

ornl

RECEIVED

JAN 25 1996

ORNL/TM-13119

OSTI

**OAK RIDGE
NATIONAL
LABORATORY**

MARTIN MARIETTA

**System/Subsystem
Specifications for the
Worldwide Port System (WPS)
Regional Integrated Cargo
Database (ICDB)**

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

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED
✓M

MASTER

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

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.

**SYSTEM/SUBSYSTEM SPECIFICATIONS FOR THE
WORLDWIDE PORT SYSTEM (WPS) REGIONAL
INTEGRATED CARGO DATABASE (ICDB)**

J. P. Rollow

**E. Z. Faby†
R. A. Ferguson‡
J. Fluker†
J. Grubb†**

**B. R. Hancock†
P. C. Shipe
L. F. Truett**

†University of Tennessee, Knoxville, Tennessee
‡Science Applications International Corporation, Oak Ridge, Tennessee

Prepared for the
**MILITARY TRAFFIC MANAGEMENT COMMAND
PRODUCT MANAGEMENT OFFICE
OFFICE OF THE DEPUTY CHIEF OF STAFF FOR INFORMATION
MANAGEMENT**
Falls Church, Virginia 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

.

.

TABLE OF CONTENTS

SECTION 1. GENERAL	1
1.1 PURPOSE OF THE SYSTEM/SUBSYSTEM SPECIFICATION	1
1.2 PROJECT REFERENCES	2
1.2.1 Organizational References	2
1.2.2 Documents	2
1.3 ACRONYMS AND ABBREVIATIONS	4
SECTION 2. SUMMARY OF REQUIREMENTS	7
2.1 SYSTEM/SUBSYSTEM DESCRIPTION	7
2.2 SYSTEM/SUBSYSTEM FUNCTIONS	9
2.2.1 Accuracy and Validity	11
2.2.2 Timing	12
2.3 FLEXIBILITY	12
SECTION 3. ENVIRONMENT	15
3.1 AIS EQUIPMENT ENVIRONMENT	15
3.2 COMMUNICATIONS ENVIRONMENT	15
3.2.1 Network Description	16
3.2.2 Physical Interface	18
3.2.3 Protocol Interface	18
3.2.4 Applications User Interface	20
3.2.5 Diagnostics	20
3.3 SUPPORT SOFTWARE ENVIRONMENT	20
3.4 SOFTWARE INTERFACES	21
3.5 SECURITY	21
3.5.1 Physical Access	21
3.5.2 Network and Dial-up Access	21
3.5.3 Unix User Access	23
SECTION 4. DESIGN DETAILS	25
4.1 GENERAL OPERATING PROCEDURES	25
4.2 SYSTEM LOGICAL FLOW	25
4.3 SYSTEM DATA	26
4.3.1 Inputs	26
4.3.2 Outputs	26
4.3.2.1 Data Transmission Files	26
4.3.2.2 Queries	27
4.3.2.3 Reports	27
4.3.3 Database/Data Bank	27

TABLE OF CONTENTS (Cont.)

4.4 SOFTWARE UNIT DESCRIPTION	28
4.4.1 User Interface Processes	28
4.4.1.1 The ATCMD Subsystem	31
4.4.1.2 The Manifest Subsystem	34
4.4.1.3 The Inquiry Subsystem	38
4.4.1.4 The Reports Subsystem	49
4.4.1.5 The Utilities Subsystem	89
4.4.2 Automated Processes	89
4.4.2.1 Data Transfer Subsystem	89
4.4.2.2 System Integrity Checks Subsystem	99
4.4.2.3 Data Conversion Subsystem	99
4.4.2.4 Data Distribution Subsystem	103
4.4.3 System Administration/Database Administration Processes	119
4.4.3.1 Audit Reports Subsystem	120
4.4.3.2 Database Administration Reports Subsystem	124
4.4.3.3 Maintenance Subsystem	133
4.4.3.4 Unix Reports Subsystem	136
4.4.3.5 Registration Subsystem	139
4.4.3.6 Menu Interface Subsystem	144

LIST OF FIGURES

Figure 2.1.	Relationships of the user organizations to ICDB.	8
Figure 2.2.	The ICDB hardware configuration	10
Figure 3.1.	Distributed ICDB Server configuration	17
Figure 3.2.	Distributed ICDB Area Hub configuration	19
Figure 3.3.	The systems and organizations with which ICDB interfaces.	22
Figure 4.1.	The ICDB process categories.	29
Figure 4.2.	User Interface processes subsystems.	30
Figure 4.3.	The ATCMD subsystem	32
Figure 4.4.	The Manifest subsystem	35
Figure 4.5.	The Inquiry subsystem.	39
Figure 4.6.	The Single Shipments subfunctions.	40
Figure 4.7.	The Multiple Shipments - Detail subfunctions.	43
Figure 4.8.	The Multiple Shipments Summary.	47
Figure 4.9.	The Ad Hoc Queries subsystem.	50
Figure 4.10.	The Reports subsystem.	51
Figure 4.11.	The Reports - Cargo Management and Documentation subfunction	52
Figure 4.12.	The Reports - Cargo Management and Documentation - Booked vs Lifted subfunction	53
Figure 4.13.	The Reports - Terminal Operations subfunction	55
Figure 4.14.	The Reports - Terminal Operations - Export Cargo On-hand subfunction	56
Figure 4.15.	The Reports - Terminal Operations - Export HHG/POV On-hand Shippable subfunction	57
Figure 4.16.	The Reports - Terminal Operations - Export Hazardous/Explosive On-hand Shippable subfunction	58
Figure 4.17.	The Reports - Terminal Operations - Import Cargo On-hand subfunction	59
Figure 4.18.	The Reports - Terminal Operations - Import HHG Cargo On-hand subfunction	60
Figure 4.19.	The Reports - Terminal Operations - Import POV Cargo On-hand subfunction	61
Figure 4.20.	The Reports - Terminal Operations - Import Hazardous/Explosive subfunction	62
Figure 4.21.	The Command-Level subfunction	66
Figure 4.22.	The Reports - Manifest-Specific Reports subfunction	69
Figure 4.23.	The Regional Processes subsystem.	71
Figure 4.24.	The Print Reject Listing subfunction	74
Figure 4.25.	The Maintain Central Database Code Tables subfunction	76
Figure 4.26.	The Maintain WPS Code Tables distributed to WPS.	79
Figure 4.27.	The Monitor Automated Data Extracts subfunction.	85
Figure 4.28.	The Print Open Discharge Dates subfunction.	88
Figure 4.29.	The Utilities subsystem	90
Figure 4.30.	The Automated Processes subsystem.	91
Figure 4.31.	The Data Transfer subsystem	92
Figure 4.32.	The Data Conversion subsystem	100
Figure 4.33.	ICDB scripts for data recipients	104
Figure 4.34.	The Data Distribution subsystem.	105
Figure 4.35.	The Data File Generation subfunction.	106
Figure 4.36.	The Data File Transmission subfunction.	114
Figure 4.37.	The System Administration/Database Administration subsystem.	121

LIST OF FIGURES (Cont.)

Figure 4.38.	The Audit Reports subsystem.	122
Figure 4.39.	The Database Administration Reports subsystem	125
Figure 4.40.	The Database Structure subfunctions.	126
Figure 4.41.	The Database Access subfunctions.	127
Figure 4.42.	The Database Objects subfunctions.	128
Figure 4.43.	The Database Privilege subfunction	129
Figure 4.44.	The Maintenance subsystem	134
Figure 4.45.	The Unix Reports subsystem	137
Figure 4.46.	The Registration subsystem	140
Figure 4.47.	The User Registration subsystem	142
Figure 4.48.	The Export Manifest Communications subfunction	145
Figure 4.49.	The Code Table Distribution subsystem	146

LIST OF TABLES

Table 4-1.	Data sets to be received by ICDB	26
Table 4-2.	Data sets to be sent by ICDB	27

ABSTRACT

This System/Subsystem Specifications for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB) documents the system/subsystem functions, provides details of the system/subsystem analysis in order to provide a communication link between developers and operational personnel, and identifies interfaces with other systems and subsystems.

