

# ornl

ORNL/TM-12799

**OAK RIDGE  
NATIONAL  
LABORATORY**

**MARTIN MARIETTA**

**Standards and Conventions for the  
Worldwide Port System (WPS)  
Regional Integrated Cargo  
Database (ICDB)**

J. P. Loftis  
L. F. Truett  
P. C. Shipe  
E. Z. Faby  
R. A. Ferguson  
J. Fluker  
J. W. Grubb  
B. R. Hancock

**MANAGED BY  
MARTIN MARIETTA ENERGY SYSTEMS, INC.  
FOR THE UNITED STATES  
DEPARTMENT OF ENERGY**

**DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED**

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

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.

## **DISCLAIMER**

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

**STANDARDS AND CONVENTIONS FOR THE WORLDWIDE PORT  
SYSTEM (WPS) REGIONAL INTEGRATED CARGO DATABASE (ICDB)**

J. P. Loftis  
L. F. Truett  
P. C. Shipe  
E. Z. Faby\*  
R. A. Ferguson\*\*  
J. Fluker\*  
J. W. Grubb\*  
B. R. Hancock\*

Energy Division  
Oak Ridge National Laboratory  
Oak Ridge, TN 37831

\*University of Tennessee, Knoxville  
\*\* Science Applications International Corporation, Oak Ridge, Tennessee

February 28, 1995

Prepared for the  
Product Management Office  
Office of the Deputy Chief of Staff for Information Management  
MILITARY TRAFFIC MANAGEMENT COMMAND  
Falls Church, VA 22041-5050  
under  
Interagency Agreement DOE No. 1405-1351-A1

Prepared by the  
OAK RIDGE NATIONAL LABORATORY  
Oak Ridge, Tennessee 37831  
managed by  
MARTIN MARIETTA ENERGY SYSTEMS, INC.  
for the  
U.S. DEPARTMENT OF ENERGY  
under contract DE-AC05-84OR21400

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

hwh

**MASTER**



## TABLE OF CONTENTS

ABSTRACT .....	vii
1. INTRODUCTION .....	1
1.1 BACKGROUND .....	1
1.2 PHILOSOPHY .....	1
2. CONSIDERATIONS FOR ESTABLISHING STANDARDS AND CONVENTIONS .....	5
2.1 PURPOSES OF STANDARDS AND CONVENTIONS .....	5
2.2 AREAS THAT NEED STANDARDS AND CONVENTIONS .....	5
2.3 STANDARD TOOLS FOR IMPLEMENTING THE USER INTERFACE .....	6
2.4 TERMS AND ABBREVIATIONS .....	7
2.5 PROJECT REFERENCES .....	10
3. SCREENPAINTING .....	13
3.1 OVERVIEW .....	13
3.1.1 Navigation .....	13
3.1.2 Modularity .....	13
3.1.3 Screen Types .....	14
3.1.4 Developmental Tools .....	14
3.2 MENU SCREENS .....	15
3.2.1. Functionality .....	15
3.2.2 Menu Layout .....	15
3.3 QUERY SCREENS .....	16
3.3.1 Overview .....	16
3.3.2 The Header Block .....	16
3.3.3 The Text Block .....	20
3.4 DATA DISPLAY SCREENS .....	24
3.4.1 Overview .....	24
3.4.2 Full-Page Display Screens .....	24
3.4.2.1 The Header Block .....	24
3.4.2.2 The Title Block .....	25
3.4.2.3 The Text Block .....	25
3.4.2.4 The Code Description Pop-Up Window .....	28
3.4.3 Additional Data Display Screens .....	30
3.5 SUMMARY REPORT SCREENS .....	30
3.5.1 The Summary Selection Screen .....	30
3.5.2 The Summary Query Screen .....	34
3.5.3 The Summary Display Screen .....	34
3.6 INPUT/UPDATE/DELETE SCREENS .....	38
4. NAVIGATION .....	41
4.1 ENTERING THE ICDB SYSTEM .....	41
4.2 ICDB USER ROLES .....	41
4.3 EXITING FROM THE ICDB SYSTEM .....	42

5. HELP AND ERROR MESSAGES .....	45
6. NAMING CONVENTIONS .....	51
6.1 FIELD NAMES AND TEXT TRIM .....	51
6.2 BLOCK NAMES .....	53
6.3 FORM NAMES .....	53
7. KEYS .....	55
7.1 GENERAL RULES .....	55
7.2 STANDARD KEY DEFINITIONS .....	55
8. VALIDATION STANDARDS FOR DATA ENTRY .....	57
8.1 SERVER-LEVEL VALIDATION .....	57
8.2 HUB-LEVEL VALIDATION .....	57
8.3 CLIENT-LEVEL VALIDATION .....	58
9. STANDARDS FOR COMMENTING ORACLE APPLICATIONS .....	59
9.1 FORM-LEVEL COMMENTS .....	59
9.2 BLOCK-LEVEL COMMENTS .....	60
9.3 TRIGGER- AND KEY-LEVEL COMMENTS .....	60
9.4 PL/SQL AND STORED PROCEDURES .....	62
9.4.1 Procedure-Level Comments .....	62
9.4.2 Function-Level Comments .....	63
10. DESIGN/DATA EXCHANGES BETWEEN ORNL AND EASTERN AREA .....	65
10.1 INITIAL ICDB DATABASE SETUP .....	65
10.1.1 System and Tablespace Structures .....	65
10.1.2 Owner of Tables .....	66
10.1.3 ICDB Table Privileges .....	66
10.1.4 Transfer Components .....	68
10.2 SUBSEQUENT CHANGES TO THE DATABASE .....	69

## LIST OF FIGURES

Figure 3.1.	The format for a menu screen	17
Figure 3.2.	The Inquiry menu screen	18
Figure 3.3.	The Multiple Shipments-Detail menu screen	19
Figure 3.4.	Format for the header block of a query screen	21
Figure 3.5.	An example of Oracle's LIST OF VALUES pop-up window on an ICDB query screen	23
Figure 3.6.	Example of a data display screen with one title line	26
Figure 3.7.	Example of a data display screen with two title lines	27
Figure 3.8.	Example of a code description pop-up window	31
Figure 3.9.	Example of an additional data display screen for container/booking/receipt data	32
Figure 3.10.	Example of an additional data display screen for GBL/stop off/remarks data	33
Figure 3.11.	Example of a summary selection screen	35
Figure 3.12.	Example of a summary selection screen	36
Figure 3.13.	Example of a summary query screen	37
Figure 3.14.	Example of a summary display screen	39
Figure 3.15.	Example of an insert/update/delete screen	40
Figure 5.1	Example of standard messages for instructing the user on appropriate keys to press to see additional shipment data	47
Figure 5.2	A standard message for a query screen	48
Figure 5.3	A standard message for a LIST OF VALUES pop-up window	49
Figure 10.1	Example tablespace creation script	66
Figure 10.2	Example user creation script	67
Figure 10.3	Example table creation script	67
Figure 10.4	Directory structure of initial setup	69
Figure 10.5	Directory structure of change version	71



## ABSTRACT

This document, prepared for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB), provides standards and conventions for the screens developed using ORACLE's SQL\*Menu, SQL\*Forms, and SQL\*Reportwriter; for the ORACLE keys; and for commenting ORACLE code. It also covers standards for database system transfers. The results of adherence to these standards and conventions by all developers at both geographically separated development sites, Oak Ridge National Laboratory (ORNL) and The Military Traffic Management Command's Eastern Area (EA), will be a consistent appearance of ICDB to users, code that is easily maintained, and a system that will be quicker to develop and integrate.

This final report of the Standards and Conventions contains general guidelines to be followed for the development of the ICDB user interface screens. Though additional ICDB user interface screens are being developed both at ORNL and EA, and existing screens may have fields added to or deleted from them, the standards and conventions presented in this document should remain unchanged.

## 1. INTRODUCTION

### 1.1 BACKGROUND

This Standards and Conventions document describes standards and conventions for the user interface to be developed for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB), a system being developed by the Military Traffic Management Command (MTMC) to provide data integration and worldwide management and tracking of surface cargo movements.

ICDB will be a data repository for the WPS terminal-level system, will be a primary source of queries and cargo traffic reports, will receive data from and provide data to other MTMC and non-MTMC systems, will provide capabilities for processing Advance Transportation and Control Movement Documents (ATCMDs), and will distribute manifests.

It is important to note that this document covers proposed standards and conventions for the ICDB user interface. Because ICDB is under development and the ICDB interface is currently in the developmental stage, the number of screens may change as the user interface is enhanced after formal implementation on the East and West Coast Hubs (see the ICDB Architectural Analysis document) and reviewed by ICDB end users. Also, actual content of these screens, especially the data fields displayed on the screens, will change as the ICDB interface design progresses. However, the purpose of this document is to describe standards for the user interface screens, for Oracle keys, for commenting code, and for database system transfers. The standards and the basic design philosophy is not expected to change unless or until MTMC moves to a newer version of Oracle tools that would produce a different design concept using a graphical user interface.

### 1.2 PHILOSOPHY

The Oracle relational database management system (RDBMS) will serve as a data repository and data communications/exchange primarily with the WPS, currently under development by MTMC. Because the development of the database schema, processing modules, and communications

components of ICDB will occur at geographically different sites [Oak Ridge National Laboratory (ORNL) and MTMC Eastern Area (EA)], it is important to develop standards and conventions to ensure consistency and maintainability in the future.

The standards and conventions for the ICDB prototype were designed to provide (1) a user-friendly interface that is simple and straight-forward, (2) an interface that uses ORACLE's predefined functions and capabilities whenever possible, and (3) an interface that uses identical keystrokes across screens/applications to provide the same functionality for a system being prototyped by multiple developers.

The following assumptions were made during the development of these standards and conventions for the ICDB system.

1. ICDB end users will have very little knowledge of the functions provided by ORACLE.
2. Users will be able to navigate through ORACLE screens with minimal help (see Sect. 7 for more information on HELP messages). Instructions for navigation will be provided automatically when they are appropriate.
3. All ICDB users will have the standard PC function keys as well as ENTER/RETURN, and Up/Down arrow keys.

The ICDB interface will be developed using Version 5.0 or later of ORACLE's SQL\*Menu and Version 3.0 or later of ORACLE's SQL\*Forms screen generator. Some screens will be designed to be displayed as pop-up windows rather than full-page screens. Specific guidelines for determining when a screen will be sized to be a pop-up window are as follows.

1. In general all look up (help) tables will be designed to be pop-up windows if the display size permits.
2. Report screens used to determine one of several choices (e.g., screens that allow users to choose columns for sorting or print destination) will also, size permitting, be pop-up windows.
3. Screens used to display additional shipment unit information, beyond the first shipment display page, will be pop-up windows. (See Sect. 3.4 for more information on additional shipment data display screens.)

The welcome screen, all menu screens, and screens that display the first-page result of a query or report will always be full-page screens.

Therefore, the term screen is used in this document to refer to either a full-page screen or a pop-up window screen. Though to a developer a SQL\*Form application may consist of more than one page, the term page is used in this document to refer to one screen the user sees. The term block is an ORACLE term. Each page the user sees may consist of more than one block of retrieved data or enterable fields, though the user is not aware that pages are divided into functional blocks. Blocks, therefore, in this document refer to logical functional groups of data on a page.

This document is divided into several sections. Section 2 discusses general standards and conventions requirements and outlines the issues addressed in the remaining sections. Section 3 discusses standards and conventions for the ICDB screens developed using SQL\*Menu , SQL\*Forms, and SQL\*Reportwriter. Section 4 discusses standards for navigation through ICDB screens. Section 5 discusses help and error messages. Section 6 discusses naming conventions. Section 7 discusses standard keys. Sections 8 and 9 discuss validation standards for data entry and standards for commenting ORACLE code. Section 10 contains standards for data exchange between ORNL and MTMC's EA during the development phase of ICDB.

