-4. Currently the max is 3000, the low is .0001	PPG_DOE_Mission_Val
PPG Dom Cons Probability Value	The probability of the dominating consequence happening. Integrity Rules: Required along with the dominating consequence code. Coding and Edit Rules: Format: numeric 9999.9999 Length: Permitted Values: The permitted range for discrete PPG factors is defined in WMC-CM-1-4. Currently the max is 3000, the low is .0001	PPG_Dom_Cons_Prob_Ftr
PPG Dom Cons Risk Value	The relative risk posed by the dominating consequence. This is defined in WMC-CM-1-4. Integrity Rules: Required with the dominating consequence code: Dimensions: Format: alpha Length: 1 Permitted Values: defined in WMC-QS-1-4	PPG_Dom_Cons_Risk_Val
PPG Dominating Consequence Cd	The dominating consequence used to determine the ppg value. This is defined in WMC-CM-1-4. Integrity Rules: Required along with PPG Value. Coding and Edit Rules: Format: alpha Length: 1 Permitted values: a dominating consequence code as defined in whc-m-1-4	PPG_Dom_Cons_Cd
PPG Env. Protection Value	The relative risk value for environmental protection. Coding and Edit Rules: Format: numeric 9999.9999 Length: Permitted Values: The permitted range for discrete PPG factors is	PPG_Env_Prot_Val

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WMC-SD-6N-C5ND-30072
REVISION NUMBER 0

A-A-12

3300DATA.V1

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 13
Excelerator

Name	Description	Alternate Name
	defined in WMC-CM-1-4. Currently the max is 3000, the low is .0001	
PPG Evaluation Date	The date the ppg value was determined. Coding and Edit Rules: Format: Date Length: 10	PPG_Evaluation_Dt
PPG Evaluation Team Text	A listing of the members in CAEG, Corrective Action Evaluation Group. Coding and Edit Rules: Format: alphanumeric Length: 250	PPG_Evaluation_Team_Txt
PPG External Reaction Value	The relative risk of an external reaction. Coding and Edit Rules: Format: numeric 9999.9999 Length: Permitted Values: The permitted range for discrete PPG factors is defined in WMC-CM-1-4. Currently the max is 3000, the low is .0001	PPG_Ext_Reaction_Val
PPG Identifier	A system generated unique key for ppg data. Coding and Edit Rules: Format: The standard for system generated id's. Length:	PPG_Id
PPG Investment Protection Value	The relative risk to investment protection. Coding and Edit Rules: Format: numeric 9999.9999 Length: Permitted Values: The permitted range for discrete PPG factors is defined in WMC-CM-1-4. Currently the max is 3000, the low is .0001	PPG_Inv_Prot_Val
PPG Public Safety Value	The risk value to public safety. Coding and Edit Rules: Format: numeric 9999.9999 Length: Permitted Values: The permitted range for discrete PPG factors is defined in WMC-CM-1-4. Currently the max is 3000, the low is .0001	PPG_Public_Safety_Val
PPG Value	The final value, determined by adding the values from all factors. Integrity Rules: Required before condition can become active. Coding and Edit Rules: Format: numeric Length: 99999.9999 Permitted Values: .0001 to 10,0000, currently defined in WMC-CM-1-4.	PPG_Val
PPG Worker Safety Value	The relative risk value for worker safety. Coding and Edit Rules: Format: numeric 9999.9999 Length: Permitted Values: The permitted range for discrete PPG factors is defined in WMC-CM-1-4. Currently the max is 3000, the low is .0001	PPG_Worker_Safety_Value
Person CEmail Name	Unique cemail name used for addressing messages by CEmail system. Integrity Rules:	Perz_CEmail_Nm

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WMC-SD-6N-CSMD-30072
REVISION NUMBER 0

A-A-13

320DATA.V1

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 14
Exceerator

Name	Description	Alternate Name
	Included in a download from PeopleCore. Coding and Edit Rules: Format: alpha Length: 30	
Query Description	A narrative description of what the query is and how to use it. Coding and Edit Rules: Format: alphanumeric Length: 250	Query_Desc
Query Identifier	A system generated unique id for stored queries. Coding and Edit Rules: Format: Integer Length: Permitted Values: 1 to max int	Query_ID
Query Text	The actual query itself. Probably SQL statements. Coding and Edit Rules: Format: text Length: 2k minimum	Query_Txt
Query Title	A text description of the query. Integrity Rules: required for each query. Coding and Edit Rules: Format: alphanumeric Length: 30	Query_Ttl
RCA Contributing Cause Code	The direct cause code as defined by the DOE order 5000.3B. Integrity Rules: Required for items originated by RL Coding and Edit Rules: Format: alpha Length: 2 Validation: Must be valid code	RCA_Cont_Cause_Cd
RCA Direct Cause Code	The direct cause code as defined by the DOE order 5000.3B. Integrity Rules: Required for items originated by rl. Coding and Edit Rules: Format: alpha Length: 2 Validation: Must be valid code	RCA_Direct_Cause_Cd
RCA Identifier	The unique identifier for root cause analysis data. (key id) Coding and edit Rules: Format: The standard for system generated id's. Length:	RCA_Id
RCA Root Cause Code	The direct cause code as defined by the DOE order 5000.3B. Integrity Rules: Required for rl originated items. Coding and Edit Rules: Format: alpha Length: 2	RCA_Root_Cause_Cd

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

MHC-SD-6N-C5MD-30072
REVISION NUMBER 0

A-A-14

320DATA.V1

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

MHC-SD-GN-CSWD-30072
REVISION NUMBER 0

DATE: 13-JUN-94
TIME: 08:56

QUEST2 Definition of Data Elements (320)

PAGE 15
Exemplarator

Name	Description	Alternate Name
	Validation: Must be valid code	
Request Actual Cost	The actual cost to work and implement the request. Coding and Edit Rules: Format: numeric \$999.999 Length: 8	Req_Actual_Cost
Request Actual Hours	The total hours to work and implement the request. Coding and Edit Rules: Format: numeric Length: 4	Req_Actual_Hours
Request Approved Indicator	A flag set when the Camb, corrective action management board, has decided the change will be implemented. Coding and Edit Rules: Format: alpha Length: 1 Permitted Values: Y or N	Req_Approved_Ind
Request Closed Date	Date the request is closed. Coding and Edit Rules: Format: date Length: 10	Req_Closed_Dt
Request Comments Text	A narrative of what was actually done to implement the request Coding and Edit Rules: Format: alphanumeric Length: 250	Req_Comments_Text
Request Current Priority Code	The priority resulting from the latest CAMB review. Coding and Edit Rules: Format: alpha Length: 1 Permitted Values: H, M, L	Req_Current_Priority_Cd
Request Current Priority Date	The date of the latest CAMB review. Coding and Edit Rules: Format: date Length: 10	Req_Current_Priority_Dt
Request Description	Text explaining the Change request. Coding and Edit Rules: Format: alphanumeric Length: 250	Req_Desc
Request Estimate Date	Date the estimate was requested by the CAMB. Coding and Edit Rules: Format: date Length: 10	Req_Estimate_Dt
Request Estimate Received Date	Date the estimate was received by the CAMB. Coding and Edit Rules: Format: date Length: 10	Req_Estimate_Received_Dt

DATE: 13-JUN-94
TIME: 08:57

QUEST2 Definition of Data Elements (320)

PAGE 16
Exccelerator

Name	Description	Alternate Name
Request Estimated Cost	The estimated cost to work and implement the request. Coding and Edit Rules: Format: money \$999.999 Length: 8	Req_Estimated_Cost
Request Estimated Hours	The total estimated hours to work and implement the request. Coding and Edit Rules: Format: numeric Length: 4	Req_Estimated_Hours
Request Identifier	The number given to the request. Coding and Edit Rules: Format: alphanumeric YY-nnnn Length: 7	Req_ID
Request Initial Priority Code	The priority resulting from the first CAMB review. Coding and Edit Rules: Format: alpha Length: 1 Permitted Values: R, M, L	Req_Initial_Priority_Cd
Request Initial Priority Date	The date of the first CAMB review. Coding and Edit Rules: Format: date Length: 10	Req_Initial_Priority_Dt
Request Justification Desc	A text explanation of why the change is necessary. Coding and Edit Rules: Format: alphanumeric Length: 250	Req_Justification_Desc
Request Needed Date	The date the submitter wants the request implemented. Coding and Edit Rules: Format: date Length: 10	Req_Needed_Dt
Request Status Code	The status of the request. Coding and Edit Rules: Format: alpha Length: 8 Permitted Values: open, rejected, closed	Req_Status_Cd
Request Submitted Date	The date the request was made. Coding and Edit Rules: Format: Date length: 10	Req_Submitted_Dt
Request Submitter Notified Date	Date the submitter is notified of the request closure. Coding and Edit Rules: Format: date Length: 10	Req_Submitter_Notify_Dt
Request Submitter Priority Code	The priority determined by the request submitter. Coding and Edit Rules: Format: alpha	Req_Submitter_Priority_Cd

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

MHC-SD-GN-CSMD-30072
REVISION NUMBER 0

A-A-16

320DATA.V1

DATE: 13-JUN-94
TIME: 08:57

QUEST2 Definition of Data Elements (320)

PAGE 17
Excelerator

Name	Description	Alternate Name
	Length: 1 Permitted Values: H, M, L	
Request Type	There are two types of change requests. A request to change the functionality of the system, C, or a request to fix a bug in the system, B. Coding and Edit Rules: Format: alpha Length: 1 Permitted Values: B or C	Req_Tp
Requirements Identifier	The unique system generated key for a requirements number. Coding and edit rules: Format: numeric (6.0) Length: Permitted Values: 1 to 999999 Validation Rules: to find the next key the system will add one to the last key.	Requirement_ID
Requirements Number	The shorthand number for the law, order, procedure or manual Integrity Rules: Required for each requirements id. This will be free form text. Coding and Edit Rules: Length: 35 Format: Alphanumeric	Requirement_No
Requirements Title	The title of the orders, laws or procedures. Integrity Rules: Required. Coding and Edit Rules: Format: alphanumeric Length: 70	Requirement_Ttl
Response Approved Date	The date the originator approved the response. Coding and Edit Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: Not Required. Not allowed unless response received. Permitted Values: Nothing before the Response Received Date.	Resp_Approved_Dt
Response Discussion Text	Narrative text responding to the initiating document, usually outlining an action plan. Entered from a word processor this may have codes displayable only by the word processor. Coding and Edit Rules: Format: text Length: 2K minimum Validation: Required to submit a response.	Resp_Discussion_Txt
Response Document Identifier	System generated unique key for an initiating document. This will be visible to the user for queries. Coding and Edit Rules: Format: This may be the assessment or condition number Length:	Resp_Doc_Id

A-A-17

330DATA.V1

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

MHC-SD-GN-CSWD-30072
REVISION NUMBER 0

DATE: 13-JUN-94
TIME: 08:57

QUEST2 Definition of Data Elements (320)

PAGE 18
Excelerator

Name	Description	Alternate Name
Response Due Date	The date the owner must respond to the originator. If the date is not met the status will go delinquent. Coding and Edit Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: Not Required. Permitted Values: Nothing before the Initiating Date. Default: 30 days from the Initiating Date.	Resp_Due_Dt
Response Received Date	The date the originator received the response. When this date is added it may change a delinquent for response back to open. Coding and Edit Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: Not Required. Permitted Values: Nothing before the Response Due Date.	Resp_Received_Dt
Response Rejected Date	The date the response was rejected. Coding and Edit Rules: Format: Date, MM/DD/YYYY Length: 10 Validation: Required for rejection. Permitted Values: Not before response received.	Resp_Rejected_Dt
Response Rejection Text	Narrative text explaining why a response was rejected. Entered from a word processor; this may have codes displayable only by the word processor. Coding and Edit Rules: Format: text Length: 2K minimum Validation: Required to reject a response.	Resp_Rejection_Txt
Response Revised Due Date Count	The number of times the response due date has been changed. Coding and Editing Rules: Format: Integer Length: two digits Validation: Start at zero for the first due date. Each time the due date changes, add one.	Resp_Rev_Due_Date_Cnt
Role Identifier	The system generated unique id for each user role. Coding and Edit Rules: Format: numeric (6.0) Length: Permitted Values: 1 to 999,999	Role_Id
Role Title	The title for each quest2 role. Integrity Rules: The OWNER role is not available from a pick list. Every system user is considered to have the owner role. Required for each role. Coding and Edit Rules: Format: alphanumeric Length: 20 Permitted Values: DBA, Assessor, System Administrator, General User, Non-Cams User, Non-Cams Assessor, Owner.	Role_Ttl

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

MHC-SD-6N-CSMD-30072
REVISION NUMBER 0

A-A-18

320DATA.V1

DATE: 13-JUN-94
TIME: 08:57

QUEST2 Definition of Data Elements (320)

PAGE 19
Excalerator

Name	Description	Alternate Name
Run Log Date	The date and time the production job was run. Coding and Edit Rules: Format: datetime Length: 15 Validation: this will be a system time stamp at runtime.	Run_Log_Dt
Run Log Identifier	The system generated unique for each production run. (key) Coding and Edit Rules: Format: numeric (6.0) Length: Permitted Values: 1 to 999,999	Run_Log_Id
Run Log Results Description	A short description of the results of the production run. This could be error messages or an ok status. Coding and Edit Rules: Format: alphanumeric Length: 250 Validation: this may be added by the system.	Run_Log_Results_Desc
Run Log Results File Name	The path name to the production run log itself. Coding and Edit Rules: Format: alphanumeric Length: 50 Validation: System defined, could be overridden by the administrator	Run_Log_Results_File_Nm
Status Code	Denotes the state of an I.D., condition or action. (key id) Coding and Edit Rules: Format: alphanumeric Length: 3 Validation: The codes will be assigned by the system based on date fields and functions performed by users. Permitted Values:	Status_Cd
Status Description	A full text description of a status code. Integrity Rules: Required for each status code. Coding and Edit Rules: Format: alphanumeric Length: 25	Status_Desc
Subject Code	Subject Codes categorize initiating documents and conditions based on a broad definition of the scope is safety, administrative or environmental. Coding and Edit Rules: Format: alphanumeric Length: 5 Validation: based on a control table Permitted Values: include, Safety, Environmental and Safety.	Subject_Code
Subject Description	A text description explaining the meaning of a subject code. Integrity Rules: Required for each subject code. Coding and Edit Rules: Format: alphanumeric	Subject_Desc

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WHC-SD-GM-CSND-30072
REVISION NUMBER 0

A-A-19

320DATA.V1

SA/RELEASE1 DELIVERABLE SET
ATTACHMENT 1
DEFINITION OF DATA ELEMENTS

WHC-SO-GN-CSND-30072
REVISION NUMBER 0

PAGE 20
Excelerator

QUEST2 Definition of Data Elements (320)

DATE: 13-JUN-94
TIME: 08:57

Name	Description	Alternate Name
User Identifier	Length: 75 The system generated unique id for each user. Coding and Edit Rules: Format: numeric (6.0) Length: Permitted Values: 1 to 999,999	User_Id
User Password Description	The users password into quest2. Integrity Rules: Required for each user. Coding and Edit Rules: Format: alphanumeric Length: 30 Permitted Values: Ideally words not in a dictionary.	User_Password_Desc
Validation Rule Field Name	The name of the database field for the validation rule. Integrity Rules: Required for each validation rule. Coding and Edit Rules: Format: alphanumeric Length: 30	Rule_Field_Nm
Validation Rule Identifier	The system generated unique id for each validation rule. Coding and Edit Rules: Format: numeric (6.0) Length: Permitted Values: 1 to 999,999	Rule_Id
Validation Rule Text	The actual text of the validation rule. This will be substituted into the programs and executed. Integrity Rules: Required for each validation rule. Coding and Edit Rules: Format: alphanumeric Length: 250	Rule_Txt
Validation Rule Title	The title is a brief description of the validation rule. Integrity Rules: Required for each validation rule. Coding and Edit Rules: Format: alphanumeric Length: 70	Rule_Ttl
View Code	This code marks a condition as part of a logical partition. Coding and editing Rules: Format: alphanumeric Length: 10	View_Cd
View Description	A free form text description of the view code. Integrity Rules: Required for each view code. Coding and Edit Rules: Format: alphanumeric Length: 70	View_Desc

1.0 PURPOSE

This document describes the state of the data of the existing systems, rules for converting that data to QUEST2, and the activities required to implement a successful conversion. This document will provide the users with the information required to determine the scale of work involved in the conversion process, and to understand outstanding data issues that must be resolved.

2.0 FUNCTIONAL STANDARDS

This document has been written during the System Architecture Phase of the QUEST2 development life cycle. Because this phase defines a "Conceptual" model of the business data, data elements required by QUEST2 may still need to be defined. Many relationships in the QUEST2 Conceptual Data Model will become data entities or data elements when the Functional Data Model is created during Functional Design. Consequently, the Conversion Rules section of this document will show existing system data elements converting to relationships rather than simply to their corresponding QUEST2 data element. This perspective is provided during this phase of the project in order to validate the data model before starting the functional design.

Each of the following sections describes the data in terms of the QUEST2 facets, defined in the Conceptual Data Model, even though the existing systems are not documented in this manner:

State of the data of the existing system: There are actually two systems with data that will merge into QUEST2: the existing QUEST system and the DOE-RL CICS system. This section will explain how the systems are structured, their scale, and any large system issues.

Conversion Rules: This section will map the existing business data elements to the corresponding QUEST2 data elements or relationships.

Conversion Activities: This section will describe the tasks, unit processes and any recommendations for proceeding with data conversion.

Data Model Validations: This section contains a discussion of any integrity rules, relationships or data elements issues that might require modifications to the Conceptual Data Model.

QUEST: SA/RELEASE1 SUPPORTING DOCUMENTS DELIVERABLE SET	Document Appendix Revision Number	WHC-SO-GN-CSWD-30072 8 0
CONVERSION RULES (P+ DOCUMENT #340)	Page Effective Date	2 of 20 May 31, 1994

3.0 STATE OF THE DATA OF THE EXISTING SYSTEM

3.1 Condition Data Facet

3.1.a Current QUEST System.

The current QUEST system data base is stored in a NOMAD application on an IBM mainframe. Performance considerations dictated that condition data be stored in two physically separate databases, one for data entry and updates (the transaction data base), the other for queries and reporting (the certified data base). Condition data exists simultaneously in both databases. The structure of the transaction data base is identical to the certified data base. Conversion rules and activities will be specified to merge these two databases into one QUEST2 database.

Existing QUEST data dates back to 1987. There are approximately 40,000 conditions, 30,000 actions and 1700 Source documents. The actions count column in the table below is the number of actions associated with conditions closed in the specified year (not the actions actually closed in that year).

Count of QUEST Conditions and
Actions by Condition Closed Date

Count of QUEST
Source Documents

DEFERRED/ OPEN/ YEAR CLOSED	CONDITIONS COUNT	ACTIONS COUNT	TOTAL COUNT	SOURCE INITIATED DATE	TOTAL COUNT
DEFERRED	152	171	323	OTHER	95
OPEN	3,610	4,658	8,268	86	5
87	2,294	553	2,847	87	33
88	5,526	2,691	8,217	88	83
89	5,910	4,046	9,956	89	187
90	5,575	4,113	9,688	90	166
91	5,865	4,082	9,947	91	349
92	6,289	3,711	10,000	92	472
93	5,316	4,032	9,348	93	260
94	1,941	1,258	3,199	94	128
	=====	=====	=====		=====
	42,478	29,315	71,793		1,778

The existing QUEST system has many data elements that are a person's name. Even though the names are entered in a consistent manner they are not validated against any reference table. QUEST2 will use downloads from PeopleCORE and PeopleCORE's unique ID numbers, as well as names. A conversion activity will involve determining the PeopleCORE ID number for existing QUEST names.

The existing QUEST system has defined and maintained data relating to buildings, facilities and other locations. Not all of the data is validated with a reference table. QUEST2 will use uploads from FacilityCORE to maintain referential integrity. The existing QUEST definition of facility is not used by FacilityCORE and will be converted to QUEST2. Buildings and other locations are stored in FacilityCORE. Validating this data with existing QUEST data is considered a conversion activity.