It must be noted that this report is being produced near the end of the initial development phase of ICDB, while formal software testing is being done. Following the initial implementation of the ICDB system, maintenance contractors will be in charge of making changes and enhancing software modules. Formal testing and user reviews may indicate the need for additional software units or changes to existing ones. This report describes the software units that are components of this ICDB system as of August 1995.

SECTION 1. GENERAL

1.1 PURPOSE OF THE SYSTEM/SUBSYSTEM SPECIFICATION

This System/Subsystem Specification describes the system objectives 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. This specifications document is written to fulfill the following objectives:

- to provide a detailed definition of the system/subsystem functions,
- to communicate details of the ongoing analysis between the user's operational personnel and the appropriate development personnel, and
- to define in detail the interfaces with other systems and subsystems and the facilities to be utilized for accomplishing the interfaces.

The ICDB will be a data repository for the WPS terminal-level system, will be a primary source for 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 Control and Movement Documents (ATCMDs), and will process and distribute manifests.

It is important to note that this system/subsystem specification document covers the initial operating capability for the ICDB operational system. ICDB is currently in the final stages of development with a projected implementation date of August 1995. This document describes initial operating capability for ICDB as of August 1995. Further enhanced capability will be added as the system evolves.

1.2 PROJECT REFERENCES

1.2.1 Organizational References

Sponsor: The Military Traffic Management Command, Worldwide Port System Project Management Office, 5611 Columbia Pike, Falls Church, VA 22041.

Users:

Development Coordinator: Military Traffic Management Command, U.S. Army Information Systems Command, ICDB Product Management Office, 5611 Columbia Pike, Falls Church, VA 22041.

Operating Center: Military Traffic Management Command, U.S. Army Information Systems Command, Eastern Area, Military Ocean Terminal, Building 42, Bayonne, NJ 07002.

1.2.2 Documents

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 Specification for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB).** REVISION 2. UNCLASSIFIED. November 1993.

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. Maintenance Manual for the Worldwide Port system (WPS) Regional Integrated Cargo Database (ICDB). UNCLASSIFIED. DRAFT, March 1994.

Oak Ridge National Laboratory. System Operation Manual for the Worldwide Port System (WPS) Regional Integrated Cargo Database (ICDB). UNCLASSIFIED. DRAFT, May, 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.

1.3 ACRONYMS AND ABBREVIATIONS

A-2000	AUTOSTRAD-2000
AAFES	Army/Air Force Exchange System
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
CISIL	Central Integrated System for International Logistics
CMS	Crisis Management System
COMPASS	Computerized Movement PLANning and Status System
CONEX	Container Express
CONUS	Continental United States
CPU	Central Processing Unit
DA	Data Administrator/Data Administration
DAAS	Defense Automated Address System
DAMMS	Department of the Army Movement Management System
DBA	Database Administrator/Database Administration
DBMS	Database Management System
DDD	Direct Distance Dialing
DDN	Defense Data Network
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
ETADS	Enhanced Transportation Automated Data System
ETR	Export Traffic Release
ETRR	Export Traffic Release Request

EOC	Emergency Operation Center
FD	Functional Description
FMS	Financial Management System
FORSCOM	U.S. Forces Command
GB	Gigabyte
GBL	Government Bill of Lading
GOSIP	Government Open Systems Interconnect Profile
GSA	General Services Administration
GTN	Global Transportation Network
GUI	Graphical User Interface
HHG	Household Goods
HP	Hewlett Packard
HQMTMC	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 Code
IOC	Initial Operating Capability
ITO	Installation Transportation Office
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
LOGSA	U.S. Material Command Logistics Support
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
MTIT	MTMC, Directorate of International Traffic
MTMC	Military Traffic Management Command
NAOMIS	Navy Material Transport Office
OCONUS	Outside the Continental United States
ODT	Open Desktop
OO	Object-Oriented
OODB	Object-Oriented Database
ORNL	Oak Ridge National Laboratory
PC	Personal Computer
PL	Procedural Language
PMO	Product Management Office
POD	Port of Debarkation
POE	Port of Embarkation
POSIX	Portable Operating System Interface for Computing Environments

POV	Privately Owned Vehicle
PPL	Point-to-Point Link
QA	Quality Assurance
QBF	Query-By-Forms
RAM	Random Access Memory
RBF	Report-By-forms
RDBMS	Relational Database Management System
RDD	Required Delivery Date
RORO	Roll-on/Roll-off
SA	System Administrator
SCAC	Standard Carrier Alpha Code
SDE	Shared Data Environment
SF	Standard Form
SITREP	Situation Report
SLIP	Serial Link Internet Protocol
SOCO	Shipping Order/Clearance Order
SPLC	Standard Point Location Code
SQL	Structured Query Language
STRADS	Strategic Deployment System
TAC	Transaction 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
TOLS	TERMS On-line System
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
WAN	Wide Area Network
WIN	WWMCCS Intercomputer Network
WPS	Worldwide Port System
WWMCCS	Worldwide Military Command and Control System
4GL	Fourth-Generation Language

SECTION 2. SUMMARY OF REQUIREMENTS

This section provides a summary of the system characteristics and requirements. It expands information contained in the ICDB Functional Description (FD) and provides additional system-specific details. This document reflects the operational system architecture.

2.1 SYSTEM/SUBSYSTEM DESCRIPTION

To effectively support visibility of worldwide cargo traffic, MTMC must integrate systems functions and data from WPS and IBS (both of which are currently in prototype status) and from currently operating automated and manual systems. The overall objectives of this system are to provide an integrated database for data management, data manipulation, communication, inquiry, and reports production. Figure 2.1 shows the relationships of the user organizations to ICDB in very general terms.

The system will track MTMC surface cargo that passes through common-user ocean ports. Through management of information associated with ocean cargo movement, ICDB will improve communication among MTMC systems and will provide requested information to non-MTMC systems (e.g., GTN). ICDB will also provide a centralized data repository that will permit queries on cargo status from offered through lifted [Initial Operating Capability -- (IOC)] and eventually to disposition (future plans). When fully implemented, ICDB will promote worldwide visibility over MTMC-controlled cargo traffic.

A specific hardware platform and communications networking designs have been determined. They are described in Sections 3.1 and 3.2. The database management system (Oracle7) and other programming tools are described in Section 3.3.