This document is a living document. Though this is the final report for the ORNL and MTMC EA developmental effort for ICDB, more details may be added as implementation change versions of the user interface require changes to this document.



## **2. CONSIDERATIONS FOR ESTABLISHING STANDARDS AND CONVENTIONS**

### **2.1 PURPOSES OF STANDARDS AND CONVENTIONS**

In order to be most effective, standards and conventions to be used in the development of the ICDB system need to be established early in the system development process and modified and expanded as work progresses.

There are several reasons to establish standards and conventions in a computer system:

- to ensure continuity between the user screens,
- to facilitate expansions and modifications,
- to support a modular programming environment in which multiple programmers can work independently,
- to make the computer code easier to read and understand, and
- to maintain a professional consistency throughout the program code.

### **2.2 AREAS THAT NEED STANDARDS AND CONVENTIONS**

This document will discuss standards and conventions to be used for the ICDB menu and query screens, for the screens that initiate data distribution processes, and for screens that call reports generated by SQL\*Reportwriter. The actual format for data distribution files is dependent on various user interface agreements between MTMC and other systems as listed in Sect. 2.5. The output of reports generated by SQL\*Reportwriter is dependent on how data is sorted, broken on a page, and subtotalled. These output reports will be included in an Appendix of the final ICDB System/Subsystem Specification document currently scheduled for release in June 1994. However, many reports and some data distribution processes will be called from an Oracle SQL\*Forms screen that will follow the standards and conventions for ICDB menu and display screens discussed in Sect. 3.

Standards for the validation of enterable fields and for the commenting of SQL\*Forms code are also included in this document.

## **2.3 STANDARD TOOLS FOR IMPLEMENTING THE USER INTERFACE**

Currently, the following software products have been chosen for ICDB:

- Unix will be the operating system.
- Oracle7 will be used for the RDBMS.
- SQL\*Net 2.0 will be used for communication processes between the ICDB repository and communication hubs, and between WPS sites and the hubs.
- Third-generation procedural code will be written in ADA.
- Oracle's SQL\*Menu will be used to generate all menu screens.
- Oracle's SQL\*Forms will be used to generate data display screens, data query screens, and pop-up windows.
- Oracle's Reportwriter or Reports 2.0 will be used to generate reports.
- Oracle's CASE tool will be used for database design and data dictionary reporting.

The ICDB staff at ORNL and MTMC have decided to use Oracle's SQL\*Menu to develop all menu interface screens. The decision to use ORACLE's SQL\*Menu to develop the menu part of the interface was made for the following reasons.

1. SQL\*Menu allows for quick and easy development of screens without having to write or maintain code.
2. SQL\*Menu allows the option to enter and maintain user roles easily. Each ICDB user will be assigned to a role. The ICDB interface will use roles to control the menu items that a user is allowed to see.

Because SQL\*Forms allows the designer the flexibility to use SQL and PL/SQL statements and to pass parameters to SQL\*Reportwriter, all shipment query screens will be developed using SQL\*Forms. This will allow users to enter values for the "where" portion of a SQL statement.

## 2.4 TERMS AND ABBREVIATIONS

A-2000	AUTOSTRAD 2000
AC	Area Command
ACI	Automated Carrier Interface
AIS	Automated Information System
AMC	Army Materiel Command
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
ASPUR	Automated System for Processing Unit Requirements
ATCMD	Advanced Transportation Control and Movement Documents
AUEL	Automated Unit Equipment List
AUTODIN	Automatic Digital Network
CASE	Computer-Aided Software Engineering
CDCP	Central Data Collection Point
CFM	CONUS Freight Management
COMPASS	Computerized Movement Planning and Status System
CONEX	Container Express
CONUS	Continental United States
COOP	Continuity of Operations Plan
CPU	Central Processing Unit
DA	Data Administrator/Data Administration
DAAS	Defense Automated Address System
DASPS-E	Department of the Army Standard Port System-Enhanced
DBA	Database Administrator/Database Administration
DBMS	Database Management System
DDD	Direct Distance Dialing
DDN	Defense Data Network
DED/D	Date Element Dictionary/Directory
DIC	Document Identifier Code
DLSS	Defense Logistics Standard System
DOD	Department of Defense
DODAAC	DOD Activity Address Code
DODIC	DOD Identification Code
DOE	Department of Energy
DOS	Disk Operating System
DTS	Defense Transportation System
EA	Eastern Area
EDI	Electronic Data Interchange
ETR	Export Traffic Release
ETRR	Export Traffic Release Request
EOC	Emergency Operation Center
FD	Functional Description
FMS	Financial Management System
FOC	Final Operating Capability

FORSCOM	U.S. Forces Command
GB	Gigabyte
GBL	Government Bill of Lading
Global Variables	Variables that are passed from one application to another(e.g., g_rdd)
GMT	Greenich Mean Time
GSA	General Services Administration
GTN	Global Transportation Network
GUI	Graphical User Interface
HHG	Household Goods
HQ MTMC	MTMC Headquarters in Washington D.C.
IBS	Integrated Booking System
ICDB	Integrated Cargo Database
ID	Identifier
IEEE	Institute of Electrical and Electronic Engineers
IMDGC	International Maritime Dangerous Goods
IME	Information Management, Eastern Area
IOC	Initial Operating Capability
ITV	In-Transit Visibility
JCCO	Joint Container Control Office
JOPES	Joint Operation Planning and Execution System
LAN	Local Area Network
LDD	Logical Data Dictionary
LDM	Logical Data Model
LOGDRMS	Logistics Data Resource Management System
LOGMARS	Logistics Application of Automated Marking and Reading Symbols
MB	Megabyte
METS	Mechanized Export Traffic System
MILSTAMP	Military Standard Transportation and Movement Procedures
MILSTRIP	Military Standard Requisitioning and Issue Procedures
MSC	Military Sealift Command
MTMC	Military Traffic Management Command
OCONUS	Outside the Continental United States
ODT	Open Desktop
OO	Object-Oriented
OODB	Object-Oriented Database
ORNL	Oak Ridge National Laboratory
Parameters	Choices a user may make for limiting a query (e.g., dates, cargo type)
PC	Personal Computer
PCFN	Port Call File Number
PMO	Product Management Office
POD	Port of Debarkation
POE	Port of Embarkation
POSIX	Portable Operating System Interface for Computing Environments
POV	Privately Owned Vehicle
QA	Quality Assurance
QBF	Query-By-Forms
R&M	Reliability and Maintenance
RAM	Random Access Memory

RB <sup>F</sup>	Report-By-forms
RDBMS	Relational Database Management System
RORO	Roll-on/Roll-off
SCAC	Standard Carrier Alpha Code
SDE	Shared Data Environment
SF	Standard Form
SITREP	Situation Report
SOCO	Shipping Order/Clearance Order
SPLC	Standard Point Location Code
SQL	Structured Query Language
SS	System/Subsystem Specification
STD	Standard
STRADS	Strategic Deployment System
SUS	Software Unit Specification
TAC	Transportation Account Code
TC ACCIS	Transportation Coordinator's Automated Command and Control Information System
TCMD	Transportation Control and Movement Document
TCN	Transportation Control Number
TCON	Trailer Container Number
TCP/IP	Transmission Control Protocol/Internet Protocol
TDR	Transportation Discrepancy Report
TERMS	Terminal Management System
Text	Text, labels, and instructions that appear on a screen
TOLS	TERMS On-line System
TSM	Terminal Support Module
TTU	Transportation Terminal Unit
UCR	Unit Cargo Release
UDM	Unit Deployment Manifest
UIC	Unit Identification Code
UMD	Unit Movement Data
USAISC	U.S. Army Information Systems Command
USMTF	United States Message Text Formatting
USTRANSCOM	U.S. Transportation Command
VCC	Vessel Completed Card
WA	Western Area
WAN	Wide Area Network
WIN	WWMCCS Intercomputer Network
WIS	Worldwide Information System
WPS	Worldwide Port System
WWMCCS	Worldwide Military Command and Control System
4GL	Fourth-Generation Language

## 2.5 PROJECT REFERENCES

Departments of the Army, the Navy, the Air Force, and the Defense Logistics Agency. Defense Traffic Management Regulation. AR 55-355, NAVSUPINST 4600.70, AFR 75-2m MCO P4600.14B, DLAR 4500.3. UNCLASSIFIED. July 1986.

Directorate of International Traffic, Military Traffic Management Command. "Interface Requirements Specifications for the Automated System for Processing Unit Requirements (ASPUR), Mechanized Export Traffic System (METS II), and Terminal Management System (TERMS)." DRAFT. UNCLASSIFIED. February 1992.

Headquarters, Department of the Army. Army Life Cycle Management of Information Systems. Army Regulation 25-3. UNCLASSIFIED. November 1989.

"Interface Agreement Between the Worldwide Port System (WPS) and the LOGSA DBs." August 1993.

"Interface Agreement Between the Worldwide Port System (WPS) and the Joint Container Control MILVAN System." August 1993.

Oak Ridge National Laboratory. Architectural Analysis for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB). FINAL. February 1993.

Oak Ridge National Laboratory. Database Specifications for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB). REVISION 2. UNCLASSIFIED. November 1994.

Oak Ridge National Laboratory. Logical Data Model for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB). DRAFT. UNCLASSIFIED. August 1992.

...Oak Ridge National Laboratory. Baseline Logical Model for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB). DRAFT. UNCLASSIFIED. September 1992.

Oak Ridge National Laboratory. Functional Description for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB). UNCLASSIFIED. FINAL DRAFT. November 1992.

Oak Ridge National Laboratory. System/Subsystem Specifications for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB). UNCLASSIFIED. VERSION 2. February 1994.

USAISC. Management Plan: AUTOSTRAD Modernization Program (A-2000). UNCLASSIFIED. January 1989.

U.S. Department of Defense. Military Standard Transportation and Movement Procedures (MILSTAMP). UNCLASSIFIED. October 1988; Change 2, October 1991; Change 3, May 1992.

U.S. Department of Defense. Military Standard: DOD Automated Information Systems (AIS) Documentation Standards. UNCLASSIFIED. DOD-STD-7935A. October 1988.

United States Transportation Command, GTN Program Management Office. "Global Transportation Network Version 2.2, Worldwide Port System (WPS) Interface Requirements Specification." DRAFT. May 1993.

Worldwide Port System. Worldwide Port System Management Plan. UNCLASSIFIED. June 1992.

Worldwide Port System. WPS Continuity of Operations Plan (COOP). DRAFT. UNCLASSIFIED. March 1992.

Worldwide Port System. WPS Database Specification. UNCLASSIFIED. December 1991.

Worldwide Port System. Worldwide Port System Functional Description (FD). UNCLASSIFIED.

June 1992.

Worldwide Port System. Worldwide Port System/Subsystem Specification (SS). UNCLASSIFIED.

July 1992.

Worldwide Port System. Worldwide Port System Software Unit Specification (SUS).

UNCLASSIFIED. July 1992.

Worldwide Port System. Worldwide port System CSA Database Prototype Operations

Manual. UNCLASSIFIED. November 1993.

Worldwide Port System. Worldwide Port System Configuration Management Operational

Requirements. UNCLASSIFIED. February 1993.

### **3. SCREENPAINTING**

#### **3.1 OVERVIEW**

The following sections present an overview of the ICDB user interface screens.

##### **3.1.1 Navigation**

The ICDB interface is designed to keep to a minimum the number of screens that the user must page through. Therefore, the menu choice screens are few, and navigation among screens (see Section 4) has been carefully considered. As much as possible the screens are simple and uncluttered; however, the applications designer should choose to include more fields or menu choices on a page, as long as the screen is still easy to read, rather than designing additional pages through which the user must scroll.