QUEST contains extensive history information dating back to 1988. History data differs from the data to be archived, discussed above, because it contains the changes made to each condition and action as they moved from open to closed status. This data exists in two forms: tapes of monthly backups and online tables. How much of this data that is to be converted and how much is to be archived remains an outstanding issue.

3.1.b The DOE-RL CICS System.

The DOE-RL CICS system is a Paradox application consisting of several independent tables. CICS includes data uploaded from PNL, HEHF and the existing QUEST system, as well as data specifically for DOE. Some information is replicated in both CICS and the existing QUEST. The conversion specifications will address data that is converted from CICS and data that is simply used directly from QUEST for a particular record. CICS has no validation of people names. In fact, sometimes several names are placed in the same data element.

The DOE-RL CICS system has approximately 3,000 initiating documents, 8,000 conditions and 1,000 actions. How much of this data that will be converted to QUEST2 and how much will be moved to an archive remains an issue.

3.2 Analysis Data Facet

There is no Analysis Data in the existing QUEST or CICS systems. Therefore, no conversion will be necessary.

3.3 System Administration Facet

The existing QUEST system contains user information about all the past and current users. Aside from the user information, the existing QUEST system administration data exists to provide security. Existing QUEST security data will not need to be converted because it will not apply to the QUEST2 application and platform.

The CICS system has only two online users and they maintain the database. CICS has no system administration data to convert.

3.4 Change Control Facet

Change control data in the existing QUEST system consists of paper forms stored in log books. This data is administered manually by the IRM QUEST system administrator. It will not be converted to QUEST2 which will commence its own conversion control management when it is implemented.

4.0 CONVERSION RULES

4.1 Condition Data Facet

4.1.a Data Elements/Entities

The following table contains the data element mapping from the existing QUEST and CICS data bases into the QUEST2 data base.

The data elements listed for the existing QUEST are from the certified data base. If the same record also exists in the transaction data base, the transaction data element should be used.

There are times when the source data elements exist in both QUEST and CICS. When the items have been originated by DOE-RL, or are Tiger Team and PAT conditions and actions, then only CICS data elements will be used. Under the Validation Derivation column the phrase "Convert as is CICS has priority" will be listed. At times both QUEST and CICS values will be converted to QUEST2, particularly for text descriptions. Since the current QUEST system downloads Tiger and PAT action data, the existing QUEST data elements will have priority for some data elements.

QUEST2 Data Element Name	Existing QUEST/CICS Data Element Name	Validation Derivation
Action Abstract Text	n/a	
Action Comments Date	QUEST: CTXT LUPDT	DO1
Action Comments Text	QUEST: CTXT_TEXT	DO1
Action Completed Date	QUEST: CACT CMP_DTE CICS: CAC DTE	Convert as is CICS has priority
Action Description Text	QUEST: CTXT_TEXT CICS: Corrective Action	DO1
Action Due Date	QUEST: CACT COMMIT_DTE CICS: P COMP DTE	Convert as is QUEST has priority

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET

CONVERSION RULES
(P+ DOCUMENT #340)

Document
Appendix
Revision Number
Page
Effective Date

WHC-SD-GN-CSMD-30072
B
0
5 of 20
May 31, 1994

QUEST2 Data Element Name	Existing QUEST/CICS Data Element Name	Validation Derivation
Action Identifier	System generated	
Action Ref Type	user defined	
Action Ref Type Alias Indicator	user defined	
Action Ref Type Title	user defined	
Action Reference Identifier	system generated	
Action Reference Number	QUEST: CACT ACTION_ID CICS: Reference Report #	Include both
Action Rejected Date	QUEST: CACT REJ DTE	convert as is
Action Revised Due Date Count	QUEST: CACT COM REV	Convert as is
Action Title	QUEST: CACT ACT TITLE	Convert as is
Assessment Abstract Text	n/a	
Assessment Comments Date	n/a	
Assessment Comments Text	n/a	
Assessment Conducted Date	CICS: COND DTE	Convert as is
Assessment Discovery Date	QUEST: CSUB OC_DSC_DT CICS: DSC DTE	Convert as is CICS has priority
Assessment Identifier	System generated	
Assessment Initiated Date	QUEST: CSUB INIT_DTE CICS: INIT DTE	Convert as is CICS has priority
Assessment Introduction Text	n/a	
Assessment Scheduled Date	CICS: SCHEDULED	Convert as is
Assessment Summary Text	n/a	
Assessment Title	n/a	
Cond Revised Completion Dt Cnt	QUEST: CSUB COM_REV CICS: PCD REV	Convert as is CICS has priority

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET

CONVERSION RULES
(P+ DOCUMENT #340)

Document
Appendix
Revision Number
Page
Effective Date

WMC-SD-6N-CSWD-30072
B
0
6 of 20
May 31, 1994

QUEST2 Data Element Name	Existing QUEST/CICS Data Element Name	Validation Derivation
Condition Abstract Text	n/a	
Condition Actions Completed Date	QUEST: CSUB CMP_DTE CICS: CAC DTE	Convert as is CICS has priority
Condition Closed Date	QUEST: CSUB CLOS_DTE CICS: CLOS DTE	Convert as is CICS has priority
Condition Comments Date	QUEST: CTXT_LUPDT	D01
Condition Comments Text	QUEST: CTXT_TEXT	D01
Condition Description Text	QUEST: CTXT_TEXT	Convert as is, D01
Condition Equipment Description	QUEST: CSP2_EQUIP	Convert as is
Condition First Tag Code	n/a	
Condition Fourth Tag Code	n/a	
Condition Identifier	System generated	
Condition Location Description	QUEST: Multiple CICS: FAC	D02
Condition Area Code	QUEST: CSUB_AREA CICS: AREA	D02
Condition Room Number	QUEST: CSUB_PLACE CSUB_SUB_FAC CICS: FAC	D02
Condition Memo Text	QUEST: CSP2_DBA_MEMO	Convert as is
Condition Planned Completion Dt	QUEST: CSUB COMMIT_DTE CICS: P_COMP_DTE	Convert as is QUEST has priority for TTA, PAT only

QUEST2 Data Element Name	Existing QUEST/CICS Data Element Name	Validation Derivation
Condition Ref Type	User defined	
Condition Ref Type Alias Ind	User defined	
Condition Ref Type Description	User defined	
Condition Reference Identifier	System generated	
Condition Reference Number	QUEST: CSUB_SUBJECT_ID CICS: Multiple	Convert as is D03
Condition Revised Completion Dt	QUEST: CSUB_COMMIT_DTE CICS: P COMP DTE	Convert as is D04
Condition Second Tag Code	n/a	
Condition Third Tag Code	n/a	
Condition Title	QUEST: CSUB_SUBJECT	Convert as is
Condition Type	User defined	D05
Condition Type Description	User defined	
Condition Verified Date	QUEST: CSUB_VERIF_DTE	Convert as is
Contractor Identifier	User defined	
Distribution List Identifier	System generated	
Document Class Description	QUEST: CDTYP_CLASS CICS: User Defined	Convert as is
Document Class Type	QUEST: CDTYP_CLASS_CDE CICS: n/a	Convert as is

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET

CONVERSION RULES
(P+ DOCUMENT #340)

Document
Appendix
Revision Number
Page
Effective Date

WMC-SD-GN-CSMD-30072
8
0
8 of 20
May 31, 1994

QUEST2 Data Element Name	Existing QUEST/CICS Data Element Name	Validation Derivation
Document Ref Type	User defined	
Document Ref Type Alias Ind	User defined	
Document Ref Type Description	User defined	
Document Reference Identifier	System generated	
Document Reference Number	QUEST: CSUB_SOURCE_ID CICS: Multiple	Convert as is D06
Good Practice Description	n/a	
Good Practice Identifier	System generated	D13
Good Practice Title	n/a	D13
Intangible Code	User defined	
Intangible Description	User defined	
Intangible Type	User defined	
Intangible Type Description	User defined	
PPG As Found Indicator	n/a	D07
PPG Assumption Text	n/a	
PPG DOE Mission Prob. Value	n/a	
PPG Dom Cons Probability Value	n/a	

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET

CONVERSION RULES
(P+ DOCUMENT #340)

Document
Appendix
Revision Number
Page
Effective Date

MHC-SD-GN-CSWD-30072
B
0
9 of 20
May 31, 1994

QUEST2 Data Element Name	Existing QUEST/CICS Data Element Name	Validation Derivation
PPG Dom Cons Risk Value	n/a	
PPG Dominating Consequence Cd	QUEST: CSUB PPG_DI CICS: PPG DI	CICS has priority
PPG Env. Protection Value	n/a	
PPG Evaluation Date	n/a	
PPG Evaluation Team Text	n/a	
PPG External Reaction Value	n/a	
PPG Identifier	System generated	
PPG Investment Protection Value	n/a	
PPG Public Safety Value	n/a	
PPG Value	QUEST: CSUB PPG_VAL CICS: PPG_VAL	Convert as is CICS has priority
PPG Worker Safety Value	n/a	
RCA Contributing Cause Code	QUEST: CDOE CAU_CODE CICS: S RC CODE	D08
RCA Direct Cause Code	QUEST: CDOE CAU_CODE CICS: P RC CODE	D08
RCA Identifier	System generated	
RCA Root Cause Code	QUEST: CDOE CAU_CODE CICS: N/A	D08
Requirements Identifier	System generated	

QUEST2 Data Element Name	Existing QUEST/CICS Data Element Name	Validation Derivation
Requirements Number	CICS: REQ	D09
Requirements Title	User defined	
Response Approved Date	QUEST: CSUB_AP_APP_DTE	Convert as is
Response Discussion Text	QUEST: CTXT_TEXT	D10
Response Document Identifier	System generated	
Response Due Date	QUEST: CSUB_R DUE_DTE CICS: R DUE_DTE	Convert as is QUEST has priority
Response Received Date	QUEST: CSUB_R REC_DTE CICS: R REC_DTE	Convert as is QUEST has priority
Response Rejected Date	QUEST: CACT_REJ_DTE	D11
Response Rejection Text	QUEST: CTXT_TEXT	D10
Response Revised Due Date Count	CICS: RESP_REV	Convert as is
Status Code	User defined	D12
Status Description	User defined	
Subject Code	User defined	
Subject Description	User defined	

4.1.b Validations and Derivations

D01 This applies to both comments and description text for actions and conditions. All of these attributes are stored in the same field in the existing QUEST system. A text type code determines what they are.

All of these text fields in the existing QUEST are unlimited in length. In QUEST2 all Comments Text fields are 250 characters. CTEXT_TEXT fields longer than 250 characters will be entered manually.

Where CICS and QUEST have description text, both text fields will be converted to QUEST2. This applies to DOE-RL originated, Tiger Team and PAT items.

- D02 The Condition Location Description is a combination of the existing QUEST data elements CSUB PLACE and CSUB SUB FAC. These are both free form fields with no validating reference table. CICS stores several types of location information in the FAC data element. Some FAC data will be facilities, buildings or some other location.

The QUEST2 room numbers are stored in the CSUB PLACE and CSUB SUB FAC fields. This data must be reviewed before conversion. FacilityCORE does not contain room numbers for each building. It may be possible, by reviewing existing QUEST data, to establish a reference table associating room numbers to buildings.

The area codes used by QUEST and CICS must be cross checked with the FacilityCORE area codes before conversion.

- D03 Condition Reference Numbers are numbers assigned to conditions from other systems, or a number referring to some other pertinent document. The existing QUEST condition identifier will be a reference number in QUEST2.

CICS contains many different types of reference numbers including PNL identifiers, HEHF identifiers or DOE identifiers. Each of these numbers will be stored in QUEST2 but they have a different Condition Reference Type.

- D04 The Condition Revised Completion Date is not stored in either QUEST or CICS however, it may be determined by examining the Cond Revised Completion Dt Cnt. If this number indicates a revision has occurred, the Condition Revised Completion Date is the CSUB_COMMIT_DTE or P_COMP_DTE, when CICS has priority.
- D05 Condition Types are not defined yet. This is an outstanding issue.
- D06 Document Reference Numbers are numbers assigned to Initiating Documents (assessments), by other systems. QUEST has one number, but CICS may have several. All of these numbers must be converted to QUEST2.
- D07 PPG As Found Indicator is not stored by QUEST or CICS. Each of these systems contains the "PPG as is value" and the "PPG as found value".

This data element should be set depending on which PPG value is converted.

- D08 Root Cause Codes: CICS uses the fields S RC CODE and P RC CODE. It is not clear if these are the Direct, Contributing or Root cause codes.
- D09 Requirements Numbers are free form text in both QUEST and CICS. However, the data in the CICS has been entered consistently so it can be used to establish a validation table. Several requirement numbers are entered in the REQ data element. They must be split out. The QUEST requirements have been entered haphazardly and should be reviewed before entry into QUEST2.
- D10 Response Discussion Text has been entered into the existing QUEST condition comments text element CTXT_TEXT. These must be reviewed manually before entry into QUEST2.
- D11 Response Rejected Dates may have been entered into the CACT_REJ_DTE for each QUEST action listed in the action plan. This field may have also been used to reject actions because they did not fix the condition. This data needs to be reviewed before added to QUEST2.
- D12 Status codes have been defined but they do not include the CICS status codes at this time. This remains an outstanding issue. CICS applies a closed status to Tiger Team and PAT actions and an NRR, Not Required to Respond, status to conditions with PPG < 11. Neither of these is used in QUEST2. The NRR status could be converted to a standard QUEST2 status and the need fulfilled by a standard query.

4.1.c Data Described as Relationships

Below are the conversion rules for data described as relationships on the conceptual data model. The data elements are the primary keys of the related entities. For data concerning initiating documents, the CICS system should take priority over the existing QUEST system with the exception of data currently downloaded to the Hanford Self Assessment Data Base (HSADB).

QUEST2 Relationship Name	QUEST/CICS Data Element Name	Validation Derivation
Lead Assessor for Init Doc (REQ1149)	QUEST: CSUB_LEAD_AUD CICS: support_person	D13
Lead Assessor Org for Init Doc (REQ1105)	QUEST: CSUB_AUD_ORG CICS: n/a	D14

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET

CONVERSION RULES
(P+ DOCUMENT #340)

Document
Appendix
Revision Number
Page
Effective Date

MHC-SD-GN-CSWD-30072
B
0
13 of 20
May 31, 1994

QUEST2 Relationship Name	QUEST/CICS Data Element Name	Validation Derivation
Person Responsible for Init Doc (RE01150)	QUEST: CSUB_RESP_PRSN CICS: RESP_PRSN	D13
Responsible Org for Init Doc (RE01150)	QUEST: CSUB_RESP_ORG CICS: n/a	D14
Originator for Init Doc (RE01151)	QUEST: CSUB_ORIGNR CICS: ORIGNR	D13
Originator org for Init Doc (RE01108)	QUEST: CSUB_ORIG_ORG CICS: n/a	D14
Init Doc has RL-Div/Branch (RE01126)	QUEST: n/a CICS: RL LINE, RL BRNCH	D14
Contractor owns Init Doc (RE01160)	QUEST: CSUB_ORIG_CONTR CICS: ORIG_CONTR	Convert as is
Init Doc assesses Contractor (RE01143)	QUEST: CSUB_RESP_CONTR CICS: RESP_CONTR	Convert as is
Init Doc is of a Doc Class (RE01101)	QUEST: CSUB_DTYP_CLCDE CICS: DOC_OC_TYPE	Convert as is
Init Doc has Status (RE01173)	n/a	D15
Init Doc assesses Facility (RE01140)	QUEST: CSUB_FAC CICS: FAC	D16
Init Doc assesses Intangible Items (RE01139)	QUEST: CSUB_ASMT_CAT CICS: Activity	D17
Requirements Drive Init Doc (RE01148)	QUEST: n/a CICS: REQ	D18
Identifier for Condition (RE01107)	QUEST: CSUB_ORIGNR CICS: support person	D13
Identifier Org for Condition (RE01168)	QUEST: CSUB_ORIG_ORG CICS: support person	D14
Owner of Condition (RE01109)	QUEST: CSUB_RESP_PRSN CICS: RESP_PRSN	D13
Owner Org for Condition (RE01137)	QUEST: CSUB_RESP_ORG CICS: n/a	D14

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET

CONVERSION RULES
(P+ DOCUMENT #340)

Document
Appendix
Revision Number
Page
Effective Date

WHC-SD-GN-C5WD-30072
B
0
14 of 20
May 31, 1994

QUEST2 Relationship Name	QUEST/CICS Data Element Name	Validation Derivation
Monitor for Condition (RE01130)	QUEST: CSUB_DOE_RL_MON CICS: support person	D13
Monitor org for Condition (RE01122)	QUEST: CSUB_DOE_RL_ORG CICS: n/a	D14
Closure Authority for Condition (RE01110)	QUEST: CSUB_CLOS_PRSN CICS: support person	D13
Closure Authority org for Condition (RE01153)	QUEST: CSUB_CLAUTH_ORG CICS: n/a	D14
Verifier for Condition (RE01136)	QUEST: CSUB_VERIF_PRSN CICS: n/a	D13
Verifier org for Condition (RE01138)	QUEST: CSUB_VERIF_ORG CICS: n/a	D14
Condition is for Facility (RE01119)	QUEST: CSUB_FAC CICS: FAC	D16
Condition is for Building (RE01112)	QUEST: CSUB_BLDG CICS: FAC	D16
Requirements violated condition (RE01137)	QUEST: multiple CICS: REQ	D18
Condition is for Intangible Items (RE01170)	QUEST: CSUB_ASMT_CAT CICS: Activity	D17
Condition has Status (RE01129) Elements: Status Code Status Change Date Status Date Condition Identifier	QUEST: CSUB_ADM_SS_IDF SH_ADM_SS_IDF SH_CHANGE_DT SH_STATDEF_DTE CICS: CURR_STAT	D15
Owner for Action (RE01121)	QUEST: CSUB_ACTIONEE CICS: RESP_PRSN	D14
Owner org for Action (RE01154)	QUEST: CSUB_ACT_ORG CICS: n/a	D14
Contractor Performs Action (RE01164)	QUEST: CACT_ACT_CONTR CICS: RESP_CONTR	Convert as is

QUEST2 Relationship Name	QUEST/CICS Data Element Name	Validation Derivation
Action has Status (RE01120) Elements: Status Code Status Change Date Status Date Action Identifier	QUEST: CSUB ADM SS IDF SH ADM SS IDF SH CHANGE_DT SH STATDEF_DTE CICS: CURR STAT	D15

4.1.d Derivations and Validations for Relationship Type Data Elements

D13 The existing systems use names for all personnel data. These must be converted to the PeopleCORE unique person ID. The CICS support person field contains many different types of persons. These must be reviewed during the preparation of conversion programs and data files.

If people in CICS or QUEST are not listed in PeopleCORE, QUEST2 will assign a unique system number. They will be marked that they do not exist in PeopleCore so the data is not changed during periodic uploads.

D14 The org codes used by the existing QUEST system are the Organization Cost Codes used in PeopleCORE. The CICS org code is represented by relationships with RL-Divisions and Branches. The existing QUEST system uses org codes to represent other agencies and organizations within the DOE. It should be verified that these exist in PeopleCORE. The CICS has a table of the correct RL-Divisions and Branches. This table will be converted to QUEST2. The appropriate RL-Division and Branch will be converted with corresponding DOE person. In those rare cases where a code does not exist, users will have to manually attain a code agreed to by the business administration processes and enter it.

D15 Initiating Documents have no status identifiers in the existing systems. It is recommended that the status be derived from the dates that are converted to initiating documents, in combination with the status of the associated conditions.

Conditions and actions have a history of each change in status. The entire history of status changes must be converted to QUEST2

D16 The existing systems contain no information relating facilities to initiating documents. The recommendation is to assign the facility of each condition to the initiating document. The existing QUEST and CICS systems only allow one facility per condition.

**QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET**

**CONVERSION RULES
(P+ DOCUMENT #340)**

Document	WMC-SD-GN-CSWD-30072
Appendix	8
Revision Number	0
Page	16 of 20
Effective Date	May 31, 1994

The Facility codes used by the QUEST system are not used in FacilityCORE. With the exception of facility codes designating non-facility areas and "other", the entire QUEST facility table should be converted to QUEST2. The removal of these codes will be acceptable because QUEST2 will not require entry of the facility data.