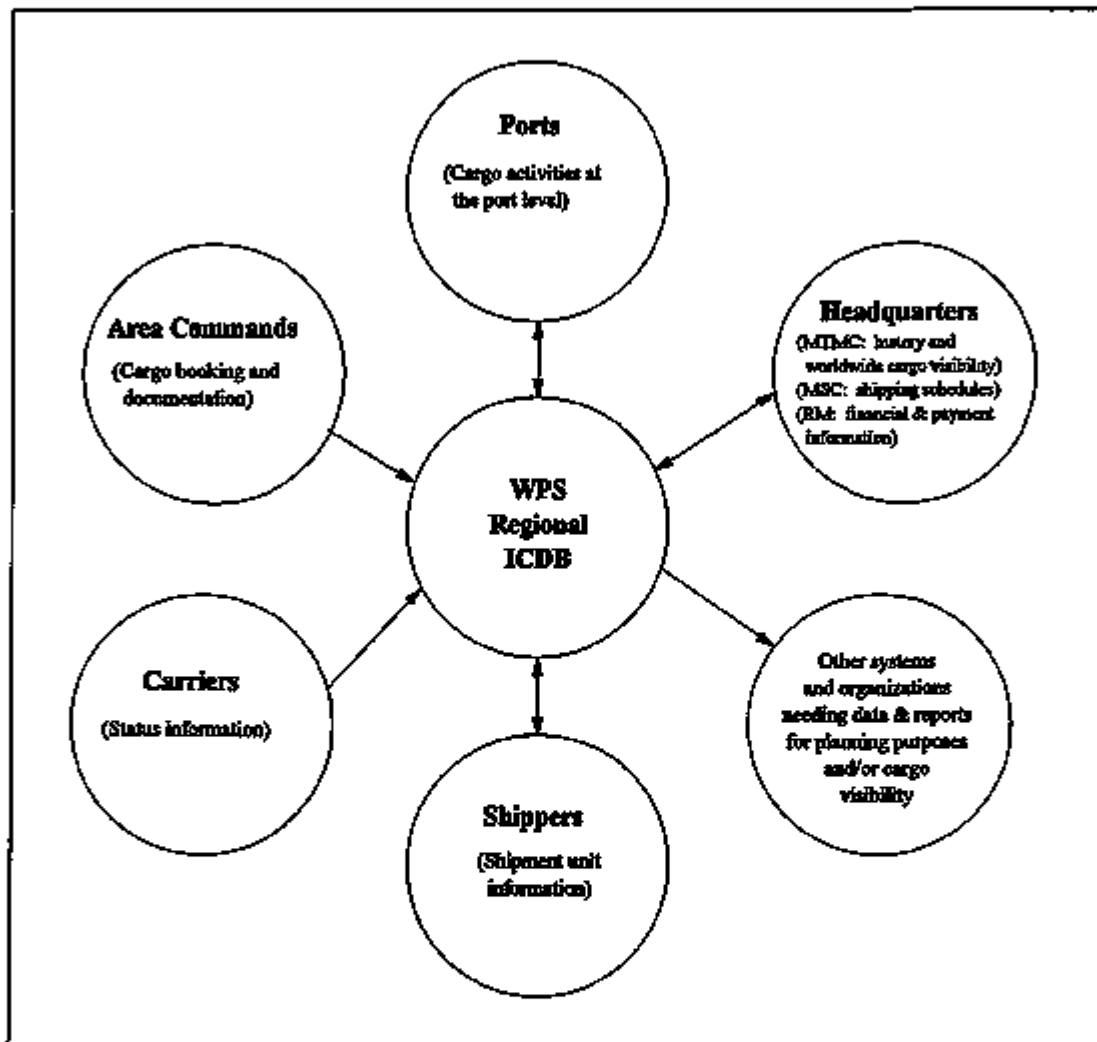


Figure 2.1. Relationships of the user organizations to ICDB.

At the present time, ICDB consists of a central database repository and two regional Hubs. The regional Hubs correspond to the geographical distribution of MTMC ocean cargo movement data (Eastern and Western Areas). Regional Hubs are connected to the central ICDB Server to support visibility over all ocean cargo movement data. The ICDB system has been designed to allow for many more regional Hubs than the two current CONUS regional Hubs. MTMC is planning to have Hubs for Europe and the Far East and other areas. It is also possible to quickly establish a "portable" Hub in a theater of operations as needed. The system software components presented in this document apply to all Hubs attached to the Central Server and to the Central Server itself. Figure 2.2 shows the Server/Hubs configuration in the continental United States (CONUS).

The ICDB system consists of three categories of processes: user interface processes, automated processes, and system administration/database administration processes. Each category of processes will consist of subsystems. These subsystems are described in the following sections.

2.2 SYSTEM/SUBSYSTEM FUNCTIONS

The ICDB system consists of three logical processing categories. The processing categories are user interface processes, automated processes, and system administration/database administration processes. Each of these process categories consists of one or more subsystems as follows.

The user interface processes allow the user to perform on-line builds of ATCMDs, to divert or resend manifests, to perform shipment unit (single or aggregate summary) queries of the database, to produce reports, or to perform regional processes. The user interface process category consists of the following subsystems:

1. ATCMD,
2. manifest,
3. inquiry,
4. reports,

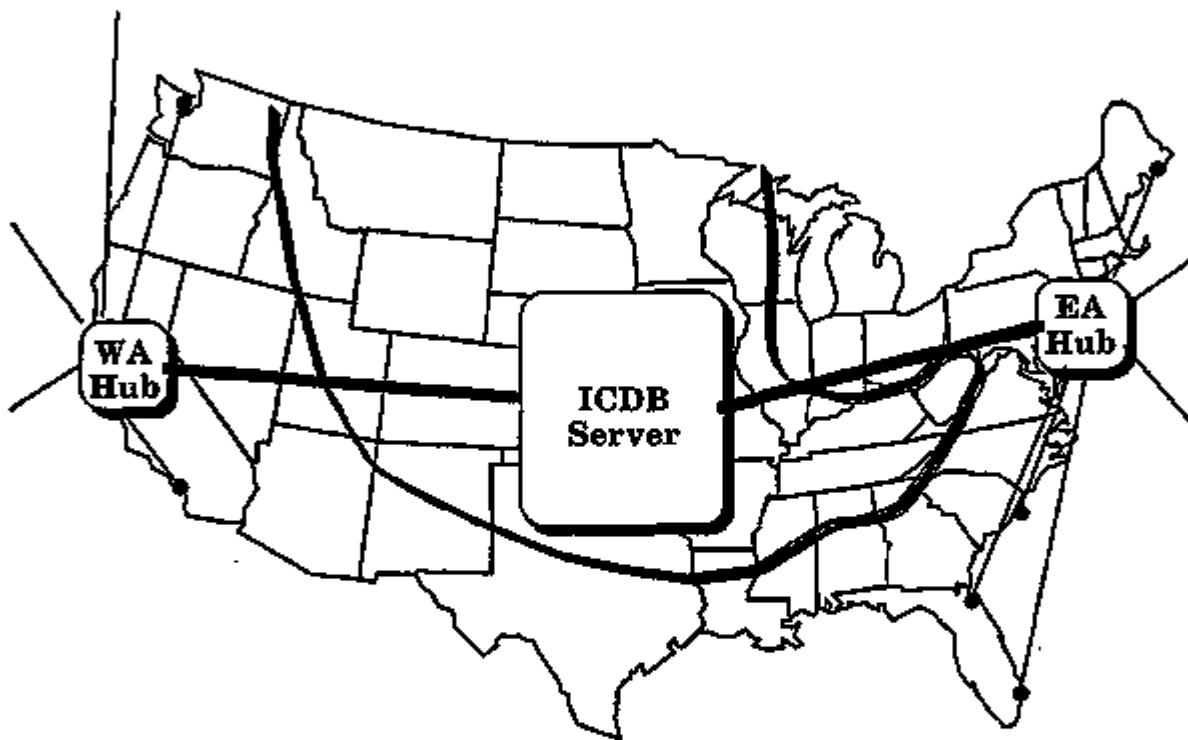


Figure 2.2 The ICDB hardware configuration.

5. regional processes, and
6. utilities.

The second category of ICDB processes are those that occur automatically. These include Unix scripts that are automatically "fixed" by the Unix utility, cron, and PL/SQL procedures that retrieve data from the WPS sites and Hubs, those that convert data into the ICDB format, scripts that extract data for transmission to other systems, and scripts that perform integrity checks on the system hardware and software. The ICDB automated processes are divided into the following subsystems:

1. data transfer,
2. data conversion,
3. data distribution, and
4. system operational integrity checks.

The third category of ICDB processes are those that help to monitor and maintain the ICDB system. These scripts are aids to be used by a system and/or database administrator. The system administration/database administration process category consists of the following subsystems:

1. Unix reports,
2. database administration reports,
3. audit reports,
4. user registration, and
5. maintenance.

More detail about the functionality of each of these subsystems is presented in Section 4.

2.2.1 Accuracy and Validity

Accuracy and precision is maintained throughout the system except where floating point numbers are involved. Wherever floating point is used, IEEE precision rules are followed. In the software design, floating point is NOT used whenever possible.

The system is equipped with input data editing routines. These routines test the contents against specific criteria. They do not allow any data into the database that does not meet the rules for accuracy and validity.

Standard data transmission protocols are responsible for telecommunications error checking ensuring the validity and accuracy of data transferred between the ICDB/WPS components.