Because shipment data displays contain a large number of data fields, these fields have been grouped onto several screens. The first display screen will contain those data fields that have been identified by ICDB users as the ones they most commonly need to see. This data display screen will be a full-page screen. The user will be given a message to press different keys to see additional shipment information. The user will press the F2 key (see Section 7 for more information on keys) to see container/booking/receipt data and/or the F3 key to see Government Bill of Lading (GBL)/stopoff/remarks data. Both of these screens will be pop-up windows that will overlay portions of the first full-page screen.

##### **3.1.2 Modularity**

All of the applications prototyped are as modular as possible. Rather than designing one long application, the approach used for the ICDB interface is to separate functional pieces into separate applications. Applications and processes called by the menus are being developed at different sites. For example, ATCMD processes, import manifest processes, and export manifest processes are being developed by MTMC EA. Responsibility for the development of regional processes is to be

determined. ORNL is developing the ICDB menu screens as well as the Inquiry and the User/Site/Table Management modules.

A separate menu application is being developed by ORNL for System Administration/Database Administration functions. Because this application will be used only by special users such as system administrators and database administrators, this application will not follow the standards and conventions set forth in this document.

### **3.1.3 Screen Types**

Currently there are five types of ICDB user interface screens:

1. menu screens that allow the user to make selections,
2. query screens that allow the user to enter values for building a SQL query,
3. display screens that show the result of a query or provide LIST OF VALUES information,
4. summary report screens, and
5. data entry/update/delete screens.

These screen types will be discussed in more detail in Section 3.2-3.5.

### **3.1.4 Developmental Tools**

Three Oracle tools will be used to develop the screens for the ICDB interface. Full-page menu screens will be developed using SQL\*Menu 5.0; query, display, and data entry/update screens will be developed using SQL\*Forms 3.0 or a later version; summary display screens will be developed using SQL\*Reportwriter 1.1 or Oracle Reports 2.0.

## 3.2 MENU SCREENS

### 3.2.1. Functionality

The full-page menu screens are designed to be simple and easy to follow. They allow the user to choose the menu path appropriate for his interest. No text is entered at the menu level. The user is presented with a series of sequentially numbered menu items. These items are designed so that the user may move the cursor to the desired item and press the **ENTER/RETURN** key (see Section 7 for key mapping) to see the screen represented by that item. The entire item line, which includes the number and the associated text, is highlighted when the user moves the cursor to that line. However, the user may at any time simply enter the number associated with a menu item rather than move the cursor to the item line. This number may be entered from any cursor position on the menu screen and will automatically appear in the menu choice box. Once the user presses the **ENTER/RETURN** key, the next appropriate screen (whether a menu or a form) will be displayed.

### 3.2.2 Menu Layout

The title for the full-page menu screen is positioned automatically by SQL\*Menu. The main title will display the user's path, with each path name separated by a slash (/). The secondary title line will display the name of the current menu. For example, if the user has chosen Inquiry from the Main menu and Multiple Shipments - Detail Data from the Inquiry menu, the titles on the current menu screen will be as follows:

Main Title Line:	Main/Inquiry
Secondary Title Line:	Multiple Shipment - Detail Menu

All menu titles will be written in initial capitals with the rest of the word in lowercase.

The second group of data on a menu screen consists of numbered menu choices and text trim. The menu choices and text trim are positioned automatically by SQL\*Menu. Text that explains the numbered choice is left-justified. A space separates the number from the text explanation. The text trim will be written in initial capitals with the rest of the word in lowercase.

The third group of data consists of a single line that consists of text trim and a field where the menu item number entered or selected by the user is displayed (the entry choice box). The text trim ("Enter your choice:") appears automatically (i.e., the developer does not type in the trim because it is automatically generated by SQL\*Menu). Both the text trim and the display field are also positioned automatically by SQL\*Menu.

All three groups of data (titles, numbers and associated text trim, and the entry choice box and associated trim) appear in the center of the page (again, this positioning is automatically controlled by SQL\*Menu). SQL\*Menu generates a box around the page from line 1 though line 23, columns 1-80. Figure 3.1 illustrates the format for full-page menu screens for the ICDB interface. Figures 3.2-3.3 show examples of other ICDB menu screens.

### **3.3 QUERY SCREENS**

#### **3.3.1 Overview**

SQL\*Forms query screens are designed to allow the user to enter values for constructing a specific query of ICDB shipment data (e.g., to allow the user to enter values for querying on specific shipment dates, and/or a specific POE, or a specific TCN). Existing ICDB query screens allow the user to enter values for querying a single shipment or querying multiple shipments. Other query screens will be developed for the ICDB user interface. All future query screens will follow the standards set forth in this section.

#### **3.3.2 The Header Block**

Every full-page SQL\*Forms screen, not just query screens, will begin with a "header" block. This block is separated from the remaining portion of the screen by a box that runs from line 1 through line 3, columns 1-80. The header block consists of display-only data that provides the user with a menu path and the current system date. It consists of the following fields:

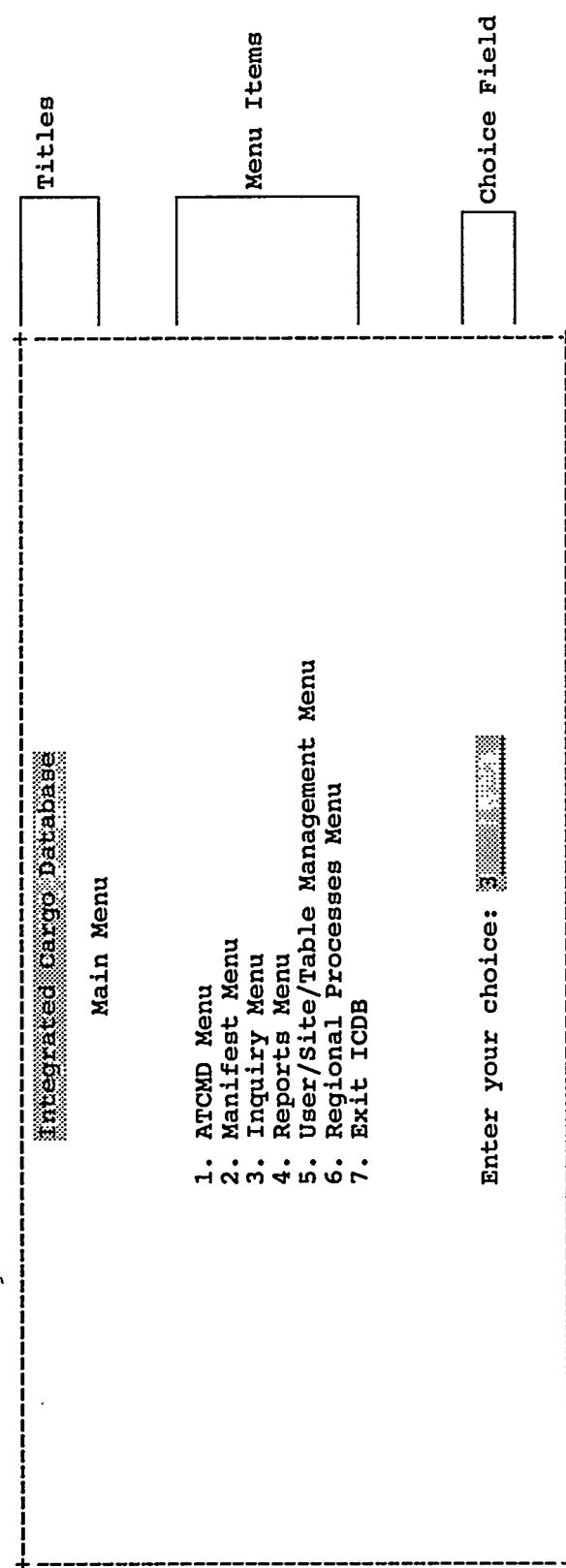


Figure 3.1. The format for a menu screen.

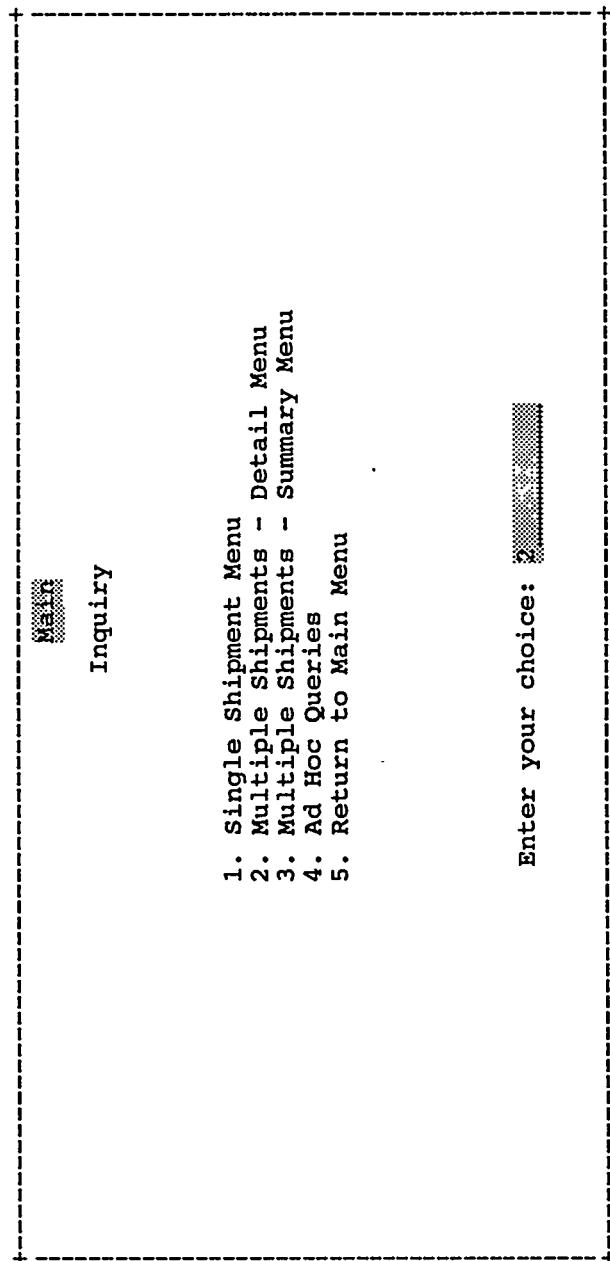


Figure 3.2. The Inquiry menu screen.

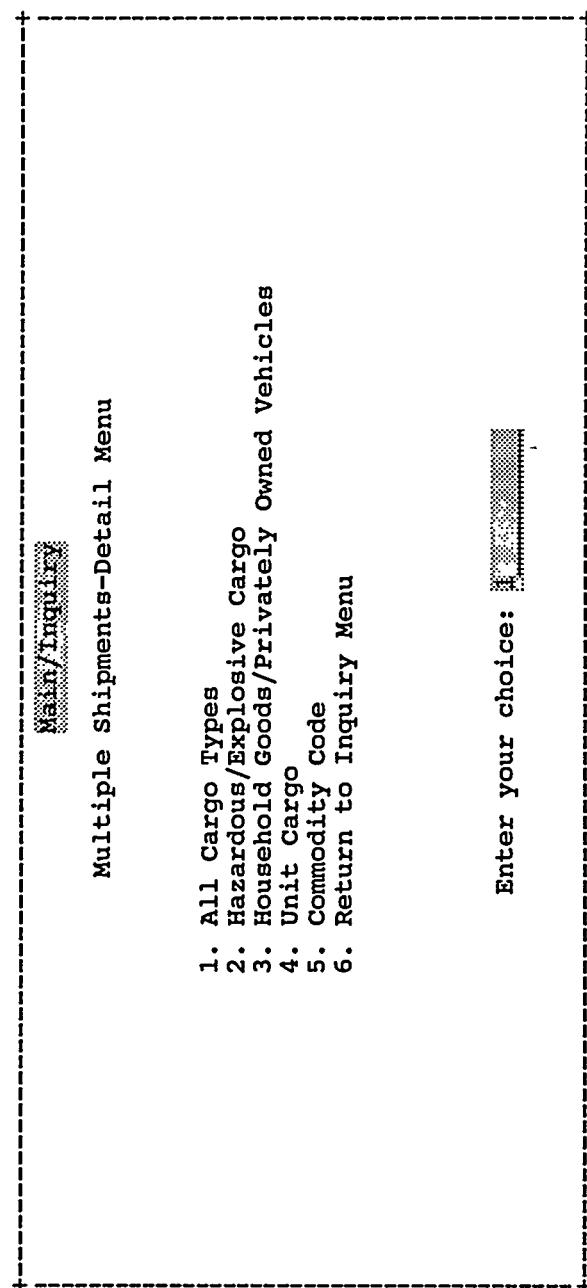


Figure 3.3. The Multiple Shipments-Detail menu screen.