The FAC field in CICS contains building and other facility codes. These must be reviewed and converted to the correct QUEST2 fields.

Building codes in QUEST are validated by a reference table, however, the codes in the table may not correspond to the codes in FacilityCORE. These should be cross-checked with a FacilityCORE download before conversion. Some CICS FAC data will not be buildings.

- D17** Intangible Items are not firmly defined in CICS or QUEST2. The definition of valid Activities and Programs, Intangible Types, is an outstanding issue. The existing QUEST assessment categories provide an excellent source for resolving this issue. Once resolved, the assessment categories will be a good source for conversion. The intangible items associated with conditions will be converted to their initiating document, in the same manner as facility codes. The activity field in CICS should also be reviewed when this issue is resolved.
- D18** Requirements are included in QUEST2 in five different fields; CSUB_REQ_DOEORD (DOE orders); CSUB_REQ_MANUAL (Manual numbers); CSUB_REQ_OTHER; CSUB_REQ_REG (regulations); CSUB_REQ_STNDRD (standards). These are free form text fields 50 characters long. Just as in the CICS REQ field, each QUEST field may contain several requirement items. However, unlike CICS the QUEST requirements were not entered in a consistent manner. The recommendation is to use the CICS requirements to populate a reference table and to review the existing QUEST requirements before conversion. This helps meet the objectives related to data consistency and integrity.

4.2 Analysis Data Facet. There is no Analysis Data in the existing QUEST or CICS systems. Therefore, no conversion will be necessary.

4.3 System Administration Facet

4.3.a Data Elements/Entities

These are shown with the same mapping strategy as previously described for the Condition Facet.

QUEST2 Data Element Name	QUEST/CICS Data Element Name	Validation Derivation
-----------------------------	---------------------------------	--------------------------

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET

CONVERSION RULES
(P+ DOCUMENT #340)

Document
Appendix
Revision Number
Page
Effective Date

WHC-SD-GN-CSWD-30072
B
0
17 of 20
May 31, 1994

Help Text	User Input	QUEST2 User Guide, WHC, DOE manuals
Help Text Identifier	system generated	
Help Text Type	User Input	
Job Frequency Code	n/a	
Job Identifier	n/a	
Job Run Date	n/a	
Job Title	n/a	
Message Identifier	n/a	
Message Text	n/a	
Message Type	n/a	
Notice Criteria Description	n/a	
Notice Text	n/a	
Notice Title	n/a	
Notice Type	n/a	
Person CEmail Name	n/a	
Query Description	n/a	
Query Identifier	n/a	
Query Text	n/a	
Query Title	n/a	
Role Identifier	n/a	
Role Title	n/a	
Run Log Date	n/a	
Run Log Identifier	n/a	
Run Log Results Description	n/a	
Run Log Results File Name	n/a	
User Identifier	QUEST: USR USERID	D019
User Password Description	System generated	

**QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET**

Document
Appendix
Revision Number
Page
Effective Date

WHC-SD-GN-CSWD-30072
B
0
18 of 20
May 31, 1994

Validation Rule Field Name	n/a	
Validation Rule Identifier	n/a	
Validation Rule Text	n/a	
Validation Rule Title	n/a	
View Code	QUEST: CAPTS AP TS ID	D020
View Description	QUEST: CAPTS AP TS TTL	D020

4.3.b Validation and Derivations

- D019** The QUEST system contains all users who have ever had access to the system. The only ones that need to be converted are those using the first release. The QUEST user ID contains the user's payroll number. This must be converted to the PeopleCORE person ID. The CICS system has no user table.
- D020** The existing QUEST system has view codes applying to source documents. QUEST2 view codes apply to conditions. There are conditions in QUEST considered to be in the WHC-CAMS view that will be non-cams in QUEST2. These must be identified. Views for the CICS data will have to be defined.

4.4 Change Control Facet

Existing system change requests that apply to QUEST2 will be entered manually using unit process UPQSTB1 Request System Change after it has been implemented. No conversion will be necessary.

5.0 DESCRIPTION OF CONVERSION ACTIVITIES:

5.1 Condition Data Facet:

Activities to convert condition data include the following:

- Obtain downloads from PeopleCore and FacilityCore and create working reference tables.
- Using existing QUEST, CICS and PeopleCore names to determine the PeopleCORE ID.
- Verify existing system area codes, facilities, buildings, rooms, contractors and agencies with FacilityCORE data.
- Move archivable data to the QUEST2 archive

- Prepare upload programs and files from the existing systems.
- Convert existing closed conditions once physical databases are stable. This should begin as soon as possible because this is the largest group of records and they are not likely to change. It will also allow some practice before converting the open records.
- Remove conflicts between QUEST transaction and certified tables. This can be done by certifying records at least a day before the conversion.
- Review existing QUEST text comment fields. If the comments text are less than 250 characters, then convert as is; if the records are closed and the comments are greater than 250, then divide into 250 character fields and convert. Open records greater than 250 characters should be reviewed manually to identify response text or other logical events in the history of the record.
- Convert CICS DOE Division and Branch tables into a QUEST2 reference table.
- Convert CICS Requirements field to QUEST2 reference table.
- Resolve CICS names with PeopleCORE. The definition of the CICS names held in some CICS fields varies among the different tables. The CICS administrators should resolve the differences so conversion can be performed automatically.
- Convert closed CICS data overriding QUEST where applicable

5.2 Analysis Data Facet

There is no Analysis Data in the existing QUEST or CICS systems, so no conversion will be necessary.

5.3 System Administration Facet

- Identify users of the first release
- Plan to update hardware or software if necessary
- Determine the users' IDs from PeopleCORE
- Convert users to the system
- Convert help text

5.4 Change Control Facet

QUEST: SA/RELEASE1 SUPPORTING DOCUMENTS DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-6N-CSWD-30072 B 0
CONVERSION RULES (P+ DOCUMENT #340)	Page Effective Date	20 of 20 May 31, 1994

Any applicable existing system change requests will be entered manually using unit process UPQST81 Request System Change after it has been implemented. No conversion will be necessary.

6.0 DATA MODEL VALIDATION CONSIDERATIONS

Several relationships between Initiating Documents, persons, and organizations are required, but there may not be any data from the existing system for conversion. WHC organizations have lead assessors (RE01105), originator (RE01108), and are responsible for initiating documents (RE01108). In reality DOE divisions or branches perform these roles and, unless PeopleCORE has organization codes that directly correspond to the DOE division and branches, this data will not be available. A WHC organization OR a DOE division/branch should be required. Another consideration is if these will be required for non-CAMS items.

This validation process has revealed two attributes required in QUEST2 that have not been included in the SA version of the Definition of Data Elements (320): The Condition Area Code and Condition Room Number will be added during the Functional Design Phase when the functional data model is developed.

QUEST2: SA/RELEASE1 SUPPORTING DOCUMENTS DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-GN-CSWD-30072 C 0
SYSTEM PERFORMANCE CRITERIA (P+ DOCUMENT #350)	Page Effective Date	1 of 5 May 31, 1994

1.0 PURPOSE

The System Performance Criteria (350) document provides a mechanism for the System Manager to evaluate how the QUEST2 system fulfilled system objectives described during the Preliminary Analysis/System Architecture (PA/SA) phase.

2.0 OPERATIONAL PERFORMANCE CRITERIA

- 2.1 **Objective #1** Provide a single and simple point of access for CAMS documents and data.

The QUEST2 system will store CAMS related data in one database. Original assessment data, PPG and RCA data and closure data will be stored and accessible in one central database.

- 2.2 **Objective #2** Provide the user with tools that will facilitate analyses such as RCA, PPG, LLE, and Trend Analysis.

The second and third releases, if approved, will support objective #2. RCA and associated reports will be supported by integrating a proven 3rd party software package such as "REASONS" by Decision Systems. The system will likewise support LLE and Trend Analysis by making QUEST2 data available on the SHARE (text search and retrieval) system. The QUEST2 system will accept and store PPG scores. QUEST2 could eventually provide an automated version of the PPG grid.

- 2.3 **Objective #3** Allow for the traceability of changes to the data.

The QUEST2 system will provide traceability for the following items:

- Condition status
- Condition owners
- Condition Point of Contact
- Organization Responsible for Condition
- Responsible Person for Initiating Document
- Comment text for Initiating Document, Condition and Action

The system will keep the original PPG score and the 'as is' PPG score.

- 2.4 **Objective #4** Provide for automated interfaces with appropriate systems, minimizing manual interfaces where possible and cost effective.

Automated interfaces: Hanford site standard cc:mail will be employed to support auto-notification feature. (See 230/240 "Basic System Concepts" in WHC-GN-SD-CSWD-20010, bullet titled "Provide Automated Interface".)

QUEST2: SA/RELEASE1 SUPPORTING DOCUMENTS DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-6N-CSWD-30072 C 0
SYSTEM PERFORMANCE CRITERIA (P+ DOCUMENT #350)	Page Effective Date	2 of 5 May 31, 1994

File transfer 'batch' interfaces:

Personnel information - PeopleCORE upload
 Building and location information - FacilityCORE upload
 Condition information upload from PNL and HEHF
 Periodic QUEST2 reports - Soft Reporting system download

- 2.5 Objective #5 Reduce yearly system maintenance costs by approximately 30%.

The QUEST2 system will be implemented in a Client/Server environment utilizing client workstations and one central database server. This will move support off the LSIS IBM mainframe. Yearly savings in maintenance costs will be approximately \$500K.

- 2.6 Objective #6 Reduce system user training and support costs per user by approximately 50 %.

The advanced technology of the GUI front-end will provide a system that is much easier to use. Users will require less training because the system will be more self-explanatory. Administrators will have access to better tools to maintain the system.

- 2.7 Objective #7 Reduce the time required to make a system software/hardware change by approximately 40-50%.

The QUEST2 system will be developed using the latest in Client/Server technology. The Client/Server model is built upon flexibility, portability and scalability. The client processing (or user interface) can be developed or modified independently of the server. In addition, the SQL DBMS can run on a variety of hardware platforms.

- 2.8 Objective #8 Support and allow for consistent applicable site data standards, data definitions and formats.

Existing data standards, definitions and formats will be utilized. Although a multitude of specialized CAMS related systems exist, very few Hanford site standards exist in this area. QUEST2 is expected to establish the site standard for much of the CAMS specific data.

- 2.9 Objective #9 Allow users to create and generate their own on-line ad-hoc queries on relevant data.

The QUEST2 system will provide a user query facility. The query tool will graphically display tables and navigation paths between tables. Users may choose columns and apply filtering, sorting and sequencing logic. Query results, once displayed on the user screen, may optionally

QUEST2: SA/RELEASE1 SUPPORTING DOCUMENTS DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-GN-CSWD-30072 C 0
SYSTEM PERFORMANCE CRITERIA (P+ DOCUMENT #350)	Page Effective Date	3 of 5 May 31, 1994

be printed or written to a system file. (See 230/240 "Basic System Concepts" bullet titled "User Queries".)

- 2.10 Objective #10** Allow immediate user notification upon an appropriate system trigger (e.g., data and/or status changes that impact a user and/or organization).

Auto notification will send messages to people on a user defined distribution list via cc:mail for these system events:

A "new" Assessment, Condition or Action
Communication surrounding the "Response" process
Condition closure.

(See 230/240 "Basic System Concepts" bullet titled "User Notification".)

- 2.11 Objective #11** Allow for an approximate 70-80% reduction in manual re-keying of CAMS data (i.e., DBA re-keying of data).

All initiating documents can be prepared by Auditors/Assessors. Integrated WordPerfect word processing support will be provided to Auditors/Assessors as part of the document-construct screens for QUEST2 data entry. As the Auditor/Assessor portion of the user community exercises their option to compile audit reports directly on the QUEST2 system, a significant decrease in DBA transcription of initiating documents will be realized.

Approximately 6,000 audits are generated each year. After implementation, use of the initiating document front end will eventually grow to achieve the 70-80% reduction in manual re-keying.

- 2.12 Objective #12** Put the corrective action operations and data at the appropriate organizational level of ownership, bypassing intermediaries where possible.

The QUEST2 system will support the CAMS business rules as defined by the Functional User Group. These business rules are reflected in the QUEST2 data models (P+ documents 170, 180, 190). System administration and data ownership can be administered in as decentralized a manner as desired.

- 2.13 Objective #13** Support non-CAMS information management by providing facilities with non-CAMS system support, as applicable.

QUEST2 system users will have the option of placing non-CAMS items on the database and tracking those items at a local level. Non-CAMS items will be secure; only those in the originating organization will be able to

QUEST2: SA/RELEASE1 SUPPORTING DOCUMENTS DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-GN-CSWD-30072 C 0
SYSTEM PERFORMANCE CRITERIA (P+ DOCUMENT #350)	Page Effective Date	4 of 5 May 31, 1994

view and manage non-CAMS items. (See 230/240 "Basic System Concepts" bullet titled "Support non-CAMS data storage and access".)

- 2.14 Objective #14 Provide input tool and report generator that allows originators to collect and report on their assessments (i.e., findings, conditions, etc.)

QUEST2 will support data collection and audit report generation (for initiating documents). (See 230/240 "Basic System Concepts" bullets titled "Initiating document preparation" and "Documents generation" in section 2.0 "Functional Concepts".)

- 2.15 Objective #15 Provide a mechanism to support system change control.

The first release will employ a manual system and associated procedures now established with the current QUEST system. If approved, the second release will automate some of these procedures, especially the change form.

3.0 TECHNICAL PERFORMANCE CRITERIA

3.1 On-line response times:

Client process (i.e., menu selection, command button, etc...) will be no more than 3 seconds 90% of the time.

Server process triggered by client request (i.e., single row database add/change/delete) will be no more than 6 seconds 90% of the time. Database ad-hoc query and standard reports (i.e., accessing multiple tables and returning many rows) will provide immediate and constant feedback to the user via a dialogue box. The dialogue box will also give the user the option to continue or cancel.

3.2 System down time

The QUEST2 system will average no more than 2 regular business hours of down time per month. A "help desk" will be available, under the direction of the System Manager, during regular business hours.

The central file server (hardware) will have 24 hour emergency maintenance support.

Note: HLAN down time is not controllable by the owner/developer of the QUEST2 applications.

QUEST2: SA/RELEASE1 SUPPORTING DOCUMENTS DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-GN-CSWD-30072 C 0
SYSTEM PERFORMANCE CRITERIA (P+ DOCUMENT #350)	Page Effective Date	5 of 5 May 31, 1994

3.3 Batch processing

Production batch reports will be available on a next-day basis. These reports include those produced for Soft Reporting and internal standard reports. A nightly batch queue will be available for user queries as well.

The criteria for archiving QUEST2 data is as follows:

- The prototyping development process and the conversion rules for QUEST2 construction will define functional specifications pertaining to archiving details. Certain record types are expected to have differing retention periods. Also, some entire records are required for certain functionality (e.g. trending analysis) and other records require only basic information for reports. Once rules are established to close records, their respective initiating documents, and all associated condition records will be retained on line for the remainder until the next archive batch process.
- The archival process will be activated at a consistent time interval e.g., January and July of each year.

3.4 Interface processing cycles

PeopleCORE upload - weekly
FacilitiesCORE upload - monthly
PNL Condition Data upload - monthly (or on demand)
HEHF Condition Data upload - monthly (or on demand)
SHARE download - weekly
Soft Reporting download - weekly
Auto-notification processing - once daily, weekdays

1.0 PURPOSE

The purpose of the Functional Standards document is:

- To specify the functional standards for the system as a whole.
- To guide the design of the Unit Process Definitions (Deliverable 490).
- To define the standards to be used for interaction with the database.
- To define the design criteria to be used for the development of the screens, reports, and functional specifications.
- To identify the functional standards and requirements for security, program navigation, communication, and functional controls.
- To identify common processes.

2.0 INTRODUCTION

Functional designers will use standards in this document as a guide in specifying what the QUEST2 system will accomplish, and how it will be perceived by the user. These standards will be used heavily during the Functional Design phase when the 490 deliverables are developed. The goal of the 360 is to ensure consistency between application interfaces for the entire system. The types of questions that will be answered include:

- What types of interface elements are available to accomplish a task?
- What elements will be present on the user's screen?
- What items will be present on a report from the system?

Custom software will be developed to provide integration between the functional areas, any supplemental "off-the-shelf" packages, and external systems.

The target platform for the system is a "Client-Server" environment with an intelligent workstation. All workstations will use the Microsoft "Windows Operating Environment" (currently version 3.1). The "Windows" environment is intended to increase operator proficiency. Proficiency is accomplished through the object-action graphical model and through the use of an integrated, consistent "look and feel" across the applications selected

QUEST: SA/RELEASE1 SUPPORTING DOCUMENT DELIVERABLE SET	Document Appendix	WHC-SD-GN-CSWD-30072 0
FUNCTIONAL STANDARDS (P+ DOCUMENT #360)	Revision Number Page	0 2 of 14
	Effective Date	May 31, 1994

by the user. The "Windows" environment has extensively defined guidelines and standards for the development and implementation of applications as discussed in paragraph 4.3.

New software will be developed together with new interfaces to accomplish specific tasks within the QUEST2 system. The QUEST2 system will require its own collection of inputs, outputs, interfaces, and reports. Specific standards for the development of this suite of interactions are the subject of this document.

To allow for knowledge gained during the software's life cycle, the 360 document should be considered a living document. As issues are identified and resolved, existing standards may require revision, and new standards will be established and recorded in this document.

3.0 DATA FLOWS

3.1 Functional Model

System and sub-system data flows are shown in the QUEST2 250 Functional Models. They describe all system "communications" between:

- processes,
- processes and data stores,
- processes and external actors, and
- processes and the user (manual processes).

Data Flows identified in the functional process models for QUEST2 (see Definition Of System Functions (250), WHC-GN-SD-CSWD-20010) represent screens, reports, views of the data (queries, creates, etc.), triggers, and other system features. These will be further specified in the Unit Process Functional Specifications (490) developed during the system construction phase of QUEST2.

3.2 External Systems - Data Exchange

QUEST2 will be an integrated data system, and will require multiple points of data exchange with other computer systems. Data exchange procedures for use with the existing systems will be developed. At this writing, these interfaces are not established. Time has been allotted during the Functional Design Phase of each release to specify these interfaces.

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENT DELIVERABLE SET

FUNCTIONAL STANDARDS
(P+ DOCUMENT #360)

Document
Appendix
Revision Number
Page
Effective Date

WHC-SD-GN-CSWD-30072
0
0
3 of 14
May 31, 1994

4.0 User Interface

4.1 Reports

Reports will display on the user's screen by default. Hard copy output, when requested, will be routed to the user's local printer. Reports may optionally be directed to an ASCII file if desired. The following standards will be adhered to in designing reports:

HARD COPY - Local reports will be produced on a Laser printer. The default font will be LETTER GOTHIC, 12pt. The default orientation will be portrait for all reports less than 80 columns in width, and landscape for all others. No cover page will be used, but each page will have a header

FILE - The report, already formatted, will be sent to a DOS ASCII file as directed by the user. Report orientation will mimic the on-line display.

Headers for all reports, on-line, local print or file will contain:

report name,
date of production, and
page # of #.

Each column on a report will have a title. Titles shall be separated from the report by a single line. Reports will contain the wording "Sensitive Data" where appropriate. The use and placement of "Sensitive Data" will be determined by organization (i.e., WHC, DOE) during detail report definition.

Query results will also be made available as delimited output suitable for download to a local DBMS such as PARADOX.

See appendix A for sample report layout.

4.2 Human - Machine Dialogue

The human - machine dialogue will be conducted via keyboard, mouse, screen, and audible tone. Although use of a mouse will be highly recommended, system navigation will be achievable without use of the mouse. The user shall have the option to disable the audible tone.

QUEST: SA/RELEASE1 SUPPORTING DOCUMENT DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-GN-CSWD-30072 D 0
FUNCTIONAL STANDARDS (P+ DOCUMENT #360)	Page Effective Date	4 of 14 May 31, 1994

Information messages to the user shall be displayed in "Dialogue Boxes". All dialogues must inform the user and prompt the user for an action.

Collection of information from the user will avoid use of a dialogue box unless the intent is to collect specific instruction from the user such as specification of path, file open parameters, and confirmation of instructions (e.g. file delete).