2.2.2 Timing

The response time for queries should be on average less than 5 seconds for simple queries and less than 1 minute for more complex queries. If a query takes longer than 5 seconds to process, the system will provide a "Working ..." message to the user. More information about target performance timeframes for queries is provided in Section 3.1.2 of the ICDB FD (Table 3.1).

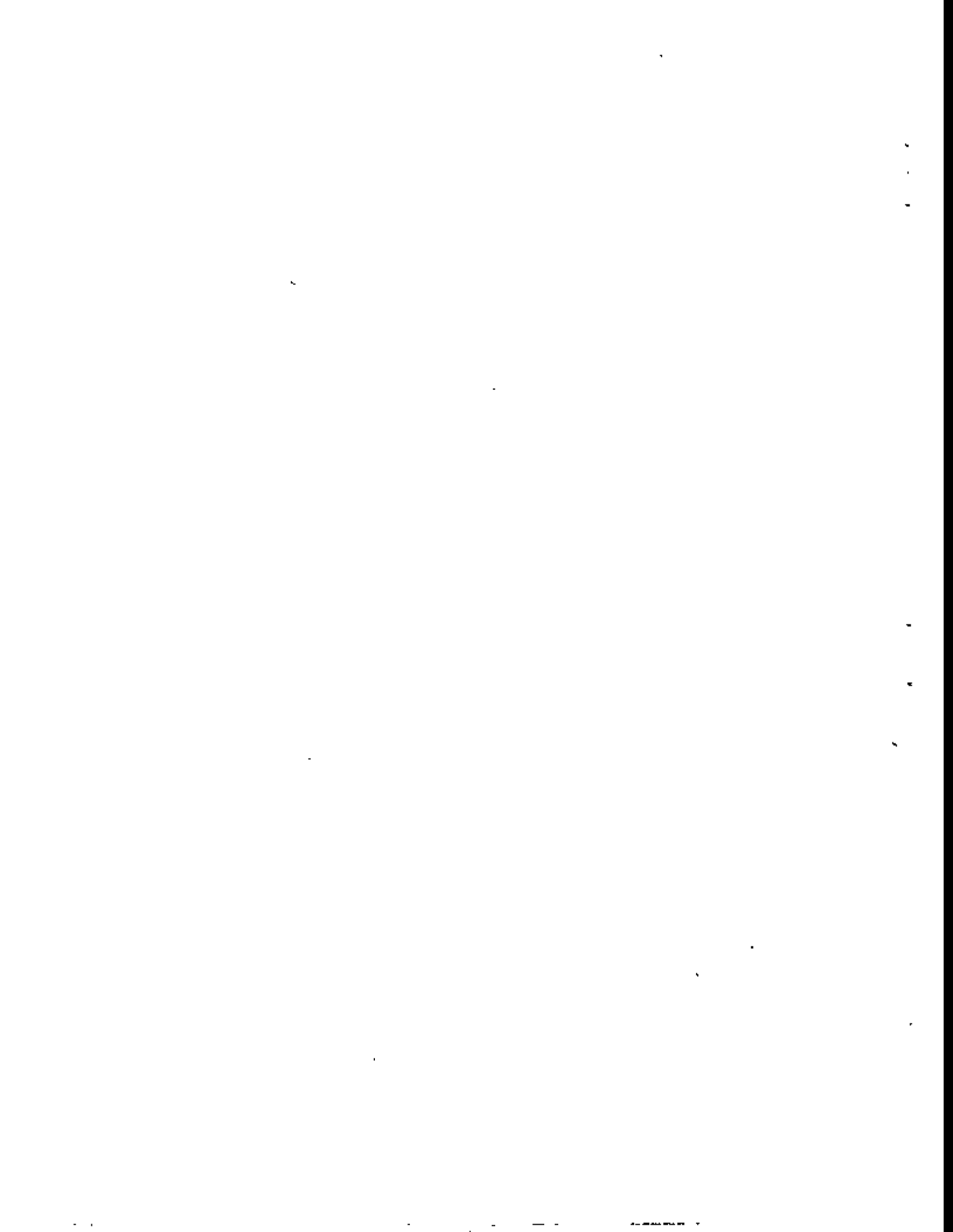
The system has the capability of dynamically prioritizing communications traffic to ensure that the highest priority communications receive any needed resources. Under certain circumstances, some users may be assigned a lower priority for ICDB query capability. This situation would most likely occur under wartime or full mobilization.

2.3 FLEXIBILITY

The ICDB system hardware and software will be designed and configured in such a way as to accommodate any economically feasible computing telecommunications technological advances, changes in operations, or changes in responsibility.

The ICDB system is designed around the Government Open Systems Interconnect Profile, second edition (GOSIP II) and its associated standards. As such, changes in hardware or software should have minimal impact on the mission of ICDB. Expansion and longevity are integral goals of ICDB system design. ICDB is not considered mission-critical but must have higher than normal availability during exercises and mobilization. This means that the ICDB system must have assured high availability.

ICDB CONUS architecture is split over two Hubs. Each Hub can operate independently for a period of time isolated from the Server or any other Hub. This time allows the failed Hub or communications links to be repaired and brought back on-line. This design also allows configuration flexibility and easy expansion, including, for example, possible addition of Hubs in the Far East or Europe and/or temporary additions of portable Hubs.



SECTION 3. ENVIRONMENT

This section provides an expansion of the environment briefly described in the FD and presented in more detail in the ICDB Architectural Analysis to reflect the additional analysis and changes to the environment.

ICDB is a multi-user automated system based on client/server, relational database technology and hosted on Unix workstations in accordance with AUTOSTRAD-2000 guidelines. Full functionality of the system will, therefore, include all the standard facilities offered by this environment (e.g., electronic mail, text editors, programming tools).

3.1 AIS EQUIPMENT ENVIRONMENT

ICDB conforms to the AUTOSTRAD-2000 philosophy and directives for open system architecture. ICDB is built around the client/server technology. Unix is the operating system.

The ICDB system has initially been installed in a development environment at ORNL on Hewlett Packard hardware. Each of the machines involved, the HP 750 for the Server and the Hub and the HP 710 for the Site, are single Unix systems. Final installation of the ICDB system has been performed on the production hardware, which for the Server and Hubs consists of HP 877 machines configured with four HP 730 compute servers and one HP 710 console. For further details about hardware configuration, see the ICDB System Operation Manual.

3.2 COMMUNICATIONS ENVIRONMENT

The ICDB system is part of the larger WPS system, which consists of a three-tier client/server architecture. The bottom tier consists of WPS port sites that connect to regional processing Hubs, which make up the middle tier. The top tier is the central ICDB database Server, which communicates with the Hubs. The flow of data is from the port sites up to the processing Hubs,

which will send the data to be stored on the central ICDB Server. Data also flows from the Hubs down to the port sites and to the central Server. In a few cases (e.g., sending of ACTMDs, manifest diversions) data flows from the Server to the Hubs to the port. There are also several user-initiated processes (e.g., ATCMD On-Line Builds) that originate on the regional Hubs. Data from these processes flow to both the port sites and to the central Server.

The WPS Regional ICDB system encompasses the central Server and the regional Hubs. Users will generally access the Hubs via network or dial-up access. The Hubs initiate all data transfers between all tiers of the architecture. The Hubs "pull" data from or "push" data to the WPS port sites. The Hubs "pull" data from or "push" data to the Server. All data transfer transactions use Oracle's SQL*Net v2. Most transactions on the Server are handled remotely through the Hub using SQL*Net v2.

Most system administration functions are implemented on the Central Server with the capability of distributing the correct administration actions to the Hubs and the WPS port sites. The administration tools are intelligent and can aid the administrators in diagnostics and administration of the ICDB system.

3.2.1 Network Description

The central ICDB Server resides at a Washington D.C. site and consists of a large HP 877 Super Mini computer system configured as a single system. The computer runs the HP/UX Unix operating system and communicates with the regional Hubs. Figure 3.1 shows the network configuration for the Server.