1. menu\_path - the menu path the user has followed to reach this screen (each menu name is followed by a slash (/),
2. date - the current system date.

The application field attributes will be defined as follows:

1. menu\_path - character 64 (columns 2-65) and
2. date - date 9 (columns 70-78).

Figure 3.4 shows a sample ICDB query screen with the header block.

### 3.3.3 The Text Block

The text block on query screens is the block that displays the data fields and trim for the fields in which the user may enter values for building a query. It will have a box drawn around it that will encompass all of the fields on that screen. The top line of this box will be drawn from line 3 (the header block line), columns 1-80. The bottom line for the screen, columns 1-80, should be drawn one line below the last line of query fields.

The text block will include a title centered at the top of the page. The title will contain the word "Query" followed by a colon, a space, and the query type. The title will be written in initial capitals and lowercase. The following are some examples of Query screen titles:

Query: Transportation Control Number

Query: Container

Query: All Cargo Types

Following the title is the major portion of the screen, which will consist of text trim and enterable fields. This portion of the screen will be painted as follows:

1. Text trim will be written in all capital letters.

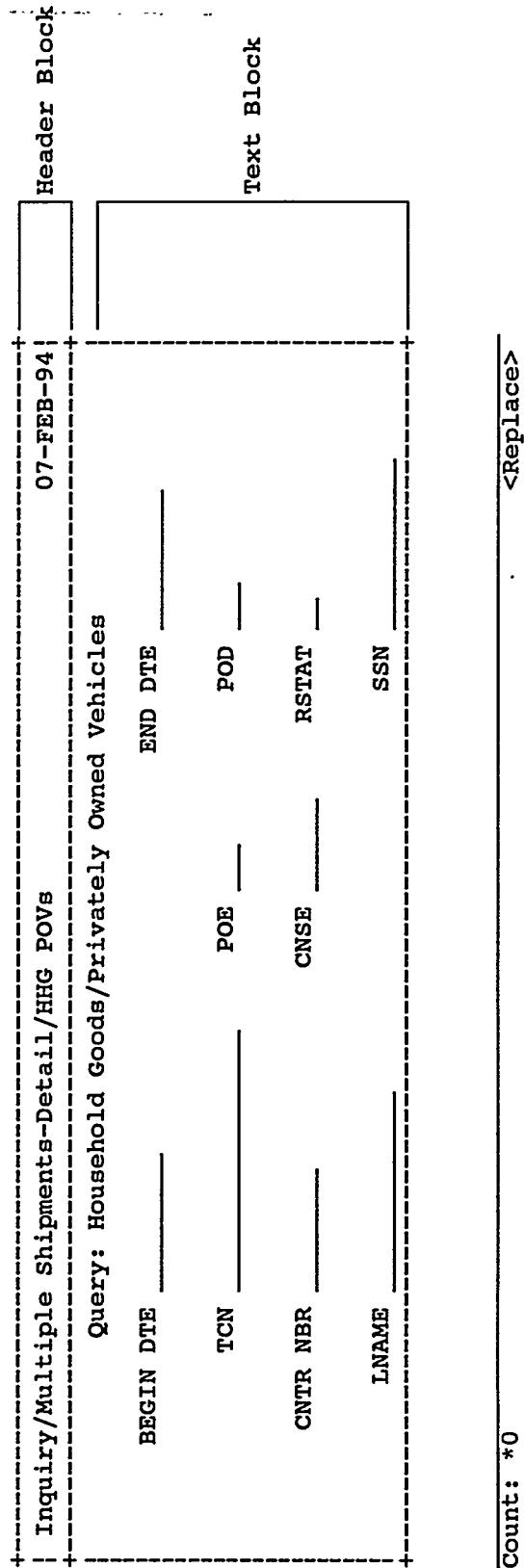


Figure 3.4. Format for the header block of a query screen.

2. Whenever possible, acronyms and abbreviations will be used in the text trim (see Section 6 for a current list).
3. No periods or colons will be used in text trim.
5. Whenever possible, data fields that occur on subsequent lines will be aligned with those on preceding lines.
6. Text trim will be right-justified.
7. A LIST OF VALUES pop-up window will be available on the query screens for any query field that is associated with a code table.
8. All query fields will be designed so that the user may enter a partial data value or the full data value (i.e., queries will be able to perform matches on the first few characters of a data value as well as on the entire data value).
9. If the user enters a value incorrectly, the data field will turn red.

The ICDB user interface will use Oracle's packaged LIST OF VALUES procedure to create a list of values for any enterable fields that are located in ICDB code tables. This packaged procedure is used because it allows for the quick creation of any LIST OF VALUES screen. Very little code has to be written or subsequently maintained by the designer/developer. This packaged procedure also allows the user to enter substring searches for code values, a MTMC requirement.

Because this window has been created by Oracle, there are fewer standards for the designer to follow than for a designer-created pop-up window. The standards are as follows:

1. There will be a title entered by the designer.
2. The designer will write the select (columns) from (code table) statement.
3. The designer will position the pop-up window close to the code field to be entered on the full-page screen.

Figure 3.5 shows an example of Oracle's LIST OF VALUES pop-up window.

The bottom line on the query screen (reserved by Oracle) will be used to display instructions and error messages to the user.

Inquiry/Multiple Shipments-Detail/FHG POVs		08-FEB-94																
Query: Household Goods/Privately Owned Vehicles																		
BEGIN DTE	_____																	
TCN	POE	POD																
CNTR NBR	Find: _____																	
LNAME	<table border="1"> <tr> <td>000</td> <td>Port 1</td> </tr> <tr> <td>1B1</td> <td>Port 2</td> </tr> <tr> <td>1B2</td> <td>Port 3</td> </tr> <tr> <td>1B3</td> <td>Port 4</td> </tr> <tr> <td>1C1</td> <td>Port 5</td> </tr> <tr> <td>1C2</td> <td>Port 6</td> </tr> <tr> <td>1D1</td> <td>Port 7</td> </tr> <tr> <td>1D2</td> <td>Port 8</td> </tr> </table>		000	Port 1	1B1	Port 2	1B2	Port 3	1B3	Port 4	1C1	Port 5	1C2	Port 6	1D1	Port 7	1D2	Port 8
000	Port 1																	
1B1	Port 2																	
1B2	Port 3																	
1B3	Port 4																	
1C1	Port 5																	
1C2	Port 6																	
1D1	Port 7																	
1D2	Port 8																	
Count: *0																		
<List><Replace>																		

Figure 3.5. An example of Oracle's LIST OF VALUES pop-up window on an ICDB query screen.

## **3.4 DATA DISPLAY SCREENS**

### **3.4.1 Overview**

Data display screens are used to display to the user data that has been retrieved from the database. Currently there are two standard types of data display screens in the ICDB user interface: full-page data display screens and pop-up window data displays that overlay part of an existing screen.

### **3.4.2 Full-Page Display Screens**

Full-page data display screens will be used to show the first-page results of a query the user has constructed. Subsequent pages, if needed, of data returned from a user query will be displayed in pop-up windows that overlay a portion of the first-page display screen.

ICDB has the following rules for ordering the data that will be displayed on the first page.

1. The data fields that appear on this page will be fields that MTMC users have identified as those most commonly needed.
2. Data fields that are common to many display screens will appear in the same location on all applicable display screens so that the user will become familiar with the screen position of common data fields.

Each data display screen will consist of a header block, a title block, and a text block as discussed in the following sections.

#### **3.4.2.1 The Header Block**

For information on the header block for full-page data display screens, see Section 3.3.2.

### **3.4.2.2 The Title Block**

The title block will contain the title of the current data display screen. The title block has the following features:

1. The title block will consist of up to two lines.
2. The first title line shows the name of the type of data display screen and may contain other information (e.g., values the user has entered for building a query). The single shipment Transportation Control Number screen, for example, displays "Transportation Control Number - XXXXXXXXXXXXXXXXXX" in the first line of the title block. The Xs stand for the value the user entered for TCN. If the user entered a partial value for this field, the title line displays the complete TCN for the current record.
3. A second title line may be used to display other information. Currently, the ICDB multiple shipment - detail screen displays use a second title line to display values, if any, that the user entered on the query screen for shipment dates.
4. The first title line appears on the line directly below the header block; the second title line appears directly below the first line (i.e, there are no blank lines in between).
5. All titles are written in initial capitals with the rest of the word in lowercase.
6. Date values, if displayed in the title, are displayed in Oracle format with the month part of the date in all uppercase (e.g., 20-JAN-92).
7. The first and second lines will be centered on the screen.
8. The title values for both lines 1 and 2 are dynamically created if they contain values entered by the user.

Figure 3.6 shows a data display screen with a single title line. Figure 3.7 shows a data display screen that has two title lines.

### **3.4.2.3 The Text Block**

The text block on data display screens is the block that displays the data fields that are returned as the result of a query. It will have a box drawn around it that will encompass all of the fields on that

Inquiry/Single Shipment/TCN		07-FEB-94	Header Block					
Transportation Control Number - HXFAZE3232AB09XXX				Title Block				
TCN	HXFAZE3232AB09XXX	POE	1M1	POD	GA3	CNSF	HXFAZE	CNSR
COMM	700Z9	PCS	376	WT	10594	CU	1136	RSTAT
TAC	A209	CNTR	NBR 4001295	PK	CT			EN
VOYDOC		VSL	NAME			VSL	STATUS	
CARRIER		CARRIER	VOYDOC			STOW	LOC	
RCPT DTE		LIFT DTE		SAIL DTE				
POD ETA		DISCH DTE		DISP DTE				
Press F2 for CNTR/Booking/RCPT data; F3 for GBL/Stopoff/Remarks data.								
<Replace>								
Count: *1								

Figure 3.6. Example of a data display screen with one title line.

Inquiry/Multiple Shipments-Detail/HHG POVs										07-FEB-94	
Household Goods/Privately Owned Vehicles - All Cargo Statuses										Dates Not Specified	
TCN	A3180558935538HXX	POE	2DC	POD	JF1	CNSE	WK4F57	CNSR			
COMM	390Z9	PCS	2	WT	2504	CU	380	RSTAT	EN		
TAC	AH35	CNTR	NBR	PK	CC						
VOYDOC	_____	VSI	NAME	_____		VSI	STATUS				
CARRIER	_____	CARRIER	VOYDOC	_____		STOW	LOC				
RCPT	DTE	LIFT	DTE	SAIL	DTE						
POD	ETA	DISCH	DTE	DISP	DTE						
NAME	SULLIVAN	BL	GRADE	E1	_____	_____	_____	STATE			
MAKE	_____	MODEL	_____	_____	_____	COLOR	_____				
DMG	CD 1	DMG	CD 2	YR	_____	DMG	CD 3	_____			
Press F2 for CNTR/Booking/RCPT data; F3 for GBL/stopoff/Remarks data.										<Replace>	
Count: *0											

Figure 3.7. Example of a data display screen with two title lines.

screen. The box will be drawn from line 4 or 5 [the title block line(s)], columns 1-80, through line 23.