Audible tones shall be used (disabled at user's discretion) to advise the user of inappropriate key depression, and error conditions.

Error messages will also advise the user of action required to correct the error.

4.3 Graphical User Interface (GUI)

All screens designed for intelligent workstation display will employ a graphical user interface. Microsoft WINDOWS shall be the standard. See Appendix B for a sample of the Windows user interface standard.

These standards and guidelines are set forth in the SYSTEM APPLICATION ARCHITECTURE, COMMON USER ACCESS ADVANCED INTERFACE DESIGN GUIDE (CUA), IBM, 1989. Additional guidelines are found in COMMON USER ACCESS TIPS AND TECHNIQUES SECOND EDITION, IBM, 1990. All software written for the QUEST2 system will adhere to the CUA standard.

Commercial, off-the-shelf products will eventually be used to provide functionality in the QUEST2 system. Where these products are written specifically for the Windows environment, they will adhere to the CUA guidelines. Guidelines are presented in the 360 standards to handle products that do not adhere to the CUA guidelines. All modifications and extensions to any of the packaged software must adhere to the standards found in the vendor's package in addition to the CUA standards.

4.3.1 Common User Access (CUA)

Standards for the use of the graphical user interface will conform to the IBM COMMON USER ACCESS. The Common User Access document describes the complete set of controls (events that a user can execute via the interface) that will be used in the QUEST2 graphical user interface. The CUA document is an extensive collection of standards. Therefore, a smaller set of these available controls and constructs is described in this document, with a brief description of the control and its expected implementation.

4.3.2 Metaphor Usage

The underlying structure of the windows in a graphical user interface is called a *metaphor*. For the QUEST2 system, a **MULTIPLE DOCUMENT INTERFACE (MDI)** shall be employed. This metaphor is also known as a forms metaphor, where the screen displays a set of completed and blank forms for the user to review or complete.

4.3.3 General "Look and Feel"

- MDI windows** The multiple document interface (MDI) is a user interface presenting multiple documents (or windows) within a single application. An MDI application has one main window. Within that window, the user can open and work with several documents. The size of the MDI frame can be changed at any time by the user.
- Child Windows** Each document appears in its own separate child window within the main application window. Because each child window has a frame, system menu, maximize and minimize buttons, and icon, the user can manipulate it as if it were a normal, independent window. However, the child window cannot move outside the main application window.
- accelerator keys** The user will have accelerator keys (ALT or CTRL in combination with another key) to accomplish selections that can also be made with mouse or cursor movement.
- dialogue boxes** Dialogue box use shall be minimized for entry of data. When dialogue boxes are used to display error messages, the dialogue is to contain the error message and a statement prompting the user for corrective action if required.
- icons** The QUEST2 system shall employ icons to depict windows and documents. Drag and drop (placing one icon on top of a second that causes a function to be executed, such as dragging a document into a trash can or paper shredder) is NOT a part of the QUEST2 interface at this time.
- function bar** The user shall have the option of displaying a function bar that will contain icon buttons to access the subsystems of QUEST2. Other actions will be chosen from the action bar (drop down menu).

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENT DELIVERABLE SET

FUNCTIONAL STANDARDS
(P+ DOCUMENT #360)

Document WHC-SD-GN-CSWD-30072
Appendix 0
Revision Number 0
Page 6 of 14
Effective Date May 31, 1994

4.3.4 Menu Bar

QUEST2 development will begin with the CUA proscribed ACTION BAR: File, Edit, View, Option, Window, and Help. Other action choices will be added as appropriate.

4.3.5 Navigation

When a selected action results in the display of a different object, or a new dialogue (such as to route for approval), the window from which the navigation began shall remain visible (the user may then choose to maximize the new window). The parameters will be automatically passed and displayed in the new window (i.e., the user could take the keys from the first window and populate the new window based on those keys).

4.3.6 Help

Help text files, when made available by the System Manager, will be accessible via the standard Microsoft Windows "Help" subsystem.

The "Basic System Concepts" (document 230/240 in WHC-GN-SD-CSWD-20010), describe the standard for the QUEST2 help system.

5.0 Client/Server Distribution

The QUEST2 system is being developed as a Client/Server system. The server will be centrally located or placed strategically on the network to optimize performance for the client base. Clients can be located virtually anywhere on the Hanford site and gain access to the QUEST2 system, provided their workstation meets the physical requirements and they are connected to the HLAN.

5.1 Data Distribution

QUEST2 data tables will reside on the server. Data required to make edits (very stable in nature) may be downloaded to the workstation for use during a work session. Integrity checks must be conducted prior to updating the server data base.

5.2 Processing Distribution - Application Code

Application code will reside on both the client and the server. All user interface code will be on the client, while all database updates will be conducted on the server.

QUEST: SA/RELEASE1 SUPPORTING DOCUMENT DELIVERABLE SET	Document Appendix	WHC-SD-GN-CSWD-30072 D
	Revision Number	0
FUNCTIONAL STANDARDS (P+ DOCUMENT #360)	Page	7 of 14
	Effective Date	May 31, 1994

Additionally, application code may be shifted from client to server, or vice versa, for system performance reasons.

The QUEST2 system will require that the code residing on the workstation be validated prior to allowing a connection to the server.

6.0 DATA INTEGRITY

Data integrity must be maintained. Data integrity assures the accuracy, correctness and validity of the database.

- o Accuracy refers to the exactness of the data. Example: an Action Summary report shows four open actions for condition ABC123, when there are really four open actions for condition ABC123 - no more, no less.
- o Correctness implies that the data truly represents the facts about the real world entities. Example: An employee's payroll number actually is 67989.
- o Validity pertains to the logical consistency of the data. Example: If an employee works for the accounting department, he/she must be associated with a valid accounting department number.

There are two kinds of integrity:

- o The first type, data integrity, is intrinsic to the database: it is an integral component of the data. In relational databases, all relationships are represented by primary and foreign key values. There are no explicit system-maintained pointers in the rows. While this provides the advantage of a simple data model and a high degree of data independence, it also means that when key values change, they must be changed every place they occur in the database. Example: In a Personnel database, if the department number (dept# column in the department table) for accounting changes from 020 to 040, every place it is referenced in the database (all foreign keys that reference dept#) must be changed to 040, for integrity to be maintained. Otherwise, key values would exist that reference nonexistent or incorrect departments.
- o The second kind of integrity is extrinsic to the database. It enforces the rules of the application of business domain being modeled. Every business has rules, policies, and procedures that must be enforced. These rules typically vary from business to business, but they must be enforced to ensure the validity of the data.

**QUEST: SA/RELEASE1 SUPPORTING
DOCUMENT DELIVERABLE SET**

**FUNCTIONAL STANDARDS
(P+ DOCUMENT #360)**

**Document
Appendix
Revision Number
Page
Effective Date**

**WHC-SD-GM-CSWD-30072
D
0
8 of 14
May 31, 1994**

Example: There may be a rule that no employee should be paid more than \$175,000, or that spouses cannot work in the same department.

6.1 Data Integrity Enforcement

Integrity can be effected in the application code, or it can be specified at the forms or even the database definition level. Every system, both application and Data Base Management System (DBMS), should guarantee the integrity of the data that it manipulates. Ensuring data integrity implies enforcing the semantics of the real world or the particular environment within the actual data. This usually means that any update to the data is constrained to fit certain rules. These rules are referred to as integrity constraints.

6.1.1 System Security

System security supports integrity by controlling the kind of access (read, write, edit) that is granted to a particular user. System security is addressed in the "Basic System Concepts", deliverable 240 under section 3.0 "Control Concepts".

6.1.2 Application Enforced Integrity

Enforcement of integrity for the QUEST2 system through application code will not be encouraged except in special instances. This will reduce the disadvantages associated with application enforced integrity, which are:

- o Application must devote significant amounts of code to data integrity logic, especially for complex constraints.
- o The integrity logic must be enforced consistently by all front-end applications - one nonconforming application can corrupt the integrity of the database.
- o Updates to the database from ad-hoc interfaces (using the Data Manipulation Language - DML) must be carefully controlled and monitored to ensure that none of the integrity constraints are violated.
- o If the integrity constraints change they could potentially affect multiple applications or modules which are utilizing the same data.

QUEST: SA/RELEASE1 SUPPORTING	Document	MHC-SD-GN-CSWD-30072
DOCUMENT DELIVERABLE SET	Appendix	D
	Revision Number	0
FUNCTIONAL STANDARDS	Page	9 of 14
(P+ DOCUMENT #360)	Effective Date	May 31, 1994

6.2 Data exchange between processes

When navigation from one object to a different window occurs, either to allow a different function to take place, or to view a related object, the selection of the data to be displayed in the new window will be as transparent to the user as is possible. For example: if a person's name and organization number are entered in the Initiating Document window, the same person/organization number information will display in the associated Condition window. This exchange between processes will minimize re-selecting or re-keying information that the system already has.

7.0 DATA BASE STANDARDS

The intent of this section is to establish the rules for interaction with the database. These guidelines provide the parameters to be utilized when generating the Functional Specifications (Deliverable 490).

7.1 Concurrent Update

The QUEST2 database is not considered highly dynamic with regard to Condition and Action data. The Sybase DBMS locking scheme is expected to be sufficient during concurrent database usage.

7.2 Audit Trails

Sybase transaction logging and periodic database backups will support audit trails. A security-oriented audit trail is not necessary.

7.3 Data Type Formats

Format standards are:

mm/dd/yy	-	dates
hh:mm:ss	-	time stamps
#,##0.00	-	currency
PeopleCORE	-	Name
Standard		

QUEST: SA/RELEASE1 SUPPORTING DOCUMENT DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-GN-CSND-30072 0 0
FUNCTIONAL STANDARDS (P+ DOCUMENT #360)	Page Effective Date	10 of 14 May 31, 1994

7.4 Archive Policy

The archive strategy is discussed in the "Basic System Concepts" (deliverable 230/240). It is described further in the "System Performance Criteria" (deliverable 350).

7.5 System Generated Identifiers

The QUEST2 system will generate unique identifiers for the following entities:

- Initiating Documents
- Conditions
- Actions and
- Comments associated with the three above.

The standard for these identifiers will be 9 character numeric and each new value must be greater than the last. The identifier will be constructed as: yymmmnnnn where y=year, m=month and n=numeric.

8.0 PROCESSING STANDARDS

8.1 Errors and Exceptions

The QUEST2 applications will always attempt to safeguard the user from error situations, but there will be times when a message must be displayed. Message boxes will be movable but not sizable, contain a title bar and a system icon. Every message box consists of a type icon, explanatory text, and one or more push buttons. See CUA, page 103 - 106 for additional information.

8.2 Information Messages

An information message tells users that a computer function is performing normally or has performed normally. The "information" icon must be displayed along with the text and an "OK" push button. A "help" push button is optional.

8.3 Warning Messages

A warning message tells users that a potentially undesirable situation could occur. Users only need to respond to the message to continue. However, corrective action may be required later to avoid an error situation: e.g. warning the user that a file is about to be deleted.

The warning message box always contains the "warning" icon, text to describe the situation and two of the following command buttons: OK and Cancel or Yes and No.

8.4 Action Messages

An action message tells users that an exception condition has occurred. Users must perform an action to correct the situation. Action messages are used in situations that range from minor application-related conditions that stop users from continuing with the current dialogue to serious system-related conditions that stop users from continuing to work with any application in the system.

The text of the message must convey the severity of the condition and suggest an action that will correct the situation (some conditions are not correctable and the message should communicate this when appropriate).

The action message box contains the "stop sign" icon, text and may contain Retry, Cancel, and/or Help push buttons.

8.5 Audible Feedback

Audible feedback is a beep from the computer that either warns users that they have performed an erroneous action or draws the users' attention to prompt action. The User has the option to turn off/on the beep in the Windows control panel.

The following situations are appropriate for audible feedback:

- o When the application displays warning and action messages.
- o When the user types a character that is not a valid mnemonic (accelerator key) in selection fields.
- o When users attempt to select an unavailable (grayed) choice from the action bar.

8.6 Process Recovery

Backup and recovery procedures will be developed to protect the availability and recoverability of QUEST2 data. These procedures will allow recovery of all completed transactions and restoration of the system within four hours of the time the system returns on-line.

QUEST: SA/RELEASE1 SUPPORTING DOCUMENT DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-GN-CSWD-30072 D 0
FUNCTIONAL STANDARDS (P+ DOCUMENT #360)	Page Effective Date	12 of 14 May 31, 1994

During an emergency situations such as a system outage, users will revert to a manual process. Data from the manual system will be entered into QUEST2 when the system is again available.

A disaster recovery plan will be developed and will include a weekly backup being stored in a separate building from the server.

8.7 Geographical Distribution of Processing

The initial releases of QUEST2 will have no distributed processing restrictions i.e., any user, anywhere on HLAN, could conceivably access any of the QUEST2 system functions. "User Type" rights will be identified which will grant/restrict access to the appropriate system functions.

9.0 CONTROL STANDARDS

9.1 Access

Users must be granted only those access privileges necessary to perform their assignments. To satisfy this requirement, the following access standards will be followed:

- o An access authorization policy will be defined prior to implementing the QUEST2 system. This policy will include the following features:
- o The physical files that will be protected from unauthorized modification, disclosure, or both.
- o The System Manager will be responsible for reviewing, authorizing and monitoring any access to the QUEST2 system.
- o The System Manager will be responsible for reviewing, authorizing, and monitoring the use of QUEST2 data and QUEST2 software on non-WHC computers (if any).
- o Authorization for individual access must be based on a documented request. This request must be signed by the user accepting responsibility for the access granted by the user's manager. A user identification form will be designed to request access to the QUEST2 system. The form will include the following information:

- User name
 - Payroll number
 - Date requested
 - Access requested (by role)
 - Justification for user access
 - Approved/requested by
 - Date granted/installed
 - Closure date
 - Date trained
- o User accounts shown inactive over 90 days must be reported to the System Administrator. These accounts will be reviewed with the account owner to assess continued need. The manager of an employee that has left his or her job (termination or transfer) will notify the System Administrator. The System Administrator will retrieve the applicable user identification form and remove the account from the system. The account closure date will be noted on the form and the form retained for one year.

9.2 Passwords

Users will be required to change passwords that are 90 days old. The system will prompt the user for a new password when it is time for the old password to expire. The required length for passwords is a minimum of six characters and a maximum of 30.

Passwords can be composed of alphanumeric or special characters.

10.0 COMMON PROCESSES

In QUEST2, "processes" are manifested through "objects". The QUEST2 system will employ the Windows standard Graphical User Interface (GUI), with PowerBuilder as the development tool to build this interface. Inherent to the Windows metaphor is the concept of object-oriented programming, a disciplined programming style that incorporates key characteristics which differentiate it from traditional structured programming models. In short, an object is anything, real or abstract, about which we store data and those operations that manipulate the data. This approach naturally promotes code modularity and code reusability.

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENT DELIVERABLE SET

FUNCTIONAL STANDARDS
(P+ DOCUMENT #360)

Document
Appendix
Revision Number
Page
Effective Date

WMC-SO-GN-CSWD-30072
D
0
14 of 14
May 31, 1994

Currently, the following common objects (processes) have been identified:

<u>Common objects</u>	<u>Unit processes affected</u>
Print	UPQST14 Print Init. Doc. UPQST24 Generate/View Response Letter UPQST36 Generate Risk Evaluation Report UPQST37 Generate RCA Report UPQST44 Create Action Reports UPQST72 Generate Canned Reports
Build Distribution Lists	UPQST11 Update Document UPQST25 Maintain Conditions UPQST43 Maintain Actions
Database Queries for Condition, Action data	UPQST17 Query Conditions UPQST26 Query/Report Cond. Status UPQST42 Query Conditions and Actions UPQST75 Review Pending Items UPQST76 Perform System Navigation UPQST77 Perform Ad-hoc Queries

The details of the common processes will be described in each of the Functional Specifications (490) for the unit processes affected. Also, additional common objects will be identified during functional specification work, i.e., Window or Menu attributes that will be common among two or more windows.

QUEST: SA/RELEASE1 SUPPORTING	Document	WHC-SD-GN-CSWD-30072
DOCUMENTS DELIVERABLE SET	Appendix	E
	Revision Number	0
OPERATING ENVIRONMENT	Page	1 of 5
(P+ DOCUMENT #370)	Effective Date	May 31, 1994

1.0 PURPOSE

This document describes the operating and development environments that will be used for QUEST2. This expands on the general environment description reflected in the Technical Alternatives document (260) (see WHC-GN-SD-CSWD-20010).

2.0 SCOPE

This document only describes the operating environment "framework". The details will be further defined during the Functional Design phase.

3.0 DEFINITIONS

ANSI	American National Standards Institute
DOE	Department of Energy
CADS	Corrective Action Data Systems (part of WHC/ESQ)
HLAN	Hanford Local Area Network
ISQL	Interactive Structured Query Language
IEEE	Institute of Electrical and Electronics Engineers
IRM	Information Resources Management (part of BCSR)
	Data Processing contractor for Westinghouse Hanford Company.
ISS	Information and Scientific Systems (part of ISS)
PB	Power Builder (the client software)
PBL	Power Builder Library
QA	Quality Assurance
	The process of assuring to management a degree of confidence that the end product conforms to established administrative and technical requirements as well as all the applicable standards in effect during the software development effort.
QUEST	Quality, Environmental, and Safety Tracking System
Server	A computer used to run the software that manages the data and the system.
SGML	Standard General Markup Language
SQL	Structured Query Language
TCF/IP	The Hanford Local Area Network standard communication protocol
VB	Visual Basic

4.0 ENVIRONMENT DESCRIPTION

4.1 Hardware

This section describes the hardware that will be used by QUEST2.

Network Server

The Network Server will be a standard UNIX Server connected to HLAN. This server may be one of the following types:

QUEST: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET

Document
Appendix
Revision Number
Page
Effective Date

WHC-SD-GN-CSWD-30072
E
0
2 of 5
May 31, 1994

OPERATING ENVIRONMENT
(P+ DOCUMENT #370)

- SUN SPARCserver 1000
- DECsystem 3000
- IBM 9000 G40
- Data General 9500

Since the Sybase Server software can run on any of the above system, the actual hardware being used is irrelevant. Listed below are the important hardware features of the platform being used:

- Minimum of 7 GBytes of Disk storage.
- Minimum of 100 TCPb rating.
- Tape Drive for Backups

End User Work Station

Each workstation must meet the following requirements:

- Minimum 386 class machine running at 20 MHz.
- Local Disk storage available at least 10 MBytes. (Developers need at least 30 MBytes.)
- Minimum of 8 MBytes of RAM.
- VGA display.
- Mouse.
- HLAN Connection.

Printers

Each workstation will have access to an HP Laserjet II or better printer through an LPT or COM port. The LPT ports may be redirected over the network. All outputs may also be redirected to a file. Access will be provided to the printers connected to the ISIS.

4.2 Software

This section describes the software required to develop and use QUEST2.

Network Server

The network server will run the following software:

- SyBase System release 10.0 server software.
- UNIX Operating System.
- TCP/IP network protocol.

End User Work Station

Each user workstations will contain the following software:

QUEST: SA/RELEASE1 SUPPORTING	Document	WHC-SD-GN-CSWD-30072
DOCUMENTS DELIVERABLE SET	Appendix	E
	Revision Number	0
OPERATING ENVIRONMENT	Page	3 of 5
(P+ DOCUMENT #370)	Effective Date	May 31, 1994

- MS/DOS 6.0 or later operating system
- Windows 3.1 or later.
- LanManager 2.2 or later using TCP/IP.
- A copy of the QUEST program.
- WordPerfect for Windows Version 5.2.

Developer's Work Station

Each developer's workstation will contain the following software:

- MS/DOS 6.0 or later operating system
- Windows 3.1 or later.
- LanManager 2.2 or later using TCP/IP.
- Access to PowerBuilder 3.0 or later.
- Access to SyBase Server toolkit Version 5.0.
- WordPerfect for Windows Version 5.2.

Other Software

Listed below is other software that is used by QUEST2.

- FTP - Unix file transfer process to move files from other UNIX or LSIS computers.
- Visual Basic. (QUEST Startup Program)
- SyBase Utilities.