The top two levels of the WPS ICDB machines are connected by two methods: ethernet and phone. The central Server and each of its Hubs are connected to Cisco gateways via the ethernet ports on each machine. The Ciscos determine the best way to connect the machines. The primary method is via dedicated lines. If these fail, the Ciscos will attempt to establish a link using the Internet/Milnet system. If this fails, the machines themselves will link to each other using phone lines. Users can access the Hubs via network access (HQ users can access the Server). In addition, dialup lines are available for access to the Hubs.

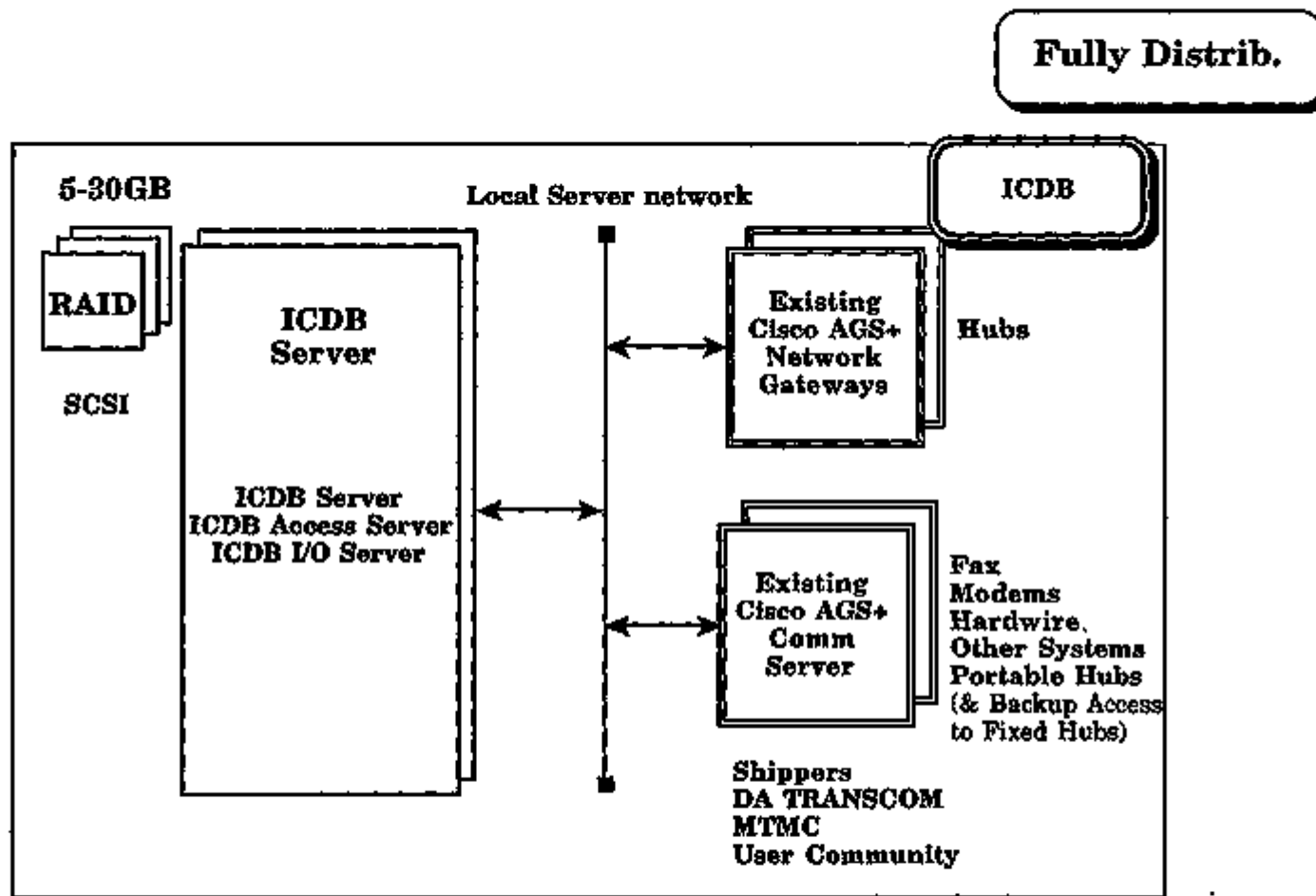


Figure 3.1. Distributed ICDB Server configuration.

Initially, two Hubs are employed. For IOC, an East Coast Area Hub is located at Bayonne, NJ, and a West Coast Area Hub is located at Oakland, CA. These Hubs are similar to the central Server but are configured more for communications and user interface processing. Figure 3.2 shows the network configuration for the Hubs.

The Hubs communicate via direct lines with each of the WPS port sites, which consist of smaller HP systems. In the event that the direct links to the WPS port site fail, the Hub can dial the WPS port sites and still extract data.

For a more detailed view of the ICDB architecture, please refer to the ICDB Architectural Analysis document.

3.2.2 Physical Interface

At each physical site in the WPS system, the ICDB system components are configured on a local LAN which is insulated from the outside world via a Cisco gateway processor. The Cisco will route only allowed traffic into the local LAN and will route outgoing packets either via the fractional T1 lines or via the Internet connection. If both of these options are unavailable, then serial dial-up modems can be employed to communicate between the various sites and maintain the "will to live" philosophy (i.e., system-initiated, system-wide survivability) employed by the ICDB system. SCSI-connected serial ports will be available for use by dial-in users.

3.2.3 Protocol Interface

The networking protocol used by the ICDB system is the Unix industry standard TCP/IP. Network traffic is routed via dedicated fractional T1 lines. The Cisco will handle routing to either the dedicated line or via the Internet connection.

If the Cisco is unable to transmit, the fall back method is to communicate via dial-up modem. In this case, the SLIP interface or one of its variations will be used. The programming of SLIP connections has been accomplished using the PPL subsystem.