This portion of the screen will consist of text trim and data fields. Data fields will be grouped logically. A blank line will separate logical groupings. The following standards apply:

1. Text trim will be written in all capital letters.
2. Whenever possible, acronyms and abbreviations will be used in the text trim (see Section 6 for a current list).
3. No periods or colons will be used in text trim.
5. Whenever possible, data fields that occur on subsequent lines will be aligned with those on preceding lines.
6. Text trim will be right-justified.
7. A code description pop-up window will be available for any data display field that is associated with a code table so that the user may see the translation for a displayed code. (See Section 3.4.2.4 for information on pop-up windows.)
8. The user will be presented with instructions on how to display a list of values, how to proceed to additional data display screens, and how to exit.

The bottom line on this screen (reserved by Oracle) will be used to display instructions and error messages to the user. When additional messages are necessary, i.e., all messages cannot be shown in the reserved line, they will be screen-painted at the bottom of the screen.

#### **3.4.2.4 The Code Description Pop-Up Window**

Each displayed field that is associated with a code table in the ICDB database will have special functionality. The user may move to this field and ask to see the description for the displayed code field by pressing F9. Oracle's packaged procedure for LIST OF VALUES does not provide the necessary functionality for a code description pop-up window; therefore, these windows will be created by the designer. The description for the code will be displayed in a pop-up window. Pop-up window screens, created using SQL\*Forms, overlay a portion of a full-page SQL\*Forms screen; therefore, they do not have a "header" block. However, other than the header block, pop-up window

screens follow the standards for the full-page menu screens as much as possible. Like the full-page screens, no text is entered at the menu level. The user is presented with a series of sequentially numbered menu items or, in this case, additional information. These items are designed so that the user may move the cursor to the desired item and press an ORACLE function key to select an item or to call the next screen or to exit the window once the displayed information is no longer needed.

Currently, the ICDB interface designer-created pop-up windows function much like a menu screen. They are used for the following purposes:

1. to display long descriptions for code fields,
2. to display additional shipment data from a user query (see Section 3.4.3),
3. to allow the user to choose columns for sorting information in a report, and
4. to allow the user to choose an output destination for a screen.

The following are standards and conventions for designer-created pop-up windows.

1. Every pop-up window will have a title that is centered inside a box.
2. The box will be drawn from line 1 through line 3.
3. The position of the window in general will be in the lower right bottom of the screen. For code description pop-up windows, the window will be positioned immediately below the associated code field unless the code field is near the bottom of the screen. In that case, the window will be positioned to the right of the code field. Each item line (if applicable) on the pop-up window is highlighted when the user moves the cursor to that line.
4. All text information will be located one space to the right of the enterable number field and will be written in initial capitals with the rest of the word in lowercase.
5. Instructions for the user appear at the bottom of the screen, written as a complete sentence followed by a period (see Section 5).

Each pop-up window will be created with the following page definition attributes enabled:

1. border and
2. remove on exit, if the window should close when the user exits or

stacked, if the window should remain open and call another pop-up window.

Pop-up windows may vary in size according to the amount of text that needs to be displayed. Figure 3.8 shows a code description designer-created pop-up menu window screen.

### **3.4.3 Additional Data Display Screens**

Frequently, data display screens returned to the user will contain more data fields than can be displayed on one page. Additional data display screens will be designed as pop-up windows. These pop-up data display screens will follow the conventions for the displayed text block discussed in Section 3.4.2.3. These pop-up windows should be positioned immediately below the first data display line on the full-page text block. The data display fields should be logically grouped. Messages that instruct the user about navigation among the screens will be screenpainted at the bottom of the window. See Figures 3.9-3.10 for examples of pop-up windows used to display additional shipment information.

## **3.5 SUMMARY REPORT SCREENS**

The summary report screens developed for ICDB will consist of three parts: selection screens, a query screen developed using SQL\*Forms, and a display screen developed using SQL\*Reportwriter or Oracle Reports 2.0.

### **3.5.1 The Summary Selection Screen**

When the user chooses to perform Multiple Shipments Summary functions from the ICDB menu, he will first see a selection screen that allows him to choose the type of summary report (e.g., a report on hazardous/explosive cargo or unit cargo) he would like to produce. This screen will look to the user like a menu screen. Currently, the decision about whether to use Oracle's Reportwriter or the latest version for reports generation, Reports 2.0, is pending. If ICDB develops these screens using Reportwriter, any selection screens that need to pass user choices through global variables will be developed using SQL\*Forms because SQL\*Menu, SQL\*Forms, and Reportwriter are not integrated.

Inquiry/Single Shipment/TCN	07-FEB-94							
Transportation Control Number - HXFAZE3232AB09XXX								
TCN	HXFAZE3232AB09XXX	POE	1M1	POD	GA3	CNSE	HXFAZE	CNSR
COMM	YOG25	PCS	376	WT	10594	CU	1136	RSTAT
Commodity Code Abbreviation and Meaning								
CODE	MEANING	ABBREVIATION	GENNOS					
	CODE MEANING	GENERAL CARGO, NOS						
Press F2 for CNTR/Booking/RCPT data; F3 for GBL/stopoff/Remarks data.								
<Replace>								
Count: *1								

Figure 3.8. Example of a code description pop-up window.

Inquiry/ Single Shipment/TCN		08-FEB-94			
Transportation Control Number - HXGMKT3232AB05XXX					
TCN HXGMKT3232AB05XXX POE 1P2 POD JH2 CNSE HXGMKT CNSR _____					
CNTR	OWNER LYKU	CNTR NBR 2095574	CNTR TCN HX7EAW6095V021TS2		
NSN	_____	TAC _____	PK ZC		
BKD	PCFN VOYDOC	BKD POE	BKD POD		
VSL	STATUS	BKD CARRIER	BKD VSL NAME _____		
RCPT	DTE	RCPT LOC	LIFT DTE		
DISCH	DTE	DISCH LOC	DISP DTE		
Press F3 for GBL/Stopoff/Remarks data; F4 for Container/Contents data.					
Count: *1		<Replace>			

Figure 3.9. Example of an additional data display screen for container/booking/receipt data.

Inquiry/Multiple Shipments-Detail/HHG POVs		07-FEB-94
Household Goods/Privately Owned Vehicles - All Cargo Statuses		
Dates not Specified		
TCN A3180558935538HXX POE 2DC POD JF1 CNSF WK4F57 CNSR _____		
GBL _____	SCAC 338	
STOP 01 _____	STOP 02 _____	STOP 04 _____
STOP 05 _____	STOP 06 _____	STOP 08 _____
REMARKS		
_____		
_____		
_____		
_____		
_____		
Press F2 for CNTR/Booking/RCPT data.		
Count: *0 <Replace>		

Figure 3.10. Example of an additional data display screen for GBL/stopoff/remarks data.

If Reports 2.0 is used for reports generation, it may be possible to develop the selection screens using SQL\*Menu.

Regardless of the development tool used to create these screens, the standards for the screens will follow those for menus discussed in Section 3.2. Figures 3.11-3.12 show examples of ICDB summary selection screens.

### **3.5.2 The Summary Query Screen**

Summary query screens are designed to allow the user to enter values for constructing a specific query of ICDB shipment data that will be passed to an Oracle report generation tool. These screens will consist of a header block and a text block. They will follow the standards and conventions described for query screens in Section 3.3. Figure 3.13 shows a sample summary query screen.

### **3.5.3 The Summary Display Screen**

The summary display screens will be generated using an Oracle Reportwriter tool, either Reportwriter or Reports 2.0. They will show the output for different selections and queries chosen and entered by the user. The report generation tool will automatically supply many of the standards for these screens by positioning data fields on the screen. User choices for sorts and breaks will determine how the data will be summarized and when subtotals are shown. These screens will not contain a header block because the report tool does not provide this capability; however, the following standards apply for these screens.

1. The display screen will contain two title lines. The first title line will show the type of summary the user has selected (e.g., "All Cargo Types," "Terminal Operations").
2. The second title line will contain the selection the user made for sorting and breaking on the displayed data (e.g., "By POE, POD, RSTAT").
3. All titles will be written in initial capitals with the rest of the word in lowercase.
4. Text trim for data fields will use the standard abbreviations and acronyms described in Section 6.1.

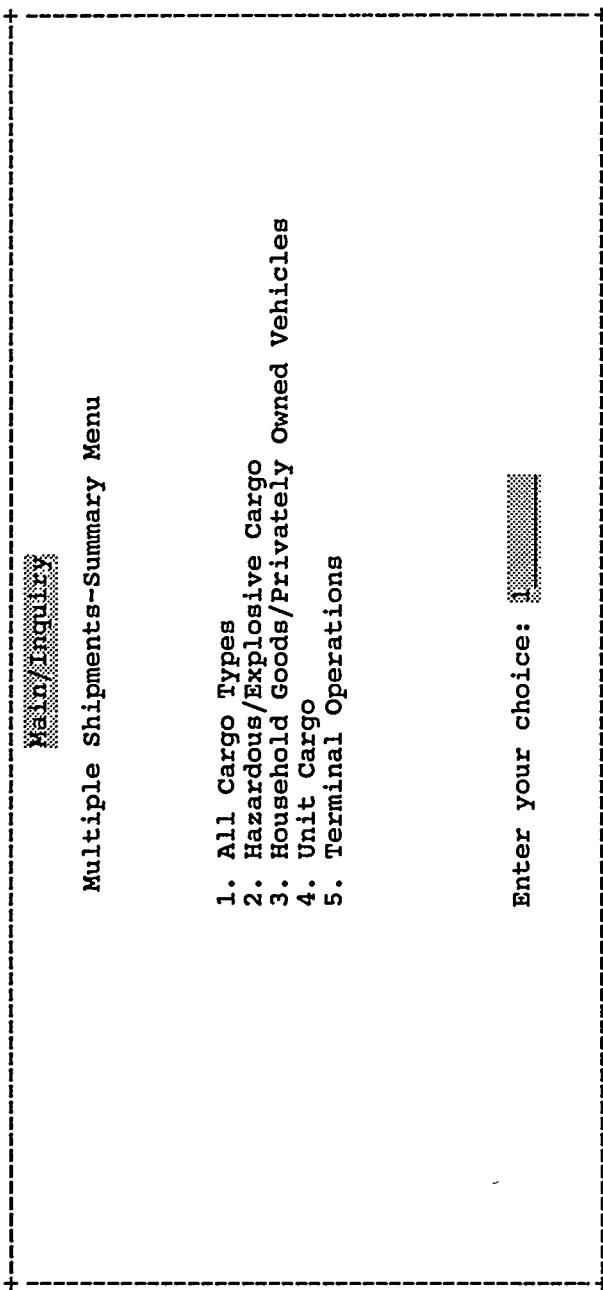


Figure 3.11. Example of a summary selection screen.