4.3 Connectivity

The work stations will access the Network server using TCP/IP. The connection will be managed by the QUEST or PowerBuilder programs.

5.0 PRINCIPLES OF UTILIZATION

5.1 Development Environment

All development will be done using local copies of the PowerBuilder and some Sybase programs. The figure 'Developer's Setup' shows the distribution of processes and data.

The 'Local Data' contains:

- PowerBuilder Developer's software.
- SyBase ISQL and other utilities.
- Local copies of the PB PBLs.

The 'QUEST Data' contains:

Developer's Setup

- QUEST SQL Data
- Stored Procedures

The 'Developer Server' contains:

- Master copy of the PB PBLs.

5.2 User Environment

The user accesses the QUEST2 system by clicking the QUEST windows icon. This will start a VB program to update the local programs before starting the PowerBuilder built program. This program will open a connection to the QUEST database.

During operations of the QUEST program some lookup tables will be created on the users' local drive. These tables will be used to validate the inputs before sending requests to the QUEST database.

The figure 'User Setup' shows the distribution of processes and data. All files for the QUEST application are stored on the Local Disk drive and are updated from the Network Server when the application is started.

User Setup

The 'QUEST Data' will contain the following items:

- QUEST SyBase data
- Master copy of the programs developed for QUEST.

The 'Local Data' will contain the following items:

- A copy of all the QUEST Powerbuilder programs.
- Static Lookup tables.

QUEST2: SA/RELEASE 1 SUPPORTING DOCUMENTS DELIVERABLE SET	Document Appendix Revision Number	WHC-SD-GN-CSWD-30072 F 0
DESIGN STANDARDS (P+ DOCUMENT #380)	Page Effective Date	1 of 2 May 31, 1994

1.0 INTRODUCTION

1.1 PURPOSE

This deliverable describes the Design Standards that will be used for QUEST2. These standards along with Functional Standards in the 360 deliverable will be guidelines for the QUEST2 developers. This will be considered a 'Living Document' and will be updated to reflect the current development environment.

1.2 SCOPE

This deliverable describes or references the design standards for PowerBuilder and Sybase code. Documentation and other standards are described in the Quest2 032, 320, and 360 deliverables (see WHC-SD-GN-CSWD-20010 and appendices A and D of this Deliverable Set).

2.0 PROCESS RULES

2.1 Structure of the Processes

This subject is addressed in the 360 deliverable.

2.2 Software features

The Quest project will follow the PowerBuilder (PB) naming conventions described in 'Introduction to Powerbuilder' Appendix A. These and others standards are described in attachment 1 which was copied from the Procurement and Materials Management (PAMM) project. The following are additional guidelines for coding.

- All PowerBuilder scripts will contain SQL code that is FIPS 127-1 (ANSI SQL 89) compliant. When SyBase extensions are required, they will be implemented using Stored Procedures or isolated into Global PB functions.
- The PB objects will be stored on the fileserver (\\WHC399\QUEST). Each Developer will have their own '.PBL' file in which they will store objects that they are presently working on. The PB Library Check Out/In routines will be used to transfer copies of the objects to the developer's PBL. Before an object is checked in, the original version will be copied to the Backup PBL with copy date added to the object name.

3.0 COMMUNICATION RULES

The following guidelines are for interprocess and application communications.

- Where ever possible Object Linking and Embedding (OLE) or Dynamic Data Exchange (DDE) will be used to communicate to other Window applications, this includes WordPerfect and CC:Mail. If the application can not use this form of communication then ASCII files will be used.
- Interprocess communications will be handled using Window and PB events and PB global variables. A listing of these global variables will be described in attachment 1.

3.1 Backups

Each developer will backup their working files to the network or to a local tape drive. The files on the fileserver are backed up nightly.

4.0 Database Rules

Boeing Productivity Plus Workbench G-6021-92WW-012 will be used for the modeling standards. Data Administration Standards WHC-CM-2-6 describes Hanford specific standards. The following additional guidelines are for the database structure.

- Referential Integrity will be implemented using Stored Procedures.
- Separate Databases will be created for Production, Test, and each developer.

4.1 Backups

The database will be backed up on a regular basis.

5.0 COMMON PROCESSES

A listing of common processes and parent windows for QUEST2 will be described in attachment 1.

Contents

1.0 Coding Standards	F-1-5
1.1 Introduction to Coding Standards	F-1-5
1.2 General Coding Standards	F-1-5
1.2.1 Commands which will NOT be used	F-1-5
1.2.2 Audit Information	F-1-6
1.2.3 Variable Declarations	F-1-6
1.2.4 Using Comments in PowerScript	F-1-7
1.2.5 Using Capitalization in PowerScript	F-1-7
1.2.6 Using Indentation in PowerScript	F-1-7
1.2.7 Using Statement Separators in PowerScript	F-1-7
1.2.8 Object Functions and User Events	F-1-7
1.2.9 Coding Menu Events	F-1-8
1.2.10 SQL Error Checking	F-1-8
1.2.11 Referencing Other Objects' Instance Variables or Attributes	F-1-8
1.2.12 Overcoding	F-1-8
1.2.13 Validation Issues - Front End vs. Back End vs. Both	F-1-8
1.2.14 Using Literals in PowerScript	F-1-9
1.2.15 Using Global Variables in PowerBuilder	F-1-10
1.2.16 Using Data types in PowerBuilder	F-1-10
1.2.17 Global Structures in PowerBuilder	F-1-10
1.2.18 Embedded SQL vs. Stored Procedures	F-1-10
1.2.19 SQL Server Script Audit Information	F-1-10
1.2.20 Using Comments in SQL Server Scripts	F-1-12
1.2.21 Using Capitalization in SQL Server Scripts	F-1-12
1.2.22 Using Indentation in SQL Server Scripts	F-1-12
1.2.23 Formatting SQL Statements	F-1-13
1.2.24 Variable Declarations	F-1-13
1.2.25 SQL Script Files	F-1-13
1.3 Transaction Management	F-1-14
1.3.1 PowerBuilder PowerScript	F-1-14
1.3.2 SQL Server Script	F-1-14
1.4 Concurrency Management	F-1-15
1.5 Handling Null Values	F-1-15
2.0 Naming Conventions	F-1-16
2.1 Object/Variable Naming & Use Standards	F-1-16

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 2 of F-1-65
Effective Date May 31,1994**

2.1.1 PowerBuilder Object Naming Conventions	F-1-16
2.1.2 Variable Naming Conventions in PowerBuilder	F-1-20
2.1.3 Object Naming SQL Server	F-1-21
2.1.4 Variable Naming Conventions in SQL Server	F-1-25
2.1.5 PowerBuilder Library Categories	F-1-25
3.0 Look and Feel (static)	F-1-26
3.1 General Development Standards	F-1-26
3.2 Fonts	F-1-27
3.3 QUEST Window Types	F-1-27
3.3.1 Document Sheets	F-1-27
3.3.2 Select Window	F-1-27
3.3.3 Response Window	F-1-28
3.4 Reports	F-1-28
3.4.1 Standard Report Layout	F-1-28
3.4.2 Standard Report Header	F-1-28
3.4.3 Standard Report Body	F-1-29
3.4.4 Standard Report Footer	F-1-29
3.4.5 Standard Report Trailer	F-1-30
3.4.6 Standard Banner Page	F-1-30
3.4.7 Sample Report	F-1-30
4.0 DataWindows	F-1-31
4.1 DataWindow Standards	F-1-31
4.1.1 Creating Update/Display DataWindows	F-1-31
4.1.2 Common Standards of DataWindows	F-1-31
4.1.3 FreeForm DataWindows	F-1-32
4.1.4 Tabular DataWindows	F-1-33
4.1.5 Grid DataWindows	F-1-33
4.1.6 Using Extended Column Definitions in PowerBuilder	F-1-33
5.0 Error Handling	F-1-35
5.1 Overview	F-1-35
5.2 SQL Server Error Handling on the Back End	F-1-35
5.3 SQL Server Error Handling on the Front End	F-1-36
5.4 Front End Error Handling	F-1-36
5.5 Message Box	F-1-37
5.6 MicroHelp	F-1-38

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 3 of F-1-65
Effective Date May 31,1994**

6.0 Navigation (dynamic)	F-1-39
6.1 Overview	F-1-39
6.2 General Navigation Concept	F-1-39
6.3 Interaction of Navigation and Security	F-1-39
7.0 Inter System Interface Standards	F-1-41
7.1 Overview	F-1-41
8.0 Application Optimization	F-1-42
8.1 Overview	F-1-42
9.0 Help Standards	F-1-43
9.1 Steps in Developing Help	F-1-43
9.2 Tools In Developing Help	F-1-43
9.3 Help Document Standards	F-1-44
9.4 Filename Conventions in Help	F-1-45
9.5 Context-Sensitive Help	F-1-45
9.6 Accessing Help	F-1-46
9.6.1 PowerScript Code	F-1-46
9.7 Reference Materials	F-1-47
10.0 Development Tools	F-1-48
10.1 Overview	F-1-48
11.0 Development Techniques	F-1-49
11.1 Overview	F-1-49
11.2 Creating, Modifying, and Dropping Stored Procedures	F-1-49
12.0 QUEST Stub Overview	F-1-51
12.1 Overview	F-1-51
12.2 PowerBuilder Global Structures	F-1-51
12.2.1 Structure: s_gvs - Global Variable Structure	F-1-51
12.2.2 Structure: s_sec - Security Structure	F-1-53
12.2.3 Structure: s_usr - User Structure	F-1-53
12.3 PowerBuilder Standard Functions	F-1-53
12.3.1 Standard Function: f_retrieve	F-1-54
12.3.2 Standard Function: f_load_dw_ddlb	F-1-54
12.3.3 Support Function: f_sp_error	F-1-55
12.3.4 Support Function: f_get_dw_col_dtype	F-1-55

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 4 of F-1-65
Effective Date May 31,1994**

12.3.5 Support Function: f_opensheet	F-1-56
12.3.6 Support Function: f_parse_error_msg	F-1-56
12.3.7 Support Function: f_displayhelp	F-1-57
12.3.8 Support Function: f_toggle_modified	F-1-57
12.3.9 Support Function: f_toggle_select	F-1-58
12.3.10 Window Function: f_set_enable	F-1-59
12.4 QUEST Stub Superclass Menus	F-1-59
12.4.1 m_QUEST_main	F-1-59
12.4.2 m_sys_sc	F-1-59
12.5 QUEST Stub Superclass Windows	F-1-60
12.5.1 w_sc_select	F-1-60
12.5.2 w_sc_doc_sheet	F-1-60
12.5.3 w_sc_doc_sheet_tab	F-1-62
13.0 Display Format Definitions	F-1-63
13.1 Introduction	F-1-63
13.1.1 Format: Amount	F-1-63
13.1.2 Format: Date	F-1-63
13.1.3 Format: Datetime	F-1-63
13.1.4 Format: Percent	F-1-63
13.1.5 Format: Time	F-1-64
14.0 User-Defined Data types	F-1-65
14.1 Introduction	F-1-65
14.1.1 Data type: dt_datetime	F-1-65
14.1.2 Data type: dt_ind	F-1-65

1.0 Coding Standards

1.1 Introduction to Coding Standards

This section will provide information regarding the coding of:
PowerBuilder (PB) Powerscripts
Structured Query Language (SQL) Server scripts

The issues covered will include:
General coding standards (PB, SQL, SQR)
Transaction Management (PB, SQL)
Concurrency Management (PB, SQL)
Null Values (PB, SQL)

1.2 General Coding Standards

1.2.1 Commands which will NOT be used

Halt()

Stops application gracelessly, can cause problems.

SetTrans()

Sets the values in the internal transaction object for a datawindow to the values for a specified transaction. Should use **SetTransObject()** instead.

1.2.2 Audit Information

The following audit information will be included in a heading preceding each script:

```
//*****  
//Creation Date:      mm/dd/yy  
//Created By:        name  
//Purpose of Script:  description of the purpose of the script  
//*****  
// Modification History  
// Date      By          Reason          Rev#  RFS #  
//-----  
//  
//*****
```

The standard heading template will be available. This file will be in PowerScript format and may be pasted in at the head of each script.

Functions will include the following additional information:

```
//Usage:  f_function( arg1, arg2, ...)  
//Where:  arg1 = arg1 description  
//        arg2 = arg2 description  
//Return: return value description
```

Each time a modification is made, a new comment will be inserted in the Modification History section briefly describing the modification.

1.2.3 Variable Declarations

All local variable declarations will be grouped together at the start of the script for which they are declared. These will be the first executable lines of code contained in the script. The group of local variable declarations will be followed by a single blank line.

1.2.4 Using Comments in PowerScript

All comments will be indicated using two slashes '//'.

All comments will be on a line by themselves (no mixing of PowerScript code and comments on the same line). The single exception to this rule is variable declarations. Each variable declaration may be followed by a comment describing the variable, on the same line.

All comments will be preceded by a single blank line.

Additional blank lines may be added to enhance readability of the script.

1.2.5 Using Capitalization in PowerScript

All SQL reserved words will be in all caps. All other PowerScript code will be lowercase. The booleans TRUE and FALSE will always be capitalized.

1.2.6 Using Indentation in PowerScript

Each step of indentation will use a <TAB> character. PowerBuilder constructs (i.e. if/then, for..next, etc.) will be indented as in the examples provided through the 'Paste Statement' functionality.

1.2.7 Using Statement Separators in PowerScript

Statement separators will not be used. Each PowerScript statement will be required to begin on its own line. All PowerScript statements which will be required to be split onto two or more lines using the continuation character if the statement is too long to fit on a single line of a maximized script entry screen.

1.2.8 Object Functions and User Events

PowerBuilder object functions and user events will be used to define a group of operations which are common to several events in an object, or which are to be inherited by descendant objects. The use of object functions and user events will reduce the number of global variables required, and aids in the inheritance of functionality between objects.

User events must be used instead of object functions when the user event needs to be modified in a descendant object. Object functions will be inherited, but cannot be modified in descendant objects.

Functions which are applicable only to a single object will be scoped to the object in which they are used (window, menu, or user-object). Functions which are used in multiple objects will be scoped to the entire application (global functions).

1.2.9 Coding Menu Events

Menu events should rarely be coded. Coded menu events, when possible, should contain only generic references. Menus should simply trigger other events in the window to which the menu is attached. User-events can be created in the window for the actual functionality that must be coded. This will allow a single menu to be used on multiple windows.

1.2.10 SQL Error Checking

Error codes will be checked after executing any embedded SQL (including stored procedures) to determine its success or failure. Use the `f_sp_error` function.

1.2.11 Referencing Other Objects' Instance Variables or Attributes

One object should never reference another object's instance variables or attributes directly. All modifications to an object's instance variables and attributes, aside from the scripts dealing with the object itself, should be performed through function calls. This will isolate the attribute changes to one place, and ease code maintenance. The use of generic references will also allow one function to handle attribute changes for many objects.

1.2.12 Overcoding

Beware of overcoding. As much functionality as possible should be consolidated into the superclass objects. Coding events in descendant objects can easily override superclass functionality unnecessarily.

1.2.13 Validation Issues - Front End vs. Back End vs. Both

Validation will occur on the following 2 levels:

DataWindow Validation:

Complete column format validation will occur when a column loses focus in a datawindow. This validation will be placed in a function called out of the 'ItemFocusChanged' event for the datawindow. This validation will be identical to the final format validation tests which will be performed on the server.

A separate function will be created to perform all standard domain validations for each data element (i.e. all validations documented in the 320). This function will be called to perform validation each time the element is 'tabbed' through on an updateable datawindow. Additional function-specific syntax validation will need to be addressed separately within the 'ItemFocusChanged' event of the datawindow.

When required (to be determined), code validations may be accomplished by loading the SQL Server code tables into memory on the client. Code tables will not be embedded into PowerBuilder for validation purposes. More complex code validations (i.e. cost code, product code, etc.) will be performed on the server on an entire datawindow at a time.

Special context validation cases may be identified where complex codes and/or relationships between columns may need to be validated on the client.

Server Validation:

All column format and database integrity rules will be enforced on the server. Much of this validation may be redundant to the validation which was performed on the client. All context validation on the server will be performed in the maintenance stored procedures.

General Validation Rules:

All checks for 'blank' will additionally check for `LEN(var) > 0` and `var=SPACE(LEN(var))` (ie. not just spaces).

1.2.14 Using Literals in PowerScript

Literals will NOT be used in PowerBuilder scripts. Constants will be defined and used in place of literals.

1.2.15 Using Global Variables in PowerBuilder

Global variables should NOT be used for performing miscellaneous processing within the scripts for an object. Passing information between objects will use parameters with the message.

1.2.16 Using Data types in PowerBuilder

The 'decimal' data type will NOT be used. Instead, variables should be declared of the type 'real' or, if necessary, 'double'. Data types in PowerBuilder do not match one on one with those in SQL Server. Special care must be taken to insure the correct data types are being used.

1.2.17 Global Structures in PowerBuilder

If several global variables are identified which reference the same entity, the number of global variables can be reduced by replacing these global variables with a single global structure. Global structures will be scoped to the entire application, and will be named as described in the Object Naming Conventions topic.

1.2.18 Embedded SQL vs. Stored Procedures

Embedded SQL

The use of embedded SQL will be minimized. It is a better practice to store all SQL code on the server as stored procedures, where it can be maintained in one place only.

Stored Procedures

Whenever possible, stored procedures will be used to perform processing against the database. Stored procedures will always be used for datawindow retrievals and updates.

1.2.19 SQL Server Script Audit Information

The following audit information will be included in a heading preceding each scripts:

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 11 of F-1-65
Effective Date May 31,1994**

Table Creation:

/* Sybase Table Creation

Table Name: *table name*
Procedure Type: [MNT/QRV]
File Name: *dos filename*
Created Objects: *object1*
object2
objectn
Purpose: *purpose description*
Author: *author name*

Creation Date: *mm/dd/yy*

Modification History

Date	By	Reason	Rev#	RFS #
------	----	--------	------	-------

***/**

Stored Procedures:

/* Sybase Stored Procedure

Procedure Name: *procedure name*
Procedure Type: [MNT/QRV]
File Name: *dos filename*
Target Tables: *target table list*
Purpose: *purpose description*
Author: *author name*
Creation Date: *mm/dd/yy*

Modification History

Date	By	Reason	Rev#	RFS #
------	----	--------	------	-------

***/**

Triggers:

/* Sybase Trigger

Trigger Name: *trigger name*
Trigger Type: *TG[i/u/d]*
File Name: *dos filename*
Target Tables: *target table list*
Purpose: *purpose description*
Author: *author name*
Creation Date: *mm/dd/yy*

Modification History

Date	By	Reason	Rev#	RFS #
-------------	-----------	---------------	-------------	--------------

***/**

The standard heading template will be available. This file may be pasted in at the head of each script. Each time a modification is made, a new comment will be inserted in the Modification History section briefly describing the modification.

1.2.20 Using Comments in SQL Server Scripts

All comments will be indicated using `/* ..... */` around the comments

All comments will be on a line by themselves (no mixing of SQL code and comments on the same line).

All comments will be preceded by a single blank line.

Blank lines will be used as required for readability.

1.2.21 Using Capitalization in SQL Server Scripts

All SQL reserved words will be in all caps. All other code will be lowercase.

1.2.22 Using Indentation in SQL Server Scripts

Each step of indentation will be a `<TAB>` character.

1.2.23 Formatting SQL Statements

Select Statements:

SQL select statements will be formatted such that each KEYWORD is on a new line, and tab stops are used to align the other portions of the statements, for example:

```
SELECT pu.concur_tm, pu.suid, pu.payrl_num
      FROM   pt_QUEST_user pu, sysusers su, master..syslogins sl, pt_QUEST_group pg
      WHERE  pu.payrl_num = @payrl_num
```

Other:

Parameter and variable declarations will be formatted such that only a single column is presented per line, and tab stops are used for alignment, for example:

```
CREATE PROCEDURE lp_QUEST_user_mnt
  (@mnt_type      char(1),
   @concur_tm     datetime,
   @suid          smallint,
   @payrl_num     char(7))
```

1.2.24 Variable Declarations

Declare all variables in one place. This simplifies tracking down variables. Declare all variables at the beginning of a procedure.