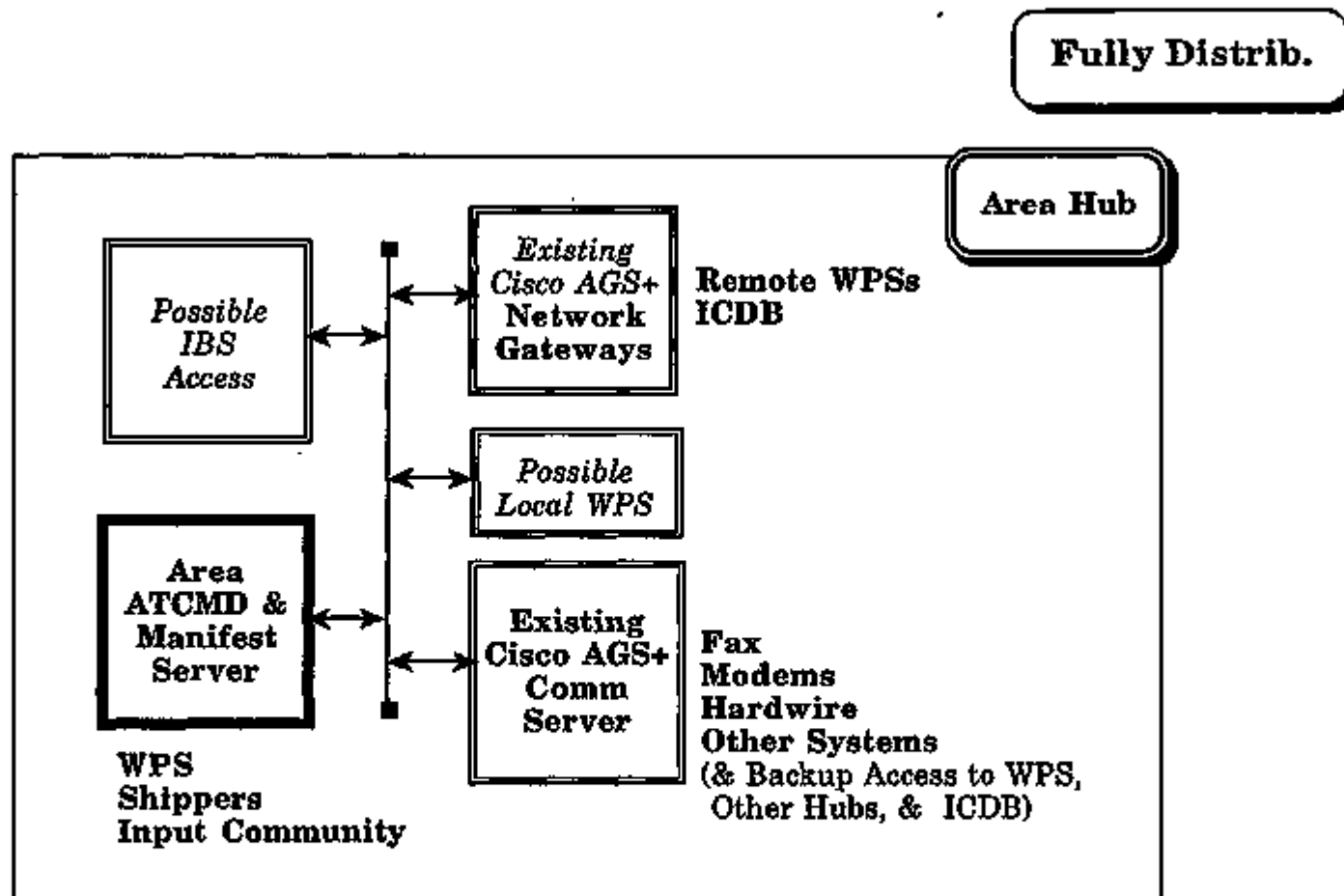


Figure 3.2. Distributed ICDB area Hub configuration.

3.2.4 Applications User Interface

The communications modules are generally executed from a cron process at scheduled times. These processes will do a variety of things. Transfer of data between computer systems is accomplished through one set of procedures, which are Bourne shell scripts that use Unix commands, SQL*Plus, PL/SQL, and SQL*Net v2. Other Bourne shell scripts check the stability of the network connections and manage the "will to live" functions. These processes are more fully described in Section 4.4.2 of this document.

System administrator scripts are available to monitor the communications processes. They are available at both the command line and via a system administrator's menu interface.

3.2.5 Diagnostics

The HP/UX operating system provides methods for monitoring process and device usage. Additionally, several system administration scripts are available to help monitor the usage of the database and of the Unix system itself.

PRC Inc., the vendor of the HP Super-Minicomputer System, provides single system view software for both console management, control file management, and load balancing. They also provide database support to the ICDB project.

3.3 SUPPORT SOFTWARE ENVIRONMENT

ICDB conforms to the AUTOSTRAD-2000 philosophy and directives for open architecture. The Oracle relational database management system (DBMS) is used for ICDB, and Oracle CASE tools are also used. Additional software packages include Ada, a third-generation language; Oracle's SQL*Forms for the front-end user interface; Oracle's SQL*Net communication software; and Oracle's Reportwriter. Other tools or upgrade versions of these tools from Oracle (e.g., those for forms, reports, and graphics) may be tightly integrated with these packages.

In addition, ICDB fully utilizes the capabilities of the Unix operation system, Unix scripting tools, Unix diagnostic and reporting tools, and Unix process controlling tools. By utilizing the full spectrum of Unix, ICDB requires no 3GL language support beyond Ada.

3.4 SOFTWARE INTERFACES

Figure 3.3 shows the systems and organizations with which ICDB interfaces.

Procedures provide both manual initiation of message sending and automatic sending and receiving of messages. For more information on ICDB data exchanges with other systems, see Section 4.4.2.4.

3.5 SECURITY

System security must be considered at four different levels: (1) physical access to the ICDB computer equipment, (2) network and modem access to the systems, (3) Unix user access, and (4) Oracle user access.

3.5.1 Physical Access

The various components of the ICDB system must be physically guarded so that no one can gain access to the system disk drives (whether to destroy or simply read them). If access to the system disk drives is gained, the Unix system can be accessed directly, with or without a password.

All persons with physical access to the systems must be authorized by a security authority of MTMC. Guests must be accompanied by authorized persons. The system must be defended from physical access by unauthorized persons.

3.5.2 Network and Dial-up Access

Logins to the ICDB system must also be monitored carefully. There are two ways to login to the ICDB systems: first via the Internet or dedicated line and second via dial-up modem. Access to the

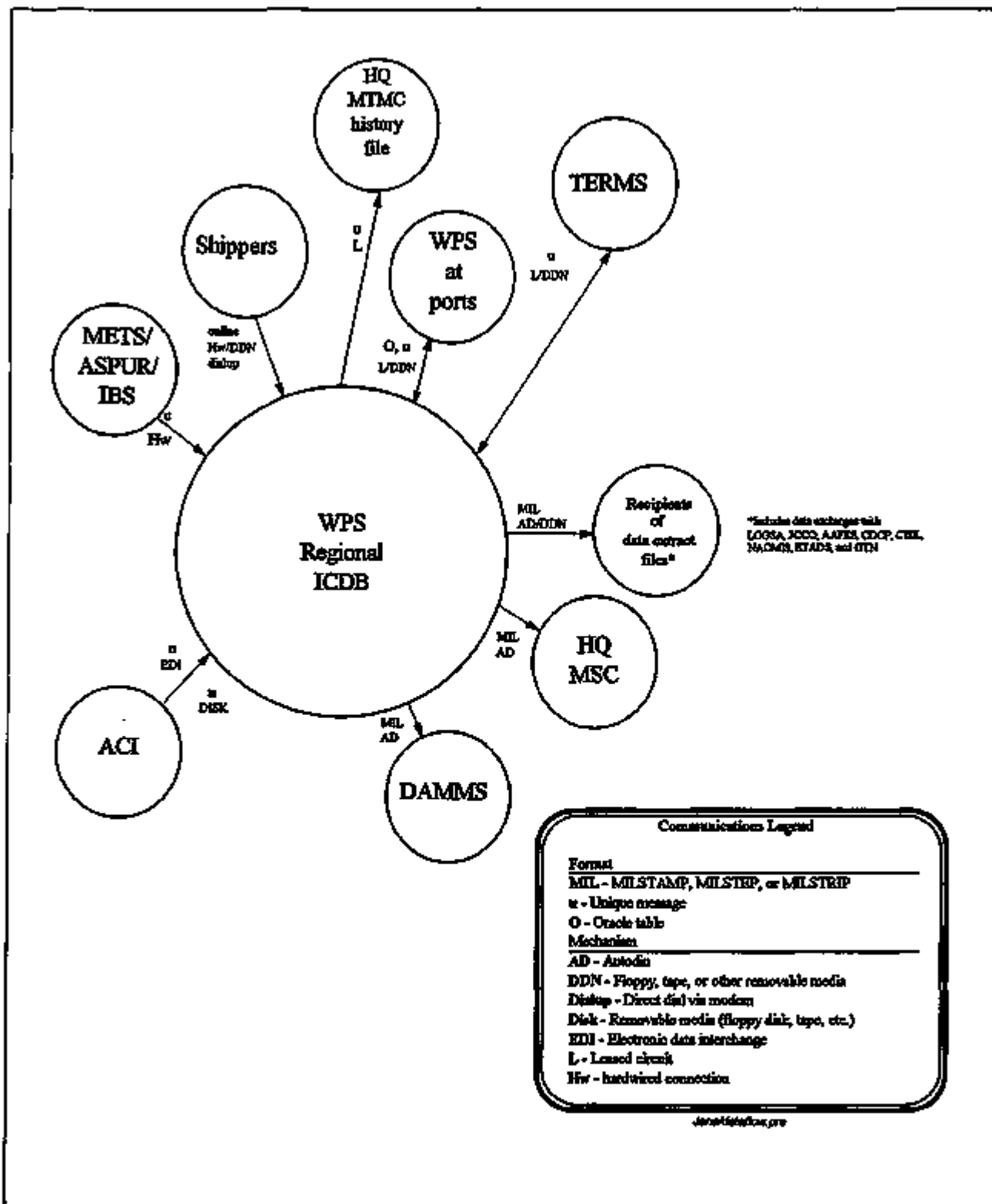


Figure 3.3. The systems and organizations with which ICDB interfaces.

system is protected first by user ID, then by secret password. Each user has the responsibility of protecting his ID and password so that unauthorized access cannot be gained by compromise. IDs and passwords can be changed as often as necessary to maintain secure access.