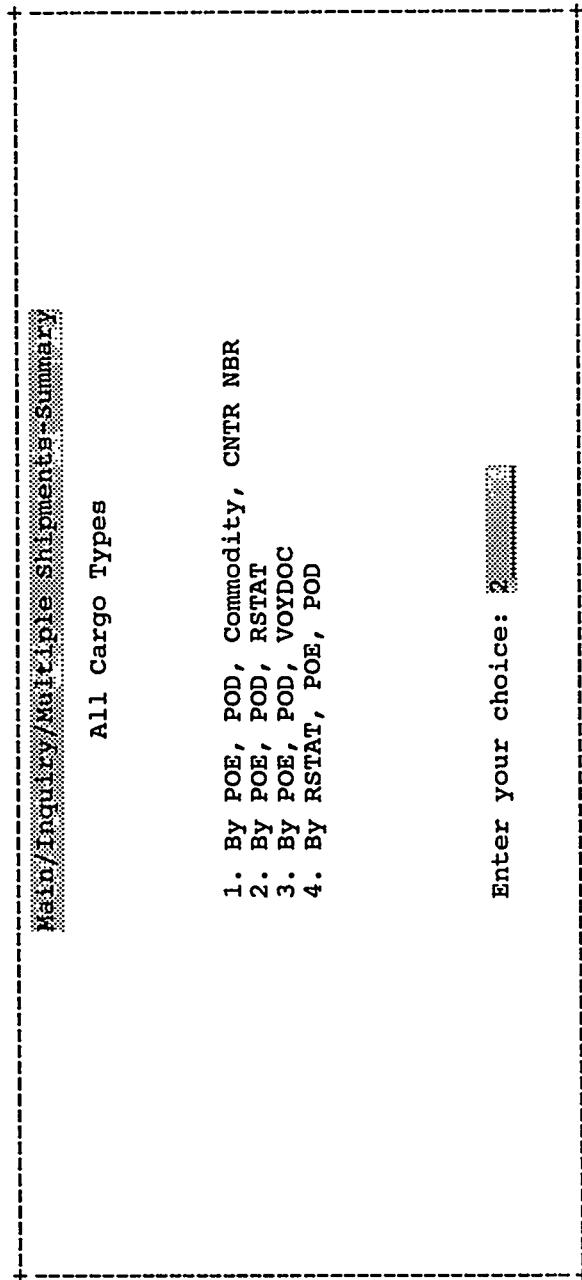


Figure 3.12. Example of a summary selection screen.

Inquiry/Multiple Shipments-Summary/All Cargo Types		08-FEB-94
Query: By POE, POD, RSTAT		
BEGIN DTE	_____	END DTE _____
POE	_____	POD _____
MANIFEST DTE	_____	VOYDOC _____
RSTAT	_____	

Enter date boundary for shipments.  
Count: \*0

<Replace>

Figure 3.13. Example of a summary query screen.

5. Subtotals for breaks will say "Total for: breakfield". For example, if a break occurs on the POE field and the first POE value is 1G1, the text would be as follows:

Total for: 1G1

6. All summary screens will contain a grand total as the last line of the report.

Figure 3.14 shows an example of a summary display screen.

### **3.6 INPUT/UPDATE/DELETE SCREENS**

For the most part ICDB user interface screens will consist of data display screens. However, there will also be screens that allow some users to update, insert, or delete data in the database. Some examples of screens with this functionality include those screens that allow shippers to do on-line builds (part of the ATCMD module), screens that allow a data administrator to maintain the ICDB code tables, and screens that allow system and database administrators to add, delete, or modify user and site accounts. Currently, the screens for on-line builds will follow the standards and conventions for WPS data entry screens. All other ICDB screens will follow the standards set forth in these sections.

These screens will contain a header block and a title block. They will follow the conventions of a full-page data display screen as described in Section 3.4.2.

The screens will also always have these additional keys explicitly enabled:

- |      |   |   |
|------|---|---|
| F10  | - | to commit inserts or updates to a record, |
| F6   | - | to insert a record,                       |
| S-F6 | - | to delete a record,                       |
| F8   | - | to execute a query (unqualified), and     |
| F7   | - | to enter a query (qualified).             |

Figure 3.15 shows an example of an insert/update/delete screen.

**Figure 3.14.** Example of a summary display screen.

user/site/Table Management/Maintain Code Tables		08-FEB-94
Activity Code		
Activity CD	<hr/>	
Activity Type	<hr/>	
CD Meaning	<hr/>	
Count: *0		
<List><Replace>		

Figure 3.15. Example of an insert/update/delete screen.

## 4. NAVIGATION

### 4.1 ENTERING THE ICDB SYSTEM

The ICDB user interface is designed to allow the user to define the output he/she would like by navigating through a minimum number of screens. Because the design is on-going, all the menu and forms screens have not yet been developed. However, all of the screens and forms will follow the same rules for navigation.

Users will enter the ICDB system by typing in the command "ICDB" or choosing "ICDB" from a menu on their PC. Once they have made this choice, they will see the first ICDB screen, which is a screen that will instruct the user about authorization for use of the ICDB system. The next screen the user sees is an ICDB welcome screen. The third screen is a menu selection screen that allows the user to choose whether he would like to query the system, perform regional processes, generate reports, or perform other processes. To navigate from the first to the third screen, the user simply needs to press ENTER/RETURN. On any of these three screens the user may exit the ICDB system by pressing the S-F10 key (see Section 7 for more information on Oracle keys).

Once the user has made a choice on the third menu screen, he may proceed to subsequent screens by pressing the ENTER/RETURN key after he has entered or selected a menu item (see Sect. 3.2). On the lowest-level screens the user may proceed by pressing the F8 key to see the results of a query. On some screens the user may also press other keys to see additional information (e.g., F9 for a list of values, F2 and/or F3 for additional shipment data), which will be displayed in a pop-up window. On each screen the user will be given an instruction that informs him of the appropriate key to press to continue or to exit.

### 4.2 ICDB USER ROLES

The menu system is designed so that not all users will see the same menu options on the third menu selection screen. For example, shippers will not be able to perform the full range of query functions

available to some users. The items a user sees on this menu screen will be controlled by the use of Oracle's SQL\*Menu role function. This function allows the DBA to assign each user to a role(s) and each high-level menu item to a role(s).

#### 4.3 EXITING FROM THE ICDB SYSTEM

The ICDB user interface is designed to allow the user to exit as quickly as possible while also allowing the user to continue to define selection criteria, data displays, reports, and data input. In general, the user will exit from a child (called) SQL\*Form and return to a parent (calling) SQL\*Form. If the calling application is a SQL\*Menu rather than a SQL\*Form, the user may exit to the next level menu or exit the ICDB user interface and return to his local PC operating system. Only those users with special privileges, such as DBAs, will be allowed to exit to the ICDB Unix workstation level. The following illustrates example navigational flows for some of the ICDB user interface screens.

<u>SCREEN TYPE</u>	<u>EXITS TO</u>	<u>DEVELOPMENT</u>
		<u>TOOL</u>
ICDB Authorization Screen	Local Operating System	SQL*Forms
Welcome Screen	Local Operating System	SQL*Forms
Main Menu	Local Operating System	SQL* Menu
Inquiry Menu	Main Menu	SQL* Menu
Single Shipment Menu	Inquiry Menu	SQL*Menu
Single Shipment Query Screen	Single Shipment Menu	SQL*Forms
Single Shipment Display Screens	Single Shipment Query Screen	SQL*Forms
Multiple Shipments-Detail-Menu	Inquiry Menu	SQL*Menu
Multiple Shipments-Detail- Query Screen	Multiple Shipments- Detail-Menu	SQL*Forms
Multiple Shipments-Detail- Display Screens	Multiple Shipments-Detail- Query Screen	SQL*Forms
Multiple Shipments-Summary-Menu	Inquiry Menu	SQL*Menu
Multiple Shipments-Summary- Query Screen	Multiple Shipments-Summary- Menu	SQL*Forms

Multiple-Shipment-Summary-  
Display Screens

Multiple Shipments-Summary-  
Query Screen

SQL\*Reportwriter  
or Reports 2.0

\*Currently, ORNL and MTMC are investigating tools to use for the development of summary screens.



## 5. HELP AND ERROR MESSAGES

Help messages may be screenpainted on the page, or they may appear in the Oracle message/error line. Some screens will use the automatic help feature or the Oracle message/error line to instruct the user about how to proceed on a screen, what key to press to continue or exit, or how to correct an error. The following standards apply for these help messages.

1. The automatic help message is displayed in the Oracle error/message line.
2. Additional help/instructional messages may be displayed when the Oracle HELP key is pressed.
3. Help messages are instructional and phrased in a positive manner.
4. The word "press" is used for instructions about the use of keys (e.g., "Press ENTER to continue.").
5. The names of Oracle keys are mapped to PC function keys and characters are written in uppercase (e.g., F8, F10, S-F10).
6. When the "S-" is used as part of a key name, the "S-" is used to indicate that the SHIFT key must be pressed in conjunction with another key (e.g., S-F10).
7. When a user query returns more than one record (whether the records are displayed one at a time or as part of a multiple record display), the user will see an "At first record." and an "At last record." message on the first and last record returned.

The following are some sample help messages.

1. Enter the TCN.
2. Enter the POD. Press F9 for valid codes.
3. Enter the POE. Press F9 for valid codes.
4. Enter the RDD. Format is DD-MMM-YY.
5. Enter the appropriate value(s); press F8 to see results.
6. Press ENTER to continue.
7. Press F10 to save data.

Section 7 provides more information on standard Oracle keys to be used in the ICDB user interface. Figure 5.1 shows the standard for screenpainted help messages that instruct the user on the appropriate keys to press to see additional shipment data. Figure 5.2 shows the standard screenpainted message for a query screen. Figure 5.3 shows the standard list of values message.

Inquiry/Single Shipment/TCN	08-FEB-94							
Transportation Control Number - HXGMKT3232AB05XXX								
TCN	HXGMKT3232AB05XXX	POE	1P2	POD	JH2	CNSE	HXGMKT	CNSR
CNTR	OWNER	LYKU	CNTR	NBR	2095574	CNTR	TCN	HX7EAW6095V021TS2
NSN			TAC			PK	ZC	
PCFN			BKD	POE		BKD	POD	
BKD	VOXDOC		BKD	CARRIER		BKD	VSL NAME	
BKD	VSL STATUS							
RCPT	DTE		RCPT	LOC		LIFT	DTE	
DISCH	DTE		DISCH	LOC		DISP	DTE	
Press F3 for GBL/stopoff/Remarks data; F4 for Container/Contents data.								
Press F9 for list of values; S-F10 to exit.								
Count: *1 <Replace>								

Figure 5.1. Example of standard messages for instructing the user on appropriate keys to press to see additional shipment data.

Inquiry/Multiple Shipments-Detail/HHG POVs		07-FEB-94	
Query: Household Goods/Privately Owned Vehicles			
BEGIN DTE	_____	END DTE	_____
TCN	_____	POE	_____
CNTR NBR	_____	CNSE	_____
LNAME	_____	RSTAT	_____
		SSN	_____

Enter partial or entire value; press F8 to see results; S-F10 to return to menu.  
 Count: \*0 <Replace>

Figure 5.2. A standard message for a query screen.

Inquiry/Multiple Shipments-Detail/HHG POVs		07-FEB-94	
Query: Household Goods/Privately Owned Vehicles			
BEGIN DTE	_____	END DTE	_____
TCN	POE 	POE	POD 
CNTR NBR	_____	Find: _____	_____
LNAME	000	Port 1	_____
	1B1	Port 2	_____
	1B2	Port 3	_____
	1B3	Port 4	_____
	1C1	Port 5	_____
	1C2	Port 6	_____
	1D1	Port 7	_____
v	1D2	Port 8	_____

Press TAB to "Find:"; DOWN ARROW to value; press ENTER.  
 Count: \*0 <List><Replace>

Figure 5.3. A standard message for a LIST OF VALUES pop-up window.



## 6. NAMING CONVENTIONS

The following subsections discuss naming conventions for the ICDB user interface screens.

### 6.1 FIELD NAMES AND TEXT TRIM

Logical names for fields are used whenever possible to aid others in deciphering code. In order to be consistent, field names in SQL\*Forms that are associated with database tables have the same name as the table columns. All fields that are passed from one form to another through the use of Oracle's global memory (global variables) are named ":global.fieldname" (e.g., :global.tcn).