1.2.25 SQL Script Files

There are several types of SQL Scripts File that will be created. Each of the types is listed below along with the purpose and dos filename extension to be used:

1. Table Creation (.TBL): Create the table, rules and defaults
2. Trigger (.TG[i/u/d]): Create trigger
3. Maintenance (.MNT): Create Maintenance stored procedure
4. Query (.QRY): Create Query stored procedure
5. Conversion (.CNV): Create Conversion stored procedure
6. Functional View Creation (.FVW): Create a functional view (database view)

7. Foreign Key Definition (.FKY): Defines foreign keys (using SQL Server sp_foreignkey)
8. Index Creation (.IDX): Create any indexes required for a given table.
9. Index Drop (.IDD): Drop any indexes required for a given table.

1.3 Transaction Management

Transaction management will be controlled primarily within the SQL Server Scripts. However, two types of updates can occur; 1) Row by row update and 2) All or nothing update. Row by row updates will have the transaction management performed entirely within the SQL Server Scripts. All or nothing updates must, however, be controlled within the PowerBuilder PowerScript.

1.3.1 PowerBuilder PowerScript

The appropriate PowerScript statements will be built into the QUEST Stub Superclass objects to enforce transaction management. The SQLCA.autocommit property will be set to TRUE, such that Begin/End transactions are not implied. When a All or nothing update is required, the SQL autocommit property will be set to FALSE and Rollbacks and Commits must be issued.

1.3.2 SQL Server Script

SQL Server has a set of transaction control statements, that allow users to define transactions. By using transactions, users can insure database consistency and recoverability of their transaction. The following rules must be followed:

- 1) Transaction names are never used.
- 2) Stored procedures Always have Rollback Tran statements where applicable.
- 3) Stored procedures for row by row transactions will have always have Begin Tran and Commit Tran statements where applicable.
- 4) Stored procedures for all or nothing transactions will never have Begin Tran or Commit Tran statements.

- 5) Triggers will always contain Rollback Tran statements, but never Commit Tran statements.

1.4 Concurrency Management

The optimistic point of view regarding concurrency will be used, which states that it will be unusual for two users to update a given record at the same time. SQL Server timestamps will be used to insure that the record has not been updated. If an update is detected then the User will be notified.

1.5 Handling Null Values

The following standards have been established for handling NULL values:

- 1) All character columns will be defined as NOT NULL in SQL Server. Any necessary exceptions to this rule will be evaluated and approved by the DBA.
- 2) When performing updates, PowerBuilder will trap occurrences of NULL for character data types. Instead of passing the NULL as a parameter to the update stored procedure, PowerBuilder will pass an empty string.

2.0 Naming Conventions

2.1 Object/Variable Naming & Use Standards

2.1.1 PowerBuilder Object Naming Conventions

All Objects

All object names will be in all lowercase.

Superclass Objects

Superclass objects are any objects whose attributes are inherited by another object. All superclass objects will be identified by inserting an 'sc_' in the second position of the object identifier. Note: all superclass objects will be in the QUEST STUB.

Example: A user object which is a radio button with the text 'Married' which is contained on an ancestor window would be identified as 'uo_sc_rb_married'.

All descendant objects are automatically named within PowerBuilder according to the identifier of the ancestor object. The name of descendant objects cannot be changed.

Naming conventions for other objects are as follows:

Object	Convention	Description
Check Box	<i>cbx_[text]</i>	where [text] is the value of the text attribute of the check box.
Command Button	<i>cb_[text]</i>	where [text] is the value of the text attribute of the command button.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 17 of F-1-65
Effective Date May 31,1994

DataWindow	<i>d_{table name} _description</i>	Where [table/view name] is the name of the table or view which may be updated by the datawindow (or the primary table or view which is being queried, if updates are not allowed); [description] is descriptive text used to distinguish between multiple datawindows for a single table or view.
DataWindow Instance	<i>dw_{object name}_{nn}</i>	Where [object name] is the name of the datawindow object (without the 'd_' prefix) which will be referenced by this instance; [nn] is a sequential number required only if there are multiple instances of the same datawindow object contained in a window (should be a rare occurrence).
Dynamic DataWindow	<i>dw_dyn_{desc}_{n}</i>	<p>A dynamic datawindow instance will be considered to be one which is not a descendant or ancestor of any other instances, and one which is to be linked to more than one datawindow object.</p> <p>Where [desc] is a description of the purpose of the window; and where [n] is a sequential number used if necessary to distinguish between multiple dynamic datawindows contained on the same window which are used for the same purpose.</p>
Drop Down List Box	<i>ddlb_{element/variable name}</i>	<p>Where [element/variable name] represents the element or variable which is to be assigned by a selection from the drop down list box. If no element or variable is assigned from the list box, a descriptive name will be used.</p> <p>Drop down list boxes may be identified which contain selections which may apply to many table elements (i.e. state abbreviations). In these special cases, a more generic name may be assigned to the drop down list box (ddlb_state ?)</p>
Function	<i>f_{funname}</i>	Where [funname] will be a meaningful name describing the purpose of the function.
Group Box	<i>gb_{text}</i>	Where [text] is the value of the text attribute of the group box.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 18 of F-1-65
Effective Date May 31,1994**

Horizontal Scroll Bar	<i>hsb [element/variable name]</i>	For horizontal scroll bars which set an element value, the name will reference the applicable table and element. For horizontal scroll bars which set a variable value, the name will contain the variable name. If no element or variable is assigned from the scroll bar, a descriptive name will be used.
Line	<i>ln [n]</i>	PowerBuilder default, where [n] is the default line number.
List Box	<i>lb [variable name]</i>	Where [variable name] represents the variable which is to be assigned by a selection from the drop down list box. If no variable is assigned from the list box, a descriptive name will be used. List boxes may be identified which contain selections which may apply to many elements (i.e. state abbreviations). In these special cases, a more generic name may be assigned to the list box (ddlb_state ?).
Menu	<i>m [menuname]</i>	Where [menuname] is a descriptive name for the menu.
Multiple Line Edit	<i>mle [element/variable name]</i>	The multiple line edit name will reference the table and element name or variable name which is being displayed/updated in the multiple line edit. If no element or variable is associated with the multiple line edit, a descriptive name will be used.
Oval	<i>oval [n]</i>	PowerBuilder default, where [n] is the default oval number.
Picture	<i>p [filename]</i>	The picture name will reference the bitmap file which is to be displayed. The filename will be appended to the picture name, minus the '.bmp' extension.
Radio Button	<i>rb [text]</i>	Where [text] is the value of the text attribute of the radio button.
Rectangle	<i>r [n]</i>	PowerBuilder default, where [n] is the default rectangle number.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 19 of F-1-65
Effective Date May 31,1994

Round Rectangle	<i>rr_[n]</i>	PowerBuilder default, where [n] is the default round rectangle number.
Single Line Edit	<i>sle_[element/variable name]</i>	The single line edit name will reference the table and element name or variable name which is being displayed/updated in the single line edit. If no element or variable is associated with the single line edit, a descriptive name will be used.
Static Text	<i>st_[element/variable name]</i>	If the static text is being used as a label, the static text name will reference the appropriate table and element or variable name.
	<i>st_[n]</i>	PowerBuilder default, where [n] is the default static text number.
Structure	<i>s_[structname]</i>	Where [structname] will be a meaningful name describing the structure. [structname] should incorporate the associated QUEST entity name if appropriate
Standard User Object	<i>uo_[object type standard]</i>	Standard user objects will be named with the prefix 'uo_', followed by the standard object name as determined by the naming conventions for standard PowerBuilder objects. Example: A user object which is a radio button with the text 'Married' would be identified as 'uo_rb_married'. Note: All user objects will be in the QUEST STUB.
External User Object	<i>uo_ex_[description]</i>	[description] will be determined by the user object administrator. Note: All user objects will be in the QUEST STUB.
Vertical Scroll Bar	<i>vsb_[element/variable name]</i>	For vertical scroll bars which set an element value, the name will reference the applicable table and element. For vertical scroll bars which set a variable value, the name will contain the variable name. If no element or variable is assigned from the scroll bar, a descriptive name will be used.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 20 of F-1-65
Effective Date May 31,1994**

Window

*w_{t}[ssfmnn]
_{winname}*

Where:

[t] - Window type

m = main;

p = popup;

r = response;

c = child;

i = multi-document interface

The *[ssfmnn]* portion of the name should be used if applicable to specify the subject, facet, function and unit process with which the window is associated. This will be specified to as narrow a scope as possible.

ss = Subject (*mm*, *pr*, *ap*, *sm*)

f = Facet #

mm = Function #

nn = Unit Process

[winname] will be a descriptive name describing the purpose of the window.

2.1.2 Variable Naming Conventions in PowerBuilder

Variable names will following these conventions:

Variable	Convention	Description
Local	<i>{varname}</i>	Where <i>{varname}</i> is a descriptive name identified by the individual developer. Note: No prefix.
Shared	N/A	Shared variables will NOT be used.
Instance	<i>i_{varname}</i>	Where <i>{varname}</i> is a descriptive name identified by the individual developer.

Global

g_{varname}

Where [varname] is a descriptive name.

When additional global variables are identified during development, these must be approved. Local and instance variables may be created and named by individual developers as required.

All Variables

If the variable is of the same data type as a window control object, the *{varname}* portion of the variable name will consist of the standard window control prefix followed by an underscore, followed by a descriptive name. Refer to the object naming standards section for a list of the standard window control prefixes.

Example

A local variable of type 'checkbox' which indicates marital status could be named 'cbx_married'.

Constants will be managed and named in the same manner as standard variables. Constants will always be named and referenced in all caps, to distinguish them from standard variables. All other variables will be named in all lowercase.

The length of all variable names will not exceed 40 characters.

2.1.3 Object Naming SQL Server

The general naming convention for all SQL Server objects:
Prefix (XX).

Name (as necessary, under 12 when possible) Abbreviated from Table or Function being performed..

Suffix (XXX) as necessary.

Example:

lp_po_li_upd

Local Procedure, Purchase Order Line Item, Update.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 22 of F-1-65
Effective Date May 31,1994

ci_po_li_ponum_linum Clustered Index on Purchase Order Line Item.

pt_po_li Permanent Table, Line Item.

The naming convention for each SQL Server Object has been defined in the related topics below.

Object	Convention	Description
Clustered Index (CI)	<i>ci_{tblname}_{key}_{xxx}</i>	The lowest level of this type of index is data (leaf level) and the key column(s) specified determine the physical sort-order of data within the table. You can create only one Clustered Index per table.
Default (DF)	<i>df_{tblname}_{col_name}</i>	Places a default value in a column, when an end-user leaves it blank.
Non-Clustered Index (NI)	<i>ni_{tblname}_{key}_{xxx}</i>	The lowest level of this type of index is a pointer, and the key column(s) specified do not affect the physical order of the underlying data within a table. You can create up to 249 Non-Clustered Index per table, with each having a maximum of 16 columns (composite key) and a maximum key length (sum of all member columns) of 256 bytes.
Local Stored Procedure (LP)	<i>lp_{tblname}_{xxx}</i>	This prefix denotes all Non-SQL Server supplied stored procedures that are used specifically by a given document for query or maintenance. The name of the Local Procedure may either be derived from the table, or from the function being performed.
	<i>lp_{name}_arc</i>	Archive
	<i>lp_{name}_bld</i>	Build Temporary Table
	<i>lp_{name}_del</i>	Delete
	<i>lp_{name}_hst</i>	History

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 23 of F-1-65
Effective Date May 31,1994

	<i>lp_[name]_ins</i>	Insert records
	<i>lp_[name]_mnt</i>	Maintenance
	<i>lp_[name]_prg</i>	Purge
	<i>lp_[name]_qry</i>	Query
	<i>lp_[name]_rpt</i>	Reports
	<i>lp_[name]_sa</i>	Stats on system operations.
	<i>lp_[name]_sts</i>	Stats on user operations.
	<i>lp_[name]_upd</i>	Update
Common Stored Procedure (CP)	<i>cp_[name]_xxx</i>	This prefix denotes all Non-SQL Server supplied stored procedures that are used by all QUEST users for query or maintenance regardless of user access capabilities. See Local Stored Procedures for other naming guidelines.
Permanent Table (PT)	<i>pt_[tablename]</i>	This prefix denotes that the table is permanent.
Stored Procedure (SP)	<i>sp_[tablename]_[xxx]</i>	This prefix will denote all SQL Server supplied stored procedures.
Rule (RL)	<i>rl_[tablename]_[colname]</i>	This prefix denotes that the object is a SQL Server Rule. SQL Server rules are used to define (specify) the domain for a given column (instance) or to set the rule for a custom data type(inherited).

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 24 of F-1-65
Effective Date May 31,1994**

Trigger (TG)	<i>tg_ [tblname]_ [xx]</i>	<p>The prefix denotes that the object is a SQL Server Trigger. Triggers are used within SQL Server to perform function based upon a user activated event. Triggers can be activated by the following actions: Insert, Update, and Delete. Triggers are often used to enforce Referential Integrity (RI), or to update summary values.</p> <p>The source (sql scripts) should be kept in separate files-- one for each type of trigger, per table.</p>
	<i>_dl</i>	Delete Trigger
	<i>_up</i>	Update Trigger
	<i>_in</i>	Insert Trigger
Temporary Table (TT)	<i>tt_ [tblname]</i>	<p>This suffix denotes a temporary table. Temporary tables are used within SQL Server to hold sorted data, grouped data, and to speed joins of complex queries. All temporary tables, that are created must begin with the '#' (pound) sign. This indicates to SQL Server that the object is to be created in the TEMPDB database, and should be destroyed after the stored procedure completes or the user exits the database. If a table needs to be used for a short period of time, but longer than a single session or execution of a stored procedure, it should be created without the '#' sign, so that SQL Server will not destroy the table automatically. This means that the creator of this type of short-term table is responsible to drop the table explicitly.</p> <p>Example:</p> <pre>create table tt_batch_rpt ...[use the table]... drop table tt_batch_rpt</pre>
View (VW)	<i>vw_ [tblname]_ [function]_ [xxx]</i>	<p>This prefix denotes that the object is a SQL Server View. Views are used for two main reasons: to control what data a user is able to see, and to simplify (hide) joins.</p>

2.1.4 Variable Naming Conventions in SQL Server

Local Variables

All local variables should will begin with the @ sign. Local variables are generally used within stored procedures to control processing logic, or to handle arguments passed from the calling application. All local variables will be named as follows:

- @ + Column name, or
- @ + PB variable name

Global Variables

Global variables are used to store information regarding system activity. All global variables start with two @@ signs. Global variables shall not be created by users, but may be referenced by them.

2.1.5 PowerBuilder Library Categories

There are two categories of PowerBuilder Libraries used on the QUEST project:

1. **QUEST Stub (STUB):** Navigational related libraries that contain the QUEST application object as well as any windows and menu objects, common libraries containing function objects used throughout the application.
2. **QUEST Release (RLS):** Release related libraries containing objects constructed for a given release.

Size: PBLs should be kept to a maximum size of 40 objects or 800K. If these size limits are reached, a new PBL should be created with the next sequential number.

3.0 Look and Feel (static)

3.1 General Development Standards

This section contains specific information regarding application development on the QUEST project using PowerBuilder, SQR or any other tools which produces information displayed on a report or CRT.

Windows Interface Standard

All development will be compliant with the Windows Interface Standard. This standard will identify the types of events which should be coded, and standard responses to these events.

Command Button Captions

All command button captions will use Windows standard terms such as SAVE, INSERT, CLOSE, etc..

Button Bar

All command buttons on a Document Sheet will appear at the top of the window in the button bar.

Button/Menu Options

All command buttons on a Document Sheet g_doc_sheet will have an equivalent menu option.

Mouseless Operation

The application must be able to be used without a mouse.

Mixed Case Usage for Data

All data entered into QUEST in will be stored and displayed in **UPPER** case. This will be enforced by utilizing the PowerBuilder DataWindow extended column definitions edit style case function indicating **UPPER** case only. The only exception to this will be for description columns (class name 'desc'), these will allow for mixed case.

3.2 Fonts

Arial, 14pt. font size will be used for all text
All label text will be **Bold**;
All entry columns will be Normal;
All group box labels will be *Italics*

3.3 QUEST Window Types

All QUEST windows will be one of the following types:

3.3.1 Document Sheets

All document maintenance will be performed on a Document Sheet. A document sheet will be created by inheriting from one of the QUEST Stub document maintenance sheet window superclasses (`w_sc_doc_sheet`, `w_sc_doc_sheet_tab`, or `w_sc_master_det_sheet`). The document sheet will contain at least one datawindow in which to perform the maintenance.

This window will be opened from either the document menu and/or a selection window. Careful management of several global variables must be performed on a document sheet. An existing functional document sheet should be used as a guide when creating new document sheets.

3.3.2 Select Window

All document selection will be performed on a Select Window. A select window will be created by inheriting from one of the QUEST Stub select window superclasses (`w_sc_select`, `w_sc_select_criteria`). The select window will contain at least one datawindow in which to perform the selection.

This window will be opened from either the document and/or file open menu selections.

Careful management of several global variables must be performed on a selection window. An existing functional selection window should be used as a guide when creating new selection windows.

3.3.3 Response Window

Response windows may used as required. A response window will be created by inheriting from the QUEST Stub response window superclass (w_sc_more) or creating a 'new' response window.

3.4 Reports

3.4.1 Standard Report Layout

Page Size	Portrait 8 1/2" x 11" (preferred)
	Landscape 11" x 8 1/2"
Font Size	not less than 8 point
	not more than 16 point
Typeset	TBD

3.4.2 Standard Report Header

The standard report header will contain the following information:

Report Number: (left margin of first line)

Contains the following information.

Unit Process Number - Unit process number (without periods) which generates the report.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 29 of F-1-65
Effective Date May 31,1994**

Separator - "-"

Report Period - Indicates report frequency

D Daily
W Weekly
M Monthly
Q Quarterly
S Semi-annual
A Annual
R On-Request

System: (first line - centered) "QUEST - " + Subsystem (ie. Procurement, System Management)

Report Date/Time: (Right Margin) Date/time the report was generated.

Report Title: (Centered) Title of the Report

Page Number: (Right Margin) format "page:

Column Headings: as appropriate - blank line should separate the report title from the column headings.

Single Line: separates the column headings from the report detail (delineates the column limits within the report detail - see example)

3.4.3 Standard Report Body

When a report generation returns no data (NULL), a statement saying, "No Records Selected", should be displayed and/or printed within the report body. Also, it may contain summaries at various levels.

3.4.4 Standard Report Footer

No standard footer defined.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 30 of F-1-65
Effective Date May 31,1994**

3.4.5 Standard Report Trailer

The Standard report trailer will contain a statement identifying the end of the specific report.
(i.e., End of Report - AP-001)

3.4.6 Standard Banner Page

To be added later

3.4.7 Sample Report

To be added Later.

4.0 DataWindows

4.1 DataWindow Standards

Of the four data sources for DataWindows in PowerBuilder, only the stored procedure data source will be used.

4.1.1 Creating Update/Display DataWindows

DataWindows created for QUEST will either serve for update or display purposes.

All DataWindows, whether for update or display, will utilize a stored procedure for retrieval. The retrieval stored procedure may or may not use arguments, depending on the required retrieval.

Update DataWindows will always 1) utilize a maintenance stored procedure to perform the updates and 2) allow updates must be set.

Display DataWindows do not require any update stored procedure or concurrency consideration as they are used for display only.

4.1.2 Common Standards of DataWindows

All DataWindows share common standards in the following areas:

Assigning Borders in DataWindows

All SLE entry columns in freeform DataWindows will have shadowbox borders.

Using Format Styles in DataWindows

Format Styles may be used as appropriate.

Using Edit Styles in DataWindows

Edit style objects may be used as appropriate.

The type and volume of data being entered or displayed should be taken into consideration when using an Edit Style. Note: DropDownListBox should be used for code tables. These may be loaded using the QUEST STUB standard function `f_load_dw_ddlb`.

Using Validation Expressions in DataWindows

Validation expressions will not be used in DataWindow objects. All validation of columns will be performed using validation functions.

Using Defaults in DataWindows

Defaults will be assigned in DataWindow objects. These will match the defaults assigned in SQL Server.

Assigning Accelerator Keys in DataWindows

Accelerator keys may be used in DataWindows as appropriate.