3.5.3 Unix User Access

Regardless of how a potential user gains access to the ICDB system, there is no way to get on without entering a username and a password (unless the user gains access to the disk drive itself). Access to the ICDB central server will be severely restricted. That is, only system administrators and some MTMC-HQ users will have an account on the Server. All other ICDB users will logon to the Hubs. This provides an extra layer of protection for corporate data.

Attempts to "hunt" for a password will be logged, and the security administrator will be notified. The HP-UX system allows only three attempts at logging in. After that, the user line (or telnet session) is disconnected and a notation is made in a failed login file. This file can be reviewed for identifying patterns or tracing sources of attempts to break into ICDB.

3.5.4 Oracle User Access

Once on the Unix system, users, unless they are system administrators or database administrators, will be placed immediately inside the ICDB Menu system. No Unix access will be allowed. Users are assigned to a particular role when they are registered, and they will be limited, accordingly, as to what they can do in the ICDB system.

For the majority of users, access to the central Server will be accomplished only through SQL*Net and the ICDB menus and procedures.

As a second layer of security, access to files which provide information about ICDB database links will be restricted. As always, the final security of the system rests on the users who are chosen to have direct access to the system.

SECTION 4. DESIGN DETAILS

This section describes how ICDB satisfies the requirements delineated in Sections 2 and 3. This section also includes technical requirements when these do not relate directly to the functions and performance seen by the user and have not therefore been described in Section 3. All of the software units that are components of the subsystems discussed in Section 3 are identified in this section.

4.1 GENERAL OPERATING PROCEDURES

The Unix System Administrator's Manual, provided with the purchase of the specific hardware of the operational system configuration, specifies Unix-level system administration procedures. A System Administration/System Operation Report was prepared for the developmental system in order for the three sites involved in ICDB development to coordinate efforts. A System Operation Manual Draft was produced in February 1994, and a Maintenance Manual was produced in June 1994.

4.2 SYSTEM LOGICAL FLOW

Each ICDB system component is implemented as a hierarchical series of menus and processes to accomplish the functions delineated in the ICDB FD. Where appropriate, such as in the ATCMD build module, the logical flow corresponds to the logical flow of processes in WPS, as described in WPS System/Subsystem Specification, Rev. 1, July 1992. Thus, users of both systems are presented a consistent interface for building on-line ATCMDs.

Charts presenting the logical flow of other functions of ICDB are included in Section 4.4 for each ICDB subsystem.

4.3 SYSTEM DATA

Receipt and output of data are primary functions of ICDB. Both input and output of data is in the form of ASCII files, messages between ICDB and other systems, and Oracle tables as described in Section 3.4. In addition, ICDB will output data in the form of reports and responses to queries.

4.3.1 Inputs

The following table shows data sets that are received by ICDB.

Table 4-1. Data sets to be received by ICDB*

Data set	Sending system	Format reference
ATCMD (skeleton)	METS/IBS	MILSTAMP
ATCMD	ASPUR/IBS	MILSTAMP
ATCMD, batch	Shippers	MILSTAMP
Shipment unit data	WPS	Oracle tables
Export manifest	WPS	MILSTAMP
Import manifest	WPS and others	MILSTAMP
Cargo traffic message	WPS	MILSTAMP
Reject message	WPS	Oracle table
Status	WPS	Oracle table
Status	ACI	Flat file
Status from POD	WPS or on-line access	MILSTAMP

*FMS data is pushed from WPS ports to FMS through Hub communication processes and is not entered into the central Server tables.

4.3.2 Outputs

Data transmission files, query responses, and reports that are output from ICDB are discussed in the following sections.

4.3.2.1 Data Transmission Files. The following table shows data files that are transmitted by ICDB as data output. Specific formats for unique messages are provided in Interface agreements between ICDB and the receiving system.

Table 4-2. Data sets to be sent from ICDB*

Data set	Receiving system	Format reference
Shipment unit data	WPS	Oracle tables
Import manifest	WPS	Oracle tables
Manifests	Various	MILSTAMP
Cargo traffic message	WPS and non-WPS ports	MILSTAMP
ATCMD	ETADS/NAOMIS	MILSTAMP
History	HQ MTMC	Unique message
Receipt, lift, discharge, disposition, vessel status, hazardous, and explosive	LOGSA	Per interface agreement
Receipt, lift, vessel status, hazardous, and explosive	NAOMIS	Per interface agreement
Receipt, lift, vessel status, hazardous, and explosive	CISIL	Per interface agreement
Receipt, lift, vessel status, hazardous, and explosive	ETADS	Per interface agreement
Receipt, lift, discharge, disposition and vessel status	GTN	Per interface agreement
Receipt, lift, discharge, disposition for containers	ICCO	Per interface agreement
Receipt, lift, and vessel status	AAFES	Per interface agreement
Discharge	CDCP	Per interface agreement

*FMS data is pushed from WPS ports to FMS through Hub communication processes and is not entered into the central Server tables.

4.3.2.2 Queries. Query capabilities for standard queries and ad hoc queries are provided. Fields upon which queries can be keyed are listed in Appendix A, "Detailed System Requirements," of the ICDB FD.

4.3.2.3 Reports. A list of reports that are required products of ICDB is given in Appendix A, "Detailed System Requirements," of the ICDB FD.

4.3.3 Database/Data Bank

Portions of the ICDB data (e.g., manifests) reside in flat files in specified formats. These datasets are maintained in the ICDB as part of the Unix file system and are distributed by ICDB to other

systems, but they will not be retained in the ICDB DBMS. The remaining ICDB data is stored in an Oracle DBMS. The details of this database are described fully in the ICDB Database Specification document.

4.4 SOFTWARE UNIT DESCRIPTION

The complete ICDB system consists of three categories of processes: user interface processes, automated processes, and system administration/database administration processes. Figure 4.1 shows the ICDB process categories. Each process category consists of several subsystems, which are discussed in the following sections.

4.4.1 User Interface Processes

User interface processes are those processes that allow the user to see data in the ICDB database displayed in a logical manner. The user interface processes consist of six subsystems as shown in Figure 4.2: the ATCMD subsystem, the Manifest subsystem, the Inquiry subsystem, the Reports subsystem, the Regional Processes subsystem, and the Utilities subsystem.