The text trim associated with a field has been carefully considered. To make the user interface screens as uncluttered as possible and therefore easy to read, abbreviations and acronyms displayed on the screens will be used as follows:

- BKD for Booked,
- CD for Code,
- CMP GRP for Compatibility Group,
- CNSE for Consignee,
- CNSR for Consignor,
- CNTR for Container,
- COMM for Commodity,
- CU for Cube,
- DISCH for Discharge,
- DISP for Disposition,
- DMG for Damage,
- DTE for Date,
- ETA for Estimated Time of Arrival,
- EXP for Explosive,
- FNAME for First Name,

- GBL for Government Bill of Lading,
- GRP for Group,
- HAZ for Hazardous,
- HDLG for Handling,
- HHG for Household Goods,
- HT for Height,
- LIC for License,
- LNAME for Last Name,
- LOC for Location,
- LOT NO for Lot Number,
- NBR for Number,
- NEW for Net Explosive Weight,
- NOS for Not Otherwise Specified,
- NSN for National Stock Number,
- PCFN for Port Call File Numer,
- PCS for Pieces,
- PK for Package,
- POD for Port of Debarkation,
- POE for Port of Embarkation,
- POV for Privately Owned Vehicles,
- RCPT for Receipt,
- RDD for Required Delivery Date,
- RSTAT for Record Status,
- SSN for Social Security Number,
- TAC for Transportation Account Code,
- TCN for Transportation Control Number,
- TCON for Transportation Container Number,
- TPRI for Transportation Priority,
- UIC for Unit Identification Code,
- ULN for Unit Line Number,
- VOYDOC for Voyage Document Number,
- VSL for Vessel,

- WT for Weight, and
- YR for Year.

The screen-painted text trim that appears beside a field is as closely associated with the field name as possible.

SQL\*Form field names that are part of the header block are standardized for ICDB (see Section 3.3.1). Names for identical fields that appear in multiple blocks on the same SQL\*Form or on multiple SQL\*Forms have the same names and trim associated with them.

## **6.2 BLOCK NAMES**

The names of SQL\*Form blocks used in the ICDB system have also been standardized. All screens have a block named "header" that identifies the menu path and the current system date. SQL\*Form screens that pass in a title dynamically will have a block named "title." Subsequent blocks will have names that are meaningful [i.e., describe the function(s) that block is performing]. For example, the block named "povqry" performs a query of the database for privately owned vehicles.

## **6.3 FORM NAMES**

Each SQL\*Form is one ORACLE application. The following guidelines were used for the naming of SQL\*Forms:

1. Forms have logical, meaningful names.
2. Abbreviations and/or acronyms may be used.
3. The file form name is not more than 13 characters (including the extension); however, when needed for clarity, a lengthier version of the form name is included at the form-level comment description in SQL\*Forms.

Standards have been creating for naming the file form name for SQL\*Form Unix files in order to make it easy for a system administrator to locate a certain file and to group files by module. They follow the following standards.

1. The fist part of the file name indicates the type of query, if applicable, or the table name for an insert/update/delete (e.g., "pov" for privately owned vehicle, "cancl" (for the cancellation code table)).
2. The second part of the file name indicates, if the file is part of the query module, whether it is from the single shipment, multiple shipment, or summary modules. Single shipment queries are indicated by "ss"; multiple shipments by "ms" and summary shipments by "sum". (e.g., "povss", "povms", povsum").
3. The third and last part of a file name for files that are part of the query module indicates whether that file is a query screen ("qry") or a data display screen ("dtl").
4. Files that are forms for maintaining the code tables all end in "\_cd" (e.g., "comm\_cd" for commodity code, "cancl\_cd" for cancellation code).

The complete file name for a privately owned vehicle single shipment query screen is "povssqry"; for the multiple shipment screen, "povmsqry"; for the summary screen, "povsumqry". Similarly, the privately owned vehicle data display screen is named "povssdtl" for single shipments, "povmsdtl" for multiple shipments, and "povsumdtl" for summaries.

## 7. KEYS

### 7.1 GENERAL RULES

Because many ICDB users will not be familiar with Oracle function keys, these Oracle keys will be mapped to standard PC function keys. Instructional messages displayed to the user (see Section 5) will use PC function key names. The following standards and conventions apply to the use of keys within the prototype applications.

1. Programmers will disable all Oracle function keys at the beginning of each SQL\*Form by assigning the null function to the KEY-OTHERS key.
2. Each key that is needed for an application is enabled at the beginning of a form (e.g., to allow the exit function, the designer would define the KEY-EXIT key as exit-form).
3. The Oracle PRINT and EXIT keys will always be enabled in a SQL\*Form.

### 7.2 STANDARD KEY DEFINITIONS

The standard use of PC function keys has been carefully considered. For users to become familiar with how PC function keys work in the ICDB user interface, a PC function key should always be used to perform the same function. The following lists the PC function key and the function to be performed by that key.

- |                            |  |
|----------------------------|--|
| 1. ENTER/RETURN            | - to go to the next field,   |
| 2. Esc TAB                 | - to go to the previous field,   |
| 3. UP Arrow/<br>DOWN Arrow | - to display the next record (whether in a multiple-record display or a single record at a time display) |
| 4. F8                      | - to execute a query on a database table or view   |
| 5. F9                      | - to display a list of values for a code field or the translation for a code                             |
| 6. F10                     | - to save data to the database   |
| 7. S-F10                   | - to exit  |
| 8. S-F8                    | - to print the current screen,   |

- |     |      |   |
|-----|------|---|
| 9.  | F2   | - to call a pop-up window that displays shipment container/booking/receipt data                       |
| 10. | F3   | - to call a pop-up window that displays shipment government bill of lading (GBL)/stopoff/remarks data |
| 11. | F4   | - to call a pop-up window that displays shipment content data   |
| 12. | F5   | - to call a pop-up window that displays shipment transshipment/minibridge data                        |
| 13. | S-F1 | - to allow the user to move between columns on a selection screen                                     |
| 14. | F6   | - to insert a record  |
| 15. | S-F6 | - to delete a record  |
| 16. | F7   | - to enter a query  |

## **8. VALIDATION STANDARDS FOR DATA ENTRY**

According to the current ICDB architectural analysis and design, there will be validation control implemented at three levels: the central server, the Hubs, and the WPS client. (See the ICDB Architectural Analysis for more information on the Hub/server configuration.) Until the ICDB hardware is operational, the physical and logical location of validation routines may change; however, specific details about the location as well as in-depth descriptions of these validation software units are contained in the ICDB System/Subsystem Specifications. In general the following validation standards will apply to ICDB.

### **8.1 SERVER-LEVEL VALIDATION**

The central server will primarily function as the ICDB data repository. Most validation at the central server will be performed through the use of Oracle7 validation procedures. For example, validation processes will check to ensure that updates/inserts to records in the server cannot occur if the original record does not exist in the ICDB database (e.g., no trailer records will be allowed without a prime record). Validation routines at the server will comply with those established by WPS. Validation procedures for global data that is beyond the scope of WPS could exist on the ICDB server.

### **8.2 HUB-LEVEL VALIDATION**

The current design for Hubs is to use them for data entry (on-line builds), communications, and data reformatting processes between WPS sites and the ICDB server. The SQL\*Menu and SQL\*Forms queries will reside on the Hubs. To ensure that the user does not query the server on a nonexistent value and waste valuable networking and processing time, all SQL\*Forms applications will have validation embedded on any fields with a list of valid values (i.e., fields that reside in a code lookup table) that the user might enter for querying purposes. Copies of the ICDB server code tables will exist on the Hubs.

The ICDB user will sometimes be expected to enter a code or an abbreviation in a field. For example, rather than entering a full cargo record status the user will enter a shorter cargo record status code. Both validation and help are offered for any codes that are entered by the user. If the user enters a cargo record status code that is not valid, the system will display a message in the Oracle help/message line telling the user that the code is invalid and to press the F9 key for a list of valid values.

This look-up of valid values is designed one of two ways; however, the display will appear the same to the user. For code/abbreviation values with a short list (no more than ten valid values), the F9 key is programmed to retrieve all values into the code/abbreviation field and the translation field, and the user may scroll through them to select the appropriate value. For those codes/abbreviations with more than ten valid values, the user will see the data one page at a time. When he/she presses the F9 key the first page of valid values will be displayed. The user will scroll through these values to select the appropriate one. If the requested display returns a long list, the user will be given a choice on the screen to enter the first two letters of a value, and the list of values function will return only those codes/translations that begin with those letters.

Because a centrally maintained copy of the code table will reside at each Hub, no central server access is needed to perform these SQL\*Forms validation and display routines. In order for the LIST OF VALID VALUES procedure to be used to validate data entry fields, it is critical that ICDB and WPS maintain the same codes and code values.

Data validation for on-line builds and ATCMD batch processing will conform to validation performed by WPS. Details of these validations are provided in the [Functional Description](#).

### **8.3 CLIENT-LEVEL VALIDATION**

Client-level validation will be done at the WPS sites and is out of the scope of this document.

## 9. STANDARDS FOR COMMENTING ORACLE APPLICATIONS

### 9.1 FORM-LEVEL COMMENTS

All applications are commented at the form level. The comments are informative but basic at this level. More specific information is provided in comments at other levels. The following format will be used for commenting all applications at the form level:

1. Application Name:
2. Date Created:
3. Programmer Name/Site:
4. Purpose:
5. Functions:
6. Changed By Programmer's Name/Site:
7. Change Date:
8. Change Purpose:

Each form will have the first five items completed at the time that the form is created. Information about changes to a form will be added (items 6-8, etc.) when a form is enhanced or a bug fix is implemented. Item 4. Purpose: should contain a brief description of the overall functionality of the form. Item 5. Functions: should contain a numbered list of the functions performed in the form. The following is an example of the standard followed for a form-level comment.

1. *Application Name:* *cont\_single\_ship*
2. *Date Created* *01-JUN-93*
3. *Programmer Name:* *Jackie Fluker - ORNL*
4. *Purpose* *This form will display container data along with corresponding content data. Data is retrieved based on the user-specified van\_owner and van\_nr or the user-specified van\_owner and tcon.*
5. *Functions:*
  1. *For each displayed container record, the user can press EXECUTE QUERY to review any content data for the associated container record.*
  2. *The user can press LIST OF VALUES to see additional information for any field on the form associated with*

- a code table (e.g., poe, pod, commodity code).*
3. *It is also possible to press the NEXT BLOCK key from either the container or the content block to move back and forth between the displayed data.*
  6. **Changed By**  
*Name/Site:* **Jackie Fluker - ORNL**
  7. **Change Date:** **01-NOV-93**
  8. **Change Purpose:** *This form was edited to allow the user to have NEXT BLOCK functionality.*

## 9.2 BLOCK-LEVEL COMMENTS

Each block in an application is commented. Comments include the name of the block, the name of the **base table** (or that the block is a control block if it has no base table) and a list of the functions of that block.

The following is an example of a comment for a block.

*Block ten is a hidden block that is not displayed to the user so that the commit to the database seems as if the user is committing from the block he sees on the screen. It is from this block that a COMMIT is fired.*

**Base Table:** *Conditions*

- Functions:**
1. *Values entered from the previous block (block nine) are copied into base table fields.*
  2. *A row that contains these values is inserted into the conditions table in the database.*
  3. *Once the commit is performed, the user exits from the application to the main parameter screen where he may choose to enter other conditional parameters or to exit.*

## 9.3 TRIGGER- AND KEY-LEVEL COMMENTS

Each trigger used in an application is commented. Triggers may appear at the form level, the block level, or the field level; however, the standards for commenting are the same wherever the trigger is defined. In order to easily locate triggers and debug applications, triggers and keys are defined at the form level or the highest level that is possible. Of course, there are times when a trigger or key

applies to only one block or only one field. For example, in the block example (Sect. 9.2) the COMMIT key was defined at the block level because the user does not have data to commit before this point, nor is he/she ready to move back to a previous application until an EXIT or a COMMIT has been performed.