Accelerator keys for entry columns will be coded in the entry column object; the label of the entry column will reflect the accelerator key by underlining the appropriate key in the label.

Example: **PO Number: 1001**

Property Settings

Vertical scroll bar: Enabled
Horizontal scroll bar: Disabled
Optional title
Optional Border

4.1.3 FreeForm DataWindows

FreeForm DataWindows will be used for creation, modification and deletion of table entries. Will be used to create, modify and delete single rows in the table.

When information is being input from source documents, the DataWindow will be designed to match the hard copy source document as closely as possible.

When information is not being input from source documents, the DataWindow will be designed in as logical a format as possible.

When the information to be entered will require more than one screen, the DataWindow will be designed as a single DataWindow rather than multiple DataWindows. During data entry the user will tab to each successive column. When the next column is off the screen, the window will automatically scroll up to display the column. DataWindows will have a vertical scroll bar but no horizontal scroll bar; data should be arranged such that no horizontal scrolling in the window is necessary.

4.1.4 Tabular DataWindows

Tabular DataWindows will be used for creating, modifying, deleting and displaying multiple rows from a table.

Will be used to create selection windows. Will be used to create, modify and delete multiple rows in the table. In this case, there will likely be several PushButton objects such as "Insert", "Delete", "Save" and "Cancel" from the QUEST Stub to act on the DataWindow. Tabular DataWindows will have a horizontal scroll bar (if necessary);

When a datawindow is designed for selection purposes, the tab order of the columns should be set to zero. This will improve the visual quality of the selection datawindow.

4.1.5 Grid DataWindows

Grid DataWindows will Not be used.

4.1.6 Using Extended Column Definitions in PowerBuilder

Extended column definitions WILL NOT be used in the database painter. Database painter extended column information is not inherited by a datawindow when a stored procedure is used for retrieval; therefore there is no advantage to maintaining extended column information in the database painter.

Extended column attributes within a datawindow will be utilized as necessary. This includes setting the following attributes as required:

Border

Display format: must use default format or a DBA approved user-defined format

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 34 of F-1-65
Effective Date May 31,1994**

Edit Style

Default: default values will be set up to match any defaults on the back end.

Labels and headings will be input directly into the datawindow layout.

Validation **WILL NOT** be addressed in the extended column attributes.

Display formats will be identified and established during the development process.

Labels and headings will be descriptive, using the Excelerator element labels and headings where they exist. Abbreviations will be the QUEST standard.

5.0 Error Handling

5.1 Overview

Error messages may result from either: 1) Errors found on the Server (SQL Server or User defined errors via the RAISEERROR function) or 2) Errors found on the front end. Each of these types will be handled in a different manner.

5.2 SQL Server Error Handling on the Back End

The SQL Server function which allows for error handling is called "RAISERROR". The syntax for the raiserror function is:

RAISERROR integer_expression "text message for error" [@local_variable]

The RAISERROR function will be always be called in a maintenance stored procedure. It will be called to 1) indicate a successful update, and return the new `concur_tm` value; 2) indicate a concurrency error and return the concurrency error code; and 3) indicate an error has occurred, and return the error number. The only exception to this is when a SQL Server error occurs, the RAISERROR is performed by SQL Server.

The format of the "text" portion of the RAISERROR message is as follows:

<u>Component</u>	<u>Size</u>	<u>Columns</u>
<code>sys_id</code>	4	1-4
<code>table_name</code>	30	5-34
<code>proc_type</code>	3	35-37
<code>error_desc</code>	255	38-292

When trapping for context errors, use the following method:

```
IF @prge_cd = "P" AND @stat_cd <> "T"
BEGIN
    EXEC @error_code = sp_showerror 20023, @table_name, @proc_type, @msg output
    RAISERROR @error_code @msg
    RETURN
END
```

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

Document	WHC-SD-GN-CSWD-30072
Appendix	F
Revision Number	0
Page	36 of F-1-65
Effective Date	May 31,1994

When trapping for SQL Server errors, use the following method:

```
BEGIN TRAN
UPDATE pt_QUEST_user
SET concur_tm = @new_concur_tm,
    passwd_chng_dt = @passwd_chng_dt,
    pin_num = @pin_num
WHERE suid = @suid
IF (@@error < > 0)
BEGIN
    ROLLBACK TRAN
    RETURN
END
COMMIT TRAN
```

When calling a stored procedure from a stored procedure, use the following:

```
EXEC @ret_status = lp_adduser @loginame, @loginame, @grp_name
IF (@ret_status < > 0)
BEGIN
    RETURN
END
```

5.3 SQL Server Error Handling on the Front End

Errors that originate on the Back End (SQL Server or User defined errors via the RAISEERROR function) will be automatically handled by the QUEST STUB on the front end. This includes the display of the error message, as well as the ROLLBACK of the SQL transactions.

Note: 1) An error is always returned, however, the error may indicate a successful update, context error or SQL Server error. 2) The static text in the window super classes (lower left hand corner) should be changed to reflect the unit process that defines the window, using UPPER case only. This text is displayed as part of the error message from the back end.

5.4 Front End Error Handling

Errors found during processing on the front end (i.e. column validations, context errors, etc.) will be handled by displaying a messagebox.

5.5 Message Box

Message Boxes will be utilized for displaying informational and error responses to users as a result of processing front end error conditions only. Message Boxes will be used to display Information. Message Boxes should contain only command button controls.

Reference page 128-130, 7.1.4 Message Dialogs, "The Windows Interface" by Microsoft, for message display standards.

Message Box Standards

Information Message Boxes will be used to inform the user about results of commands. The user acknowledges messages by clicking the OK button. The Information Message Box will use the standard contents as described on page 129 of the "The Windows Interface" by Microsoft.

Warning Message Boxes will be used to inform the user of an error condition or situation which requires a decision to be made by the user before proceeding. The Warning Message Box will use the standard contents as described on page 129 of the "The Windows Interface" by Microsoft. In addition to these standards, the Warning Message Box will contain an Error Code and Description.

Examples:

- Verifying the deletion of a record.
- Validation of Charge Code with option to ignore.

Critical Message Boxes will be used to inform the user that a serious exception condition has occurred and that some action must be taken to correct the situation. The Critical Message Box will use the standard contents as described on page 129 of the "The Windows Interface" by Microsoft. In addition to these standards, the Critical Message Box will contain an Error Code and Description.

Examples:

- Purchase Order not on file.
- Invalid Storage Location.
- Hardware failure detected.

The text portion of all Message Boxes should contain detailed information which adequately communicates the message. If the Message Box indicates that an error has occurred, the message should be constructed using the approved set of error class words.

5.6 MicroHelp

MicroHelp will be used to describe the functionality of a menu item which has been focused on by the user. MicroHelp will also be used to indicate a computer function is performing.

Examples:

Saving.....

Opening Window....

6.0 Navigation (dynamic)

6.1 Overview

Two important aspects of navigation will be discussed here: 1) General navigation concept; and 2) interaction of navigation and security.

6.2 General Navigation Concept

Navigation within the QUEST application is accomplished at several levels:

1. QUEST has Subsystems - Subsystem Panel
2. Subsystems have Documents - Document List Window
3. Documents have Associated Documents - Associated Document List Window

The user will make a selection at each level to navigate to windows to perform functions.

6.3 Interaction of Navigation and Security

All navigation within the QUEST application is dynamically controlled by the security tables that are maintained within QUEST. A users authorized navigation path is based on the security rights the user has. Only those navigation paths authorized will be available for selection.

When a user logs into the QUEST application, a connection/login attempt is made to the QUEST server. If successful, the users security related information is read into memory. This information includes the subsystems, documents, windows and CRUD (create, read, update and delete) rights granted for each. At each layer of navigation, the users security is taken into account.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 40 of F-1-65
Effective Date May 31,1994**

The users selections will only include those that are authorized with a minimum of read access at the following levels:

Power Panel (Subsystems)

Document List

Associated Document List

Window: only options allowed based on CRUD will be enabled.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 41 of F-1-65
Effective Date May 31,1994**

7.0 Inter System Interface Standards

7.1 Overview

To be defined later
DBA to complete this section

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 42 of F-1-65
Effective Date May 31,1994**

8.0 Application Optimization

8.1 Overview

To be defined Later.

9.0 Help Standards

9.1 Steps in Developing Help

1. Write the help document in Microsoft Word for Windows using the Doc-To-Help or RoboHelp format styles.
2. Assign index entries as necessary. If new indexes are assigned to the document, remove and rebuild the indexing.
3. Update the Table Of Contents and Index.
4. Make the document into a help file with Doc-To-Help. Make sure to run the Doc-To-Help Diagnostics and correct any errors.
5. Create a new report of the Context Map Numbers if necessary.
6. Distribute the new help file.

9.2 Tools In Developing Help

Word Perfect for Windows 2.0 will be the tool for creating and editing the help document, adding the control codes and converting the document into Rich Text Format. To ensure help standards and data integrity, no other word processor will be used.

Doc-To-Help 1.1 Hypertext Word Processing for Microsoft Word for Windows will be used to create and maintain the help documentation. All modifications to the documentation must be made through Doc-To-Help.

The help file in Rich Text Format will be compiled using the Microsoft Help Compiler 3.0 from the Microsoft Software Developer's Toolkit 3.0. To ensure help standards and data integrity, no other help compiler will be used.

PowerBuilder 3.0 will be used to develop the QUEST application and all calls to the help system.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 44 of F-1-65
Effective Date May 31,1994**

9.3 Help Document Standards

Text in the help document should be formatted according to the following table:

Text	Doc-To-Help Style
Chapter Heading	heading1
Section Heading	heading2
Paragraph Heading	heading3
Paragraph Sub-Heading	heading4
Body Text	body

For specific formatting standards refer to the Doc-To-Help Reference Manual. It provides specific standards and information for a variety of issues including graphics, indexes and page formats. As a general rule, Doc-To-Help defaults will be used as the QUEST Help Standards.

9.4 Filename Conventions in Help

Doc-To-Help will create the following documents when creating on-line help:

Filename	Description
QSTHELP.DOC	Help document saved in Microsoft Word format
QSTHELP.RTF	Help document saved in Rich Text Format (RTF).
QSTHELP.HPJ	Help project file used to build the help resource file by the Microsoft Help Compiler saved in DOS text format.
QSTHELP.HLP	Compiled help resource file created by the Microsoft Help Compiler.

9.5 Context-Sensitive Help

Context-sensitive help may be available for columns in datawindows, windows and buttons.

1. To assign context-sensitive help to a column, control or window:
2. Enter the help topic text into the help document using Doc-To-Help.
3. Make the help document into on-line help.
4. Generate a new Show Context Mapping report.
5. For columns and controls, add the map number from the report as the tag value for the object. For windows, change the static text st_context_id to the map number.

It is important to note that these map numbers are maintained in the Help Project File (HPJ). As topics are added and deleted from the document, the HPJ file updates the topics and their mappings. If the HPJ file is deleted, all topics will be assigned new map numbers which may or may not be the same as the originals. Therefore, it is extremely important to prevent the HPJ file from being deleted.

9.6 Accessing Help

Help will be available in the following methods:

- selecting Help from the menu with the mouse;
- selecting Help from the menu with the keyboard (Alt-H);
- pressing F1 hotkey;
- pressing Shift-F1.

The help menu will appear as follows:

Contents will display the index of topics in the QUEST Help System.

Help For This Window will display the window-specific help topic.

How to Use Help will display Windows' "How to Use Help" help system (WINHELP.HLP).

About QUEST... will display information about QUEST.

9.6.1 PowerScript Code

To display the help contents, the PowerBuilder ShowHelp() function will be used.

To display context-sensitive help, the user-defined f_displayhelp() function will be used. F_displayhelp() uses the Windows 3.1 API WinHelp() function. WinHelp() provides the ability to directly retrieve a topic with a specific map number; ShowHelp() simulated context-sensitive help by doing a keyword search - if two topics shared a keyword, the first would always be retrieved.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 47 of F-1-65
Effective Date May 31,1994**

9.7 Reference Materials

Refer to the following documentation for more information on creating on-line help documentation:

**Microsoft Software Developer's Toolkit
Doc-To-Help Reference Manual**

10.0 Development Tools

10.1 Overview

The following is a list of software tools each member of the development team should have.

DOS 6.0
LANMAN Enhanced 2.1
Windows 3.1
PowerBuilder 3.0
Word Perfect for Windows 5.2

11.0 Development Techniques

11.1 Overview

This section contains various techniques that will be used during development activities.

11.2 Creating, Modifying, and Dropping Stored Procedures

The handling of SQL Server permissions to execute stored procedures has been integrated into the system. The process of granting and revoking stored procedures is completely data-driven. Because of this, care must be taken to ensure that the database accurately represents all the stored procedures utilized by the system, and that the actual privileges in effect mirror what is represented in the database.

Within QUEST, the stored procedures required by each document are identified on the document maintenance screen. It is essential that this list be kept up to date, and that proper privileges are always in effect on the listed stored procedures. To accomplish this, additional actions are required every time a stored procedure is added, modified, or dropped.

When adding a stored procedure:

Make sure the stored procedure added is assigned to each of the documents which requires the procedure.

When modifying a stored procedure:

Modifying a stored procedure should not require any changes to data, but will require that the privileges on the stored procedure be re-established. The process of modifying a stored procedure requires the procedure to be dropped and re-created. This means that any privileges which had been granted on the procedure will have been lost.

To re-establish the privileges on a stored procedure, access the document maintenance screen for each document which the stored procedure is assigned to and press the SAVE button. You are not required to change any data on the screen; pressing the SAVE button will always re-establish the privileges on all the stored procedures for the document. It is helpful to execute a query to determine which documents a stored procedure is assigned to.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 50 of F-1-65
Effective Date May 31, 1994

When dropping a stored procedure:

When a stored procedure is dropped, all the privileges granted to the procedure will be lost. However, the stored procedure will still be assigned to any documents it was related to. Whenever a stored procedure is permanently dropped, the stored procedure must be de-assigned from any documents it was related to. Again, it is helpful to execute a query to determine which documents the stored procedure is assigned to.

Note on common procedures: Common procedures (all procedures beginning with the characters 'cp_') are granted to all QUEST users, and are not assigned to any individual documents. No data changes are required when working with common procedures. However, whenever a common stored procedure is added or modified the stored procedure must be granted to public.

12.0 QUEST Stub Overview

12.1 Overview

The QUEST Stub is a collection of PowerBuilder Libraries and objects that comprise the backbone of the QUEST application. This includes the application object (QUEST), global structures, standard functions and superclass menus and windows. The QUEST Stub includes script that enforces application standards, as well as reducing the development time required for new releases.

12.2 PowerBuilder Global Structures

12.2.1 Structure: s_gvs - Global Variable Structure

Type	Name	Description
window	w_curr_active	current active window
string	curr_sub_system	current subsystem
string	curr_document	current document
long	err_code	SQL error code
string	err_debug_code	SQL error debug code
string	err_msg	SQL error message
w_QUEST_mdi	mdi_frame	mdi_frame
boolean	new_sheet_ind	new sheet mode indicator
boolean	new_doc_ind	new document mode indicator
boolean	closequery_cancel_ind	cancel indicator from closequery dialog
string	key_message[10]	message array for key values

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 52 of F-1-65
Effective Date May 31,1994**

boolean	select_cancel_ind	cancel indicator from select dialog
int	concur_tm_msg_num	SQL error number for successful updates
string	concur_col_name	name of column for concurrency checks
string	comm_message[10]	message array for misc. values
string	last_save_type	indicates last save type performed SAVE button
string	closequery_msg_title	title to be displayed in closequery dialog
string	closequery_msg_text	text to be displayed in closequery dialog
int	mdi_sheet_cnt	number of sheets currently open
string	mst_proc_return_str	return data from SQL Server
string	doc_select_mode	association vs. document selection
string	crud_message	crud for current document
string	win_directory	directory path where windows is located
string	curr_version_number	current software version number
int	win_list_menu_pos	menu position for sheet list
window	mdi_sheet[10]	array of mdi sheets currently open
int	next_sheet_pos	next position for sheet
string	help_file	help file name
string	error_col_name	column of the most recent SQL rule error

12.2.2 Structure: s_sec - Security Structure

Type	Name	Description
string	subsys	subsystems
string	doc[50]	documents
string	crud[50]	crud
string	window[50]	windows

12.2.3 Structure: s_usr - User Structure

Type	Name	Description
int	gid	group id
string	grp_name	group name
string	payrl_num	payrole number
string	sig_lvl_cd	signature level code
string	loginame	login name
string	pin_num	pin number
boolean	su_ind	superuser indicator (QUEST_SA)

12.3 PowerBuilder Standard Functions

The QUEST STUB standard and support functions are global functions which were created to standardize and simplify calls to common routines. Standard functions are typically used in descendent windows, while support functions are typically used by the ancestor (superclass) windows. The functions may, however, be used as required.

12.3.1 Standard Function: f_retrieve

Purpose

Retrieve data into the specified datawindow via the RETRIEVE() function.

Usage

f_retrieve(dw_name)

Where

dw_name = datawindow name

Returns

The number of rows retrieved. Returns -1 if an error occurs.

12.3.2 Standard Function: f_load_dw_ddlb

Purpose

Load a ddlb edit style column within a datawindow

Usage

f_load_dw_ddlb(dw_name, col_name, sp_name)

Where

dw_name = datawindow name

col_name = column name

sp_name = stored procedure name for code table retrieve

Returns

The number of items loaded

12.3.3 Support Function: f_sp_error

Purpose

Checks SQLCA return code, displays error message if necessary, and returns success or failure of embedded stored procedure calls

Usage

f_sp_error()

Where

N/A

Returns

SQLCA.SQLCODE

12.3.4 Support Function: f_get_dw_col_dtype

Purpose

Return the datatype of the specified datawindow column

Usage

f_get_dw_col_dtype(dw_id, col_name)

Where

dw_id = datawindow name

col_name = column name

Returns

string - where: n = number, c = string, d = date

12.3.5 Support Function: f_opensheet

Purpose

Open new sheet w_name, override the system cascade position and reposition the window

Usage

f_opensheet (string w_name)

Where

w_name = name of the window to open a sheet for.

Returns

N/A

12.3.6 Support Function: f_parse_error_msg

Purpose

Set gvs.err_code

Parse error string returned from dberrormessage() and set gvs.err_debug_code and gvs.err_msg

Usage

f_parse_error_msg(error_str, error_code)

Where

error_str = dberrormessage()

error_code = dberrorcode()

Returns

N/A (0)

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 57 of F-1-65
Effective Date May 31,1994**

12.3.7 Support Function: f_displayhelp

Purpose

Display context-sensitive help. ShowHelp() function doesn't support calling topics by map number so the external WinHelp() function must be used.

Usage

f_displayhelp (tag)

Where

string tag = object tag string

Returns

N/A

12.3.8 Support Function: f_toggle_modified

Purpose

This is used to track the modified status of a row within a datawindow.

Usage

f_toggle_modified (w_name, row_num);

Where

dw_name = datawindow name

row_num = row number

Returns

N/A (0)

12.3.9 Support Function: f_toggle_select

Purpose

This is used select and unselect a row within a datawindow as it is clicked.

Usage

f_toggle_select (dw_name, row_num)

Where

dw_name = datawindow name

row_num = row number

Returns

N/A (0)

12.3.10 Window Function: f_set_enable

Purpose

This is used to enable/disable control bar and menu item objects (new, save and delete only). This function will consider security requirements when the enable/disable request is made. Only those objects which the current user has the rights to use will be enabled.

Usage

f_set_enable (object_cd, enable_ind)

Where

object_cd = 'New', 'Save', or 'Delete' object
enable_ind = TRUE or FALSE

Returns

N/A (0)

12.4 QUEST Stub Superclass Menus

12.4.1 m_QUESTION_main

This menu is used on the QUEST MDI FRAME window. This menu is only displayed when QUEST is running, and no sheets are currently open. This menu includes the following menus items:

12.4.2 m_sys_sc

This menu is used on all QUEST document sheet windows. This menu includes the following menus items:

Document

Association

Options

12.5 QUEST Stub Superclass Windows

Several superclass windows have been developed for use on the QUEST application. Any window that is developed as part on QUEST will inherit from one of the following superclass windows, exceptions will be reviewed and approved by the development teams.

12.5.1 w_sc_select

The Window Superclass Select object provides a response window to allow the user to select a row to maintain, or select to create a new row.

Steps in using:

1. Create a new window by inheriting from w_sc_select.
2. Replace the datawindow object with the required datawindow. This datawindow should contain any columns required by the user to identify the row, and must include a column to be used as the key for retrieval of the row for maintenance.
3. code the following function call in the open event
f_waf (search_col_name, key_col_name, w_to_open)

where string search_col_name = Name of the column by which the search will occur

string key_col_name = Name of the column that will be the key
string w_to_open = Name of the maintenance window to open

4. Code the appropriate retrieve statment on the open event
5. Center the window (using window style position).

User Defined Events:
None

12.5.2 w_sc_doc_sheet

The Window Superclass Document Sheet object provides a MDI sheet window to allow the user to maintain a row. Steps in using:

1. Create a new window by inheriting from `w_sc_doc_sheet`
2. Replace the datawindow object with the required datawindow. This datawindow should contain any columns required by the user to maintain the row, and must include the key column as well as a `concur_tm` column.
3. Code the following function call in the `ue_open_window` event

`f_waf (subsystem_name, doc_name, all_or_nothing_ind, sp_primary_name,
w_title, w_title_col_name, w_select_name)`

where string `subsystem_name` = name of the subsystem to which the document being maintained belongs

string `doc_name` = name of the document

boolean `all_or_nothing_ind` = indicator whether the update will be all or nothing

string `sp_primary_name` = name of the stored procedure used for the updates

string `w_title` = window main title

string `w_title_col_name` = name of the column from which the window title will be built

string `w_select_name` = Name of the select window to be used. If no select window is used, then use the empty string "".

4. Code other events (including user defined) as required

User Defined Events:

<code>ue_old_doc</code>	any code required when an existing document is to be opened
<code>ue_new_doc</code>	any code required when a new document is to be opened
<code>ue_open_window</code>	any code required when a new or existing document is to be opened
<code>ue_successful_save</code>	any code required when a successful save is completed
<code>ue_clicked_prior</code>	any code required to be executed prior to the save button clicked script. setting the instance variable 'prior_result_code' to 1 will prevent the save button script from executing.

The following user events should not be used. These events are for internal superclass processing only:

ue_cb_delete, ue_cb_insert, ue_cb_close, ue_cb_save, ue_m_new, ue_m_open, ue_m_print, ue_association, ue_m_document, ue_set_title, ue_open_select

12.5.3 w_sc_doc_sheet_tab

The Window Superclass Document Sheet object provides a MDI sheet window to allow the user to maintain rows in a tabular style datawindow. This window is inherited from w_sc_doc_sheet, refer to w_sc_doc_sheet except as follows:

Steps in using:

1. Create a new window by inheriting from w_sc_doc_sheet_tab
2. Replace the datawindow object with the required datawindow.
3. Code the following function call in the ue_open_window event
4. f_waf (subsystem_name, doc_name, all_or_nothing_ind, sp_primary_name, w_title, w_title_col_name, insert_rowatend_ind)

where string subsystem_name = name of the subsystem to which the document being maintained belongs

string doc_name = name of the document

boolean all_or_nothing_ind = indicator whether the update will be all or nothing

string sp_primary_name = name of the stored procedure used for the updates

string w_title = window main title

string w_title_col_name = name of the column from which the window title will be built

boolean insert_rowatend_ind = indicator whether to insert new rows at the end of the datawindow, or at the current position

5. Code other events (including user defined) as required

User Defined Events: (See w_sc_doc_sheet)

Additional user events: Superclass processing only:

ue_cb_delete_item

13.0 Display Format Definitions

13.1 Introduction

When creating DataWindows that use stored procedures, the column definitions do not default to the extended column definitions. Any extended column definitions must be manually applied within each data window.

The following Display Format Definitions have been created:

13.1.1 Format: Amount

Type: Number

Format: ###,###,##0.00' ';###,###,##0.00'-'

Reason: Standardize the display of monetary amounts system wide.

Use: Should be used for all columns in a datawindow which display a monetary amount.

13.1.2 Format: Date

Type: datetime

Format: mm/dd/yyyy

Reason: Standardize the display of dates system wide.

Use: Should be used for all columns in a datawindow which display a date.

13.1.3 Format: Datetime

Type: datetime

Format: mm/dd/yyyy h:mm:ss AM/PM

Reason: Standardize the display of datetime values system wide.

Use: Should be used for all columns in a datawindow which display a datetime value.

13.1.4 Format: Percent

Type: Number

Format: #0.000' ';#0.000'-'

Reason: Standardize the display of percentages system wide.

Use: Should be used for all columns in a datawindow which display a percentage.

**QUEST2: SA/RELEASE1 SUPPORTING
DOCUMENTS DELIVERABLE SET
DESIGN STANDARDS
(P+ DOCUMENT #380)
Attachment 1 - Coding Standards**

**Document WHC-SD-GN-CSWD-30072
Appendix F
Revision Number 0
Page 64 of F-1-65
Effective Date May 31,1994**

13.1.5 Format: Time

Type: datetime

Format: h:mm:ss AM/PM

Reason: Standardize the display of times system wide.

Use: Should be used for all columns in a datawindow which display a time.

14.0 User-Defined Data types

14.1 Introduction

In addition to the system supplied data types (int, float, char, etc.) available within Sybase (SQL Server), additional data types may be created as required (user-defined datatypes). The following User-Defined Data types have been created:

14.1.1 Data type: dt_datetime

Type: datetime

Reason: Validation is required on all dates entered to ensure that the year is in the 20th or 21st century. A user-defined data type with a rule bound to it can enforce this validation system-wide.

Description:

'dt_datetime' will behave as follows:

Entered year value: Stored year value:

4 character year 'nnnn', < 1900 Error

4 character year 'nnnn', > 2099 Error

Use: Should be assigned to all columns in the system which utilize the 'datetime' datatype. A few exceptions may be identified.

14.1.2 Data type: dt_ind

Type: char(1) NOT NULL

Reason: All indicators (flags) in the system should behave the same. A user-defined data type can standardize the acceptable values and validations required.

Description:

'dt_ind' will behave as follows:

Acceptable values:

"Y", "y", "N" or "n"

Use: Should be assigned to all columns in the system which are of the 'indicator' data element class. A few exceptions may be identified.

1.0 PURPOSE

This document describes a prototyping approach for producing certain unit processes in Release 1 of the QUEST2 project. It includes the activities and deliverables related to the physical building of the system that will be done to support the detailed definition of each unit process. This document specifies the steps and guidelines for prototyping unit processes using an approach called "evolutionary prototyping".

2.0 BACKGROUND

This document covers the usual activities necessary to accomplish evolutionary prototyping. To avoid confusion and to reach the best performance with this approach, it is very important that the following deliverables related to the overall system are completed before starting:

- The system architecture phase must be fully completed with approved deliverables:
 - 250 Definition of System Functions
 - 320 Definition of Data Elements
 - 340 Conversion Rules (for conversion unit processes)
 - 360 Functional Standards
 - 370 Operating Environment
 - 380 Design Standards
- The functional data model has completed the first two steps and is ready to start the third step:
 - First step: Translation of the conceptual model into a functional data model
 - Second step: Optimization of the functional data model based on the "250 Definition of Functions"
 - Third step: Fine tuning of the functional data model based on the functional specifications of each unit process.
- The naming convention for the physical units is defined and ready to use.
- Common processes (code to be shared) are mostly identified and ready to develop. (The list will be completed during the process of prototyping.)

The 1st release of the QUEST2 system is composed of two prototypes and ten individual unit processes. The development of the ten individual unit

processes will not be addressed in this deliverable. They will be developed using the traditional P+ methodology: design, construct, test. The two prototypes will consist of the following:

- Prototype 1

- UPQST11 Update Document
 - UPQST13 Input Audits
 - UPQST25 Maintain Conditions
 - UPQST28 Create Conditions
 - UPQST43 Maintain Actions
 - UPQST61 Update Tables
 - UPQST63 Add/Update Users

- Prototype 2

- UPQST22 Create Sub-Conditions
 - UPQST23 Consolidate Conditions
 - UPQST42 Query Conditions/Actions
 - UPQST77 Ad Hoc Query

The remainder of this deliverable will describe the process for developing these prototypes in terms of the deliverables that will be produced and roles (responsibilities) of the different parties involved.

3.0 ROLES

It is assumed that these personnel roles will be filled during the prototyping exercise (Note that one person may have more than one role):

Analyst/developer: Will analyze and develop the prototype

Functional/analyst: Will update the models (conceptual and functional data and process models); support the analyst/programmers in their understanding of the architecture; verify that the boundaries of each unit process are respected; ensure quality control of the developed functionalities.

Database expert: Will ensure that the accesses to the database are well performed; ensure that the rules described in the design standards (re:database and performance) are respected.

**EVOLUTIONARY PROTOTYPING
(P+ DOCUMENT #235)**

- Client-server expert: Will ensure the quality and performance of the technical components in accordance with HANTAG and/or HANTRB findings and other site standards.
- User interface expert: Will ensure the quality and performance of the user interfaces in accordance with the functional and design standards.
- User: Has the responsibility and authority to identify requirements, and to verify that the system functionality meets these requirements.
- IV&V Analyst: An independent analyst who is familiar with our System Development Lifecycle and understands the requirements addressed within DOE 1330-D, QR-19, P+, and CM 3-10. This individual must understand the process to verify and validate that everything that is done is in accordance with existing standards and guidelines.

Each of the "experts" will update the 360 (Functional Standards) and 380 (Design Standards) related to their domain of expertise, if needed.

4.0 TESTING

Testing for the evolutionary prototyping environment will include two forms; Unit and Prototype Testing. These are described in detail in the 032 Procedure on Testing, Section 6. For clarification, a brief description of each is provided below:

- Unit Testing: The testing of each and every object in the system. This testing is performed by the developers as the system is developed. Unit testing is documented on controlled forms and may be handwritten.
- Prototype Testing: Testing of each 490 Functional Specification. Test reports are developed and run by the users to validate the prototype. These will set the scope of the reviews with the customers.

5.0 PROCESS DESCRIPTION FOR EACH EVOLUTIONARY PROTOTYPING PROCESS

Attachment 1 provides a graphical representation of the prototyping processes and how they relate to the persons and deliverables involved in the process. The processes are:

- Get understanding of the unit process

Revisit the Definition of the Unit Process (250). Also, to ensure understanding of the context of its application, revisit the definition of any manual unit process that triggers this automated unit process.

Identify, on the Functional Data Model (460), the functional entities and functional links that are covered by the prototyping of this unit process (Functional view of a unit process).

Read the Definitions of the Functional Entities and Functional Links (470-480). Obtain the characteristics of all related data elements (definition, length, type, coding and editing rules, etc.) (320). Read the conceptual data model integrity rules for each entity (180) and relationship (190) corresponding to the functional entities.

Use the existing prototype (600-695) as input.

Functional Analyst and User Interface Expert will make sure that they:

- have a good understanding of this unit process;
- respect the scope of this unit process;

Project management:

Functional Analyst and User Interface Expert time is recorded as part of the 490-Functional Specifications and application code (Cycle 1) for this unit process.

- Structure the unit process.

Identify the operations contained in the unit process. Use the 490-template (see Attachment 2) to design the structure of the unit process.

Use the 360 (Functional Standards) to guide the functional structure of the unit process.

The functional/analyst will also update the process models (250) if needed. Since these are approved, the functional Analyst must prepare a

EVOLUTIONARY PROTOTYPING
(P+ DOCUMENT #235)

change request to explain the reason for the change and make sure the user accepts the change.

Deliverable :

Functional Specifications (490) - operation descriptions

Project management:

Functional analyst time is recorded as part of the 490-Functional Specifications and code (Cycle 1) for this unit process.

- Design screen drafts.

Use the 360-functional standards document as guidelines for the screen layouts. At this point, no "tables" are supporting the screens. The prototype at this time should consist mainly of screen mock-ups.

Deliverable:

Screen Layout (660) built with POWERBUILDER and a printed version (490)

Project management:

User Interface Expert time is recorded as part of the 490-Functional Specifications and application code (Cycle 1) for this unit process.

- First review with user

Present the screens to the users that support each unit process identified by the 490's.

Capture all comments on the 052 Prototype Review/Approval Form (Attachment 3). Draw on the screen printout any modifications requested by the users and build the 490 body with them:

- algorithms
- validations and derivations. Produce a list of messages that should be issued.
- criterions to use when selecting the data presented on the screen (and ordering sequence of lists)

EVOLUTIONARY PROTOTYPING
(P+ DOCUMENT #235)

- the frequency of use for each type of operation that can be done with this unit process and the variation of this frequency over time.
- Performance criteria: document expectations.

Following that meeting, the User Interface Expert will modify the screens.

Deliverable:

Completed 052 Prototype Review/Approval Form(s) with all comments.

- Update functional specifications

The Functional Analyst will use the 052 Prototype Review/Approval form generated in the user review to update the 490 Functional Specifications. Review updated 490 with the 360 Functional Standards to insure that functional integrity is maintained.

Deliverable:

Updated 490 Functional Specifications

- Design technical structure

The development team will use the updated 490 and the 380 Design Standards to map the technical design of each of the unit processes. This technical design framework or partitioning of the 490 will direct the developers on the different objects that need to be developed. It will also identify where data will reside in a client server environment with respect to the overall system design. This design work may result in a better or more efficient way of satisfying the users needs and may change the 490. These changes would be captured on a 052.

Deliverables:

540/550 Partitions of the System
052 Prototype Review/Approval Form

- Review specifications

Review the updated 490 with the users. Trace comments from the 052 from the user review to the updated 490 and promote the design changes from the development team for incorporation into the 490. If any additional comments are made, capture on an 052 for incorporation into the 490.

EVOLUTIONARY PROTOTYPING
(P+ DOCUMENT #235)

- Update functional specifications

The Functional Analyst will use the 052 Prototype Review/Approval form generated in the user review to update the 490 Functional Specifications. Review updated 490 with the 360 Functional Standards to insure that functional integrity is maintained.

Deliverable:

Updated 490 Functional Specifications

- Prepare prototype tests

Users prepare test cases exercising the functionality defined for the unit processes involved. They use the updated 490 as the basis for these test cases.

Deliverables:

User will prepare the 695 Prototype Test Cases

- Build front/back end

Review the technical design framework (partitioning) of the unit process.

Code the front end.

Code the data structure for the front end part if needed.

For each unit process, revisit and identify the preferred access paths on the functional data model. Document "fine-tuning" operations in the functional data model (460-470-480). Revisit the conceptual data model if needed (170-180-190). Changes to the conceptual data model should be logged on a change request (see 042/044 Change Request Log document).

Note: The analyst/developer should try to identify, as soon as possible, any negative impacts on system performance (e.g. Complex access paths, query results that would return too many occurrences, etc.). This may result in a re-design of the unit process.

Identify on the "functional view" grid (460) the functional entities that are created, updated, read, or deleted by the unit process.

The development team will approve the 490-Function Specifications of the unit process to indicate feasibility.

Responsibilities:

- Analyst/developer provides the information regarding the accesses desired in this unit process.
- Functional Analyst analyses and updates the models.
- Analyst/developer updates the functional specifications and the grid.
- Development team approves the 490
- Analyst/developer develops screens, tables, and code.

Deliverables:

- Approved (by system architect) Functional Specifications (490)
- Updated Functional View Grid (460)
- Updated Conceptual Data Models (170-180-190-320)
- Updated Functional Data Models (460-470-480)
- Updated Partitioning (540/550)
- Unit Testing (570)
- Data Structures and Tables (600-610-620)
- Screens (660)
- Code (670)

Project management:

- Analyst/programmer time is reported in 490-Functional Specifications and application code for this unit process.

• Integrate Objects

Combine objects that constitutes a unit process. Test and document these objects for unit testing. Prepare the Installation Instructions (720) for this unit process. Get Project Experts (User Interface, Database, Client Server) approval for these instructions.

Responsibilities:

EVOLUTIONARY PROTOTYPING
(P+ DOCUMENT #235)

- Analyst/programmer completes unit testing
- Analyst/programmer prepares the instructions
- Experts approves them

Deliverable:

- Unit Testing (570) for these objects
- Installation Instructions (720) for each unit process

Project management

Analyst/programmer time will be reported in the 490-Functional Specifications for this unit process

• Test/Review with User

Analyst/programmer presents the prototype to the user, IV&V analyst, and the QA Analyst. Comments and modifications will be written on the "Request for Review and Approval of Deliverable" form (052 of the Functional Specification for this unit process (490))

Analyst/programmer revisits the comments with the Functional Analyst to determine if it has an impact on the scope and the functional view.

Analyst/programmer will make the modifications to the prototype and also modify the documentation if needed. Unit testing will be performed again and the prototype will be re-presented to user for retest and approval.

When satisfied, user signs the review form. Prototype will be presented to System Manager for final approval of the Functional Specifications (052 of the 490).

This approval is not an official approval of the unit process since the functional testing activities are not yet done. This is an approval of the screen layout, table contents, and the specifications (490). One more iteration of the prototyping cycle may be needed to finalize any changes to the prototype that are within the scope of the 490 as approved to this point.

Responsibilities:

- Analyst/programmer presents to user and system manager prototype and specification documentation as well.

- System Manager approves the specifications.
- Functional Analyst reviews and approves the specifications

Deliverables:

Final Functional Specifications (490) and Approval Form (052)

Project management :

Analyst/programmer time will be reported in 490-Functional Specifications and code (cycles 2 or 3) for this unit process.

Once this cycle has been completed through the second review, any final updates are made to the 490 in the Update Functional Specifications process and follows through the prototyping cycle. The entire prototype will then be subjected to a another of formal prototype testing as described above. Final acceptance of the prototype will be provided and it will enter into the functional testing phase described in the 032 Procedures on Testing (Section 6).

All noted changes not part of the prototype scope will be recorded as change requests and incorporated into future release planning. If the System Owner determines that some new changes (i.e., changes that are not part of the 490 at that point) are critical, they may be incorporated into the prototype and a new round of updates and testing will occur. However, this work will be considered new project scope and will necessitate a change in project cost and schedule.

ATTACHMENT 1

Evolutionary prototyping during Release development

ATTACHMENT 2: 490 TEMPLATE

1.0 Purpose

2.0 Operations

2.1 Business Operation #1

2.1.1 Description

2.1.2 Frequency of use

2.1.3 Performance Criteria

2.1.4 Security

3.0 Screens

3.1 Screen/Operation Matrix

3.2 Screens Definitions

3.2.1 Physical Screen Layout

3.2.2 Screen Object Definitions

OBJECT	EVENT	DESCRIPTION
"...." Button	Clicked	Help Message

3.2.3 Screen Data Requirements and Validations

QUEST2: SA/RELEASE: DELIVERABLE SET

Document WHC-SD-GN-CSWD-30072

Appendix G

Revision Number 0

EVOLUTIONARY PROTOTYPING
(P+ DOCUMENT #235)

Page 13 of 14

Effective Date May 31, 1994

ATTACHMENT 3: 052 PROTOTYPE REVIEW/APPROVAL

Unit Process: _____

Description: _____

Release: _____

Cycle 1

Date: _____

Attendees:

Comments: (See attached pages)

Cycle 2

Date: _____

Attendees:

Approved (Y/N): _____ (Requires Signature)
Approved with Comments: _____ (Requires Signature)
Reviewed with Comments: _____ (See comments)
Change Request Required (Y/N): _____ (See comments)
Comments: (See attached pages)

Cycle 3

Date: _____

Attendees:

Approved (Y/N): _____ (Requires Signature)
Approved with Comments: _____ (Requires Signature)
Change Request Required (Y/N): _____ (See comments)
Comments: (See attached pages)

QUEST2: SA/RELEASE1 DELIVERABLE SET

Document WMC-SD-GN-CSWD-30072

Appendix G

Revision Number 0

EVOLUTIONARY PROTOTYPING
(P+ DOCUMENT #235)

Page 14 of 14

Effective Date May 31, 1994

COMMENTS

Cycle 1 Date:

#	POC	Comment Description

Cycle 2 Date:

#	POC	Comment Description

Cycle 3 Date:

#	POC	Comment Description