The following is an example of a KEY\_STARTUP trigger and comments for that trigger:

*Trigger Name: KEY-STARTUP*

*Comment: This trigger calls two procedures (generate\_path and execute\_query) that display information to the users (e.g., titles, dates) and initializes two form variables. Also, a query is performed on the first block (ss\_van) of the form (cont\_single\_ship) based upon the user-specified values for the fields van\_owner, van\_nr, or tcon.*

**Text:** generate\_path;  
initialize\_variables;  
:van\_owner\_title := 'Container Owner - ';  
:cntr\_nbr\_title := 'Container Number - ';  
execute\_query;

An example of comments for the KEY-OTHERS trigger is as follows:

*Comment: This trigger disables all keys. For each key that is needed for an application, the designer must explicitly enable it at the beginning of the form.*

**Text:** NULL;  
MESSAGE('This function is not available.');

Some triggers (earlier versions of Forms) consist of a series of steps. Each step is not commented unless the trigger is a complicated one. Usually a comment at the trigger level that describes why this trigger is being performed and what functions are being performed in the series of steps is sufficient. An example of a trigger with steps follows:

```
*copy_vals
;SQL>
#copy eight.user_name8 ten.user_id
/
;Message if value not found :
;Must value exist Y/N :
Y
```

```

#copy eight.app_name ten.app_name
/
;Message if value not found :
;Must value exist Y/N :
Y
#select userenv('TERMINAL') into ten.terminal_id from sys.dual

;Message if value not found :
;Must value exist Y/N
Y

```

The comments for this trigger follow.

*The trigger copy\_vals copies values and displays them on the first line of the page as a "banner" for the user. The trigger copy\_vals copies the values in the fields user\_name8, and app\_name in block eight to the fields user\_id and app\_name in block ten. It also selects the user terminal definition into the terminal\_id field in block ten.*

## 9.4 PL/SQL AND STORED PROCEDURES

PL/SQL and stored procedures, which may be called from more than one application or more than one Unix shell script, will be commented at the procedure level and at the block/function level.

### 9.4.1 Procedure-Level Comments

Each PL/SQL procedure will be commented at the procedure level to give the name of the procedure and to explain the general purpose of the procedure. It will also present a number list, if this is a stored procedure, that names all applications or scripts that call this procedure. The following is an example of the standard for commenting a PL/SQL procedure written inside SQL\*Forms.

<i>Procedure Name:</i>	<i>query_container_data</i>
<i>Purpose:</i>	<i>This procedure selects the appropriate container su_id for for the purpose of building a "where" clause. It retrieves corresponding content records for a specified container owner and number.</i>
<i>Called By:</i>	<i>PRE-QUERY trigger on the ssn_van block in the cont_single_ship form.</i>

A procedure that exists outside of SQL\*Forms should follow these same standards for comments.

#### 9.4.2 Function-Level Comments

Each PL/SQL "block" will also be commented for every procedure. A block might be defined as a logical unit of code that is part of a procedure. For example, part of a procedure might include a procedure to build the where clause of a SQL statement based on fields for which the user has entered values. This functional block would include statements that look for values the user has entered. The following is an example of commenting PL/SQL functions inside a stored procedure called by SQL\*Forms:

```
procedure query_container_data
is
  *** Cursor declaration                                ***
  cursor su_id is
    select c.su_id
    from single_ship c, singel_ship ss
    where c.su_id = ss.van_su_id
    and c.van_owner like :header.hvan_owner
    and rtrim(c.van_nr) like :header.hvan_nr;
  *** Variables for use in procedure                  ***
  query_string := '#in(';
  cntr := 1;
  open suidlist;
  *** This block will process the retrieved          ***
  *** su_ids each time one is fetched from          ***
  *** the database. The value will be                ***
  *** assigned to a variable the first time          ***
  *** through. For each additional retrieval,        ***
  *** the value will be concatenated to the          ***
  *** existing variable to form a "where"            ***
  *** clause.                                      ***
loop
  exit when suidlist%notfound;
  fetch suidlist into suid;
  if cntr = 1 then
    query_string := query_string || suid;
  else
    query_string := query_string || ',' || suid;
```

end if;

\*\*\* This block processes each row one at a time. It will attempt to \*\*\*  
\*\*\* process an update for each record. If the record does not exist, it \*\*\*  
\*\*\* will create a new record.

```
FOR table2 in action_cur LOOP
    IF table2.field2 = 'X' THEN
        UPDATE table3 SET field = table2.field
        WHERE field = table2.field;
    IF SQL%NOTFOUND THEN
        INSERT INTO table new
        VALUES (table2.field1, table2.field3);
    END IF
    .
    .
    .
```

## 10. DESIGN/DATA EXCHANGES BETWEEN ORNL AND EASTERN AREA

During the development phase of ICDB, it is important to consider how design changes will be exchanged between ORNL and Eastern Area (EA). It is our understanding that ORNL will be responsible for initiating any changes to the shared database structure during the initial development. ORNL change versions should be sent to the EA Command approximately every two weeks, though there may be occasional situations that call for a more frequent exchange. These change versions will be sent on tape or electronically, depending on system access at participating sites.

### 10.1 INITIAL ICDB DATABASE SETUP

#### 10.1.1 System and Tablespace Structures

The exchange of information from ORNL to EA will often be made by sending UNIX shell scripts that can be executed from within SQL\*Plus. Although it is not absolutely necessary that the Oracle home directory structure and location of data files be the same on both systems, it will make it much easier to write scripts to exchange information if they are the same. Several key steps should be followed in preparation for information exchange.

1. The Unix directory structures on each machine should be the same.
2. The Oracle Home Directory structures should be the same.
3. The database datafiles should be stored in the same directory on each machine.
4. The initsid.ora files and other configuration files should be kept similar.
5. Each site should have a core set of tablespaces that are the same.

If these steps are observed, then scripts written at ORNL to create tables such as the explosive table will run at EA without any modifications. Therefore ORNL will send EA all tablespace creation scripts which are needed to hold shared tables. Fig. 10.1 shows a sample tablespace creation script. Furthermore, during the initial database setup (and in the future if ORNL changes parameters), ORNL will send EA the Oracle initialization file, initsid.ora, that contains the initialization parameters

used for the ORNL ICDB. It will not be mandatory that ORNL and EA keep the same initialization parameters, but communication between ORNL and EA about these parameter settings will become important as the system is tuned for performance.

```
create tablespace icdb
  datafile '/db/ORACLE/icdbA/icdb01.dbf' size 99900K
  default storage (
    initial 10K
    next 50K
    maxextents 121)
/
```

**Figure 10.1. Example tablespace creation script.**

### **10.1.2 Owner of Tables**

ORNL will create users called icdb, hub, and icdbhub that will be the owners of all of the physical tables created for the ICDB database. Because many of the scripts exchanged between ORNL and EA will need to be run by the owner of the ICDB tables, ORNL will provide EA with a script that will create the user icdb with a default tablespace of icdb. See Fig. 10.2 for a sample user creation script for the ICDB user. Scripts to create other users may be provided later to facilitate the transmission of CASE applications or Menu applications owned by these users.

### **10.1.3 ICDB Table Privileges**

During the development stage of ICDB, initial table creation scripts will create public synonyms for the ICDB tables to enable all Oracle users to reference tables without using the username prefix (i.e., the outsize table can be referenced as outsize rather than icdb.outsize). Also, the table creation scripts will grant read, write, update, and delete privileges to these tables for all ICDB developers.

```
CREATE USER icdb
IDENTIFIED BY xxxx
DEFAULT TABLESPACE icdb
TEMPORARY TABLESPACE temp
QUOTA UNLIMITED ON icdb

GRANT CONNECT,RESOURCE,DBA to icdb
```

Figure 10.2. Example user creation script.

Figure 10.3 shows an example table creation script.

```
drop table icdb.explosive
/
create table icdb.explosive
(su_id          char(15)      ,
explosive_seq_nr  number(2)      ,
CONSTRAINT pk_explosive
PRIMARY KEY(su_id,explosive_seq_nr)
USING INDEX
TABLESPACE indexes
explosive_cube    number(5)      ,
explosive_pcs     number(5)      ,
explosive_wt      number(6)      ,
net_explosive_wt  number(6)      ,
lot_nr           varchar2(14)    )
tablespace icdb
/
grant all on icdb.explosive to public
/
CREATE PUBLIC SYNONYM explosive FOR icdb.explosive
```

Figure 10.3. Example table creation script.

#### 10.1.4 Transfer Components

For the initial ICDB database setup, ORNL will provide EA with the following.

1. A 'README' file which will explain what information is contained in the transmission as well as where information is contained (i.e., identify what file types are contained in which directory). The 'README' file will also identify which scripts should be run in SQL\*Plus, and the username for login to SQL\*Plus.
2. A 'master\_script.sql' file which will be executable from SQL\*Plus to perform the needed initialization steps.
3. A directory called dbcreate. The dbcreate directory will contain tablespace creation scripts for the tablespace icdb (where icdb tables will reside) as well as scripts to expand existing tablespaces and to add new users.
4. A directory called tables. The tables directory will contain creation scripts for all ICDB tables and an export file for importing any existing data, if available.
5. A directory called case. The case directory will contain an export file for importing each CASE application.
6. A directory called other. The other directory will contain SQL\*Net programs developed to populate ICDB tables.

Figure 10.4 illustrates the components of the initial database exchange.

## 10.2 SUBSEQUENT CHANGES TO THE DATABASE

After transfer to EA of the initial files for database setup, ORNL will send change versions to EA approximately every two weeks. These change versions will include additions to the database (i.e., new tables that have been added and new screens that have been created since the last version) and modifications (e.g., elements that have been added or deleted or screens whose appearance or functionality have changed). These change versions will be placed in the directory \$ORACLE\_BASE/local/to\_ea. Change versions which EA sends to ORNL should be placed in the

## INITIAL DATABASE SETUP

### initsetDDMMYY

README	file
master_script.sql	file
dbcreate	directory
tables	directory
case	directory
other	directory

### dbcreate

add_users
create_tablespace
expand_tablespace

### tables

create_table
--------------

### case

case_application_export
-------------------------

### other

program1
----------

Figure 10.4. Directory structure of initial setup.

directory \$ORACLE\_BASE/local/to\_ornl. It is assumed that when ORNL sends a new set of tables owned by the user icdb, a new set of menus, or a new CASE system, that these will replace the old ones. In addition, when EA sends change versions to ORNL, these will replace previous database objects which EA develops.

ORNL will provide the following to EA for change versions:

1. A 'README' file that provides instructions for installing the change version.
2. A 'master\_script.sql' file that can be executed from SQL\*Plus.
3. A directory called tables that will contain export files for the user icdb and any other user who may own tables pertinent to ICDB.
4. A directory called case that will contain an export of the most recent CASE application.
5. A directory called screens that will contain two other directories forms and menus. The forms directory will contain screen forms that are used in the icdb system. The menus directory will contain scripts to create menu applications.
6. A directory called other that will contain SQL\*Net and other miscellaneous programs.

Figure 10.5 illustrates the components of subsequent change versions.

---

## CHANGE VERSIONS

DDMMYY.1

README	file
master_script.sql	file
tables	directory
case	directory
screens	directory
other	directory

tables

user\_icdb\_export

case

case\_application\_export

screens

forms	directory
menus	directory

forms

screenform1

menus

menu\_application\_unload

other

program1

---

Figure 10.5. Directory structure of change version.