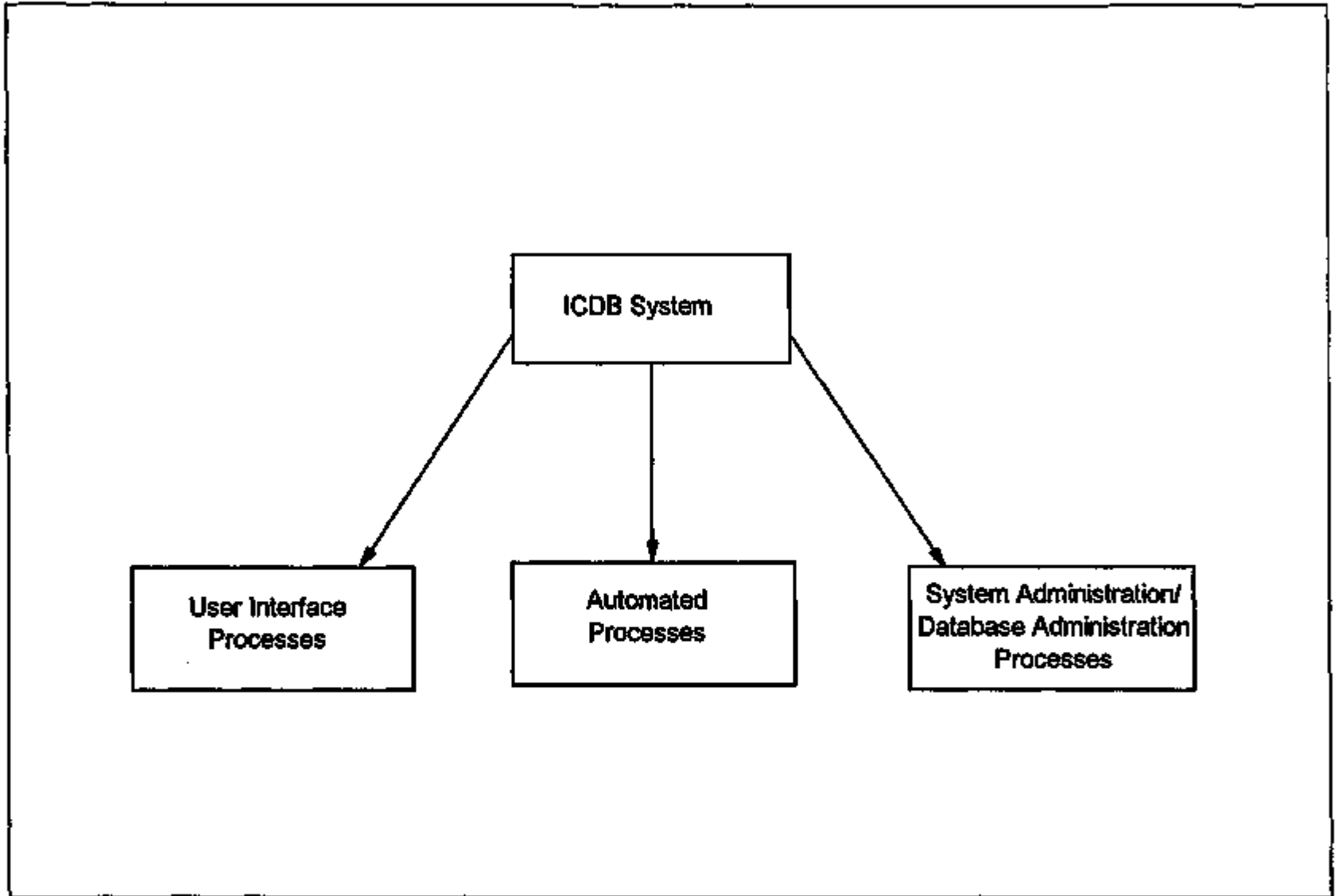


Figure 4.1. The ICDB process categories.

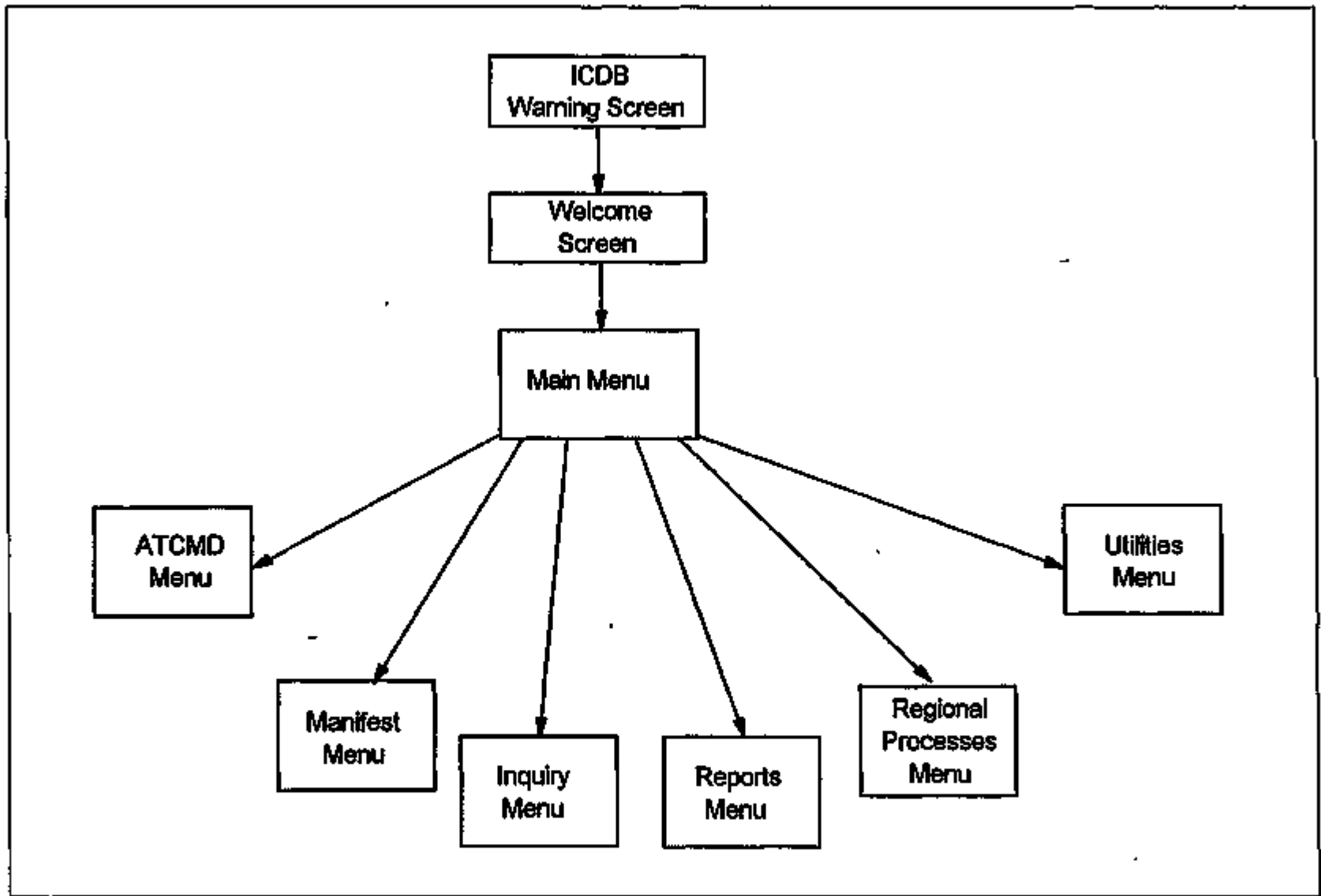


Figure 4.2. User Interface processes subsystems.

4.4.1.1 The ATCMD Subsystem. The ATCMD subsystem consists of a series of processing modules which preprocess flat file ATCMD data, load it to ORACLE, and convert it from MILSTAMP multiple 80-column record format to WPS database format. Additional modules facilitate expansion of multiple vehicle shipments and reinput of former rejects corrected by users. Figure 4.3 provides a block diagram of the user interface screen flow. Screens are provided for low-volume users to build ATCMD documents and for Area Command personnel to reforward previously received ATCMD documents to WPS.

Software Unit Name: preproc
Software Unit Type: Ada
Software Unit Description: This program pre-processes flat file ATCMD data, validates numeric fields, and connects GBL headers with their details. The program also rejects records with unknown Document Identifier Codes and has a interim routine to shunt data to the TOLS system based on a user maintained table.

Software Unit Name: reinput.sql
Software Unit Type: SQL script
Software Unit Description: This module removes records marked by the atcmdcor.frm from the ATCMD reject table and reinputs them to the ATCMD process.

Software Unit Name: xplov.sql
Software Unit Type: PL/SQL script
Software Unit Description: This module expands properly coded government vehicle shipments with multiple piece counts to multiple records in accordance with MILSTAMP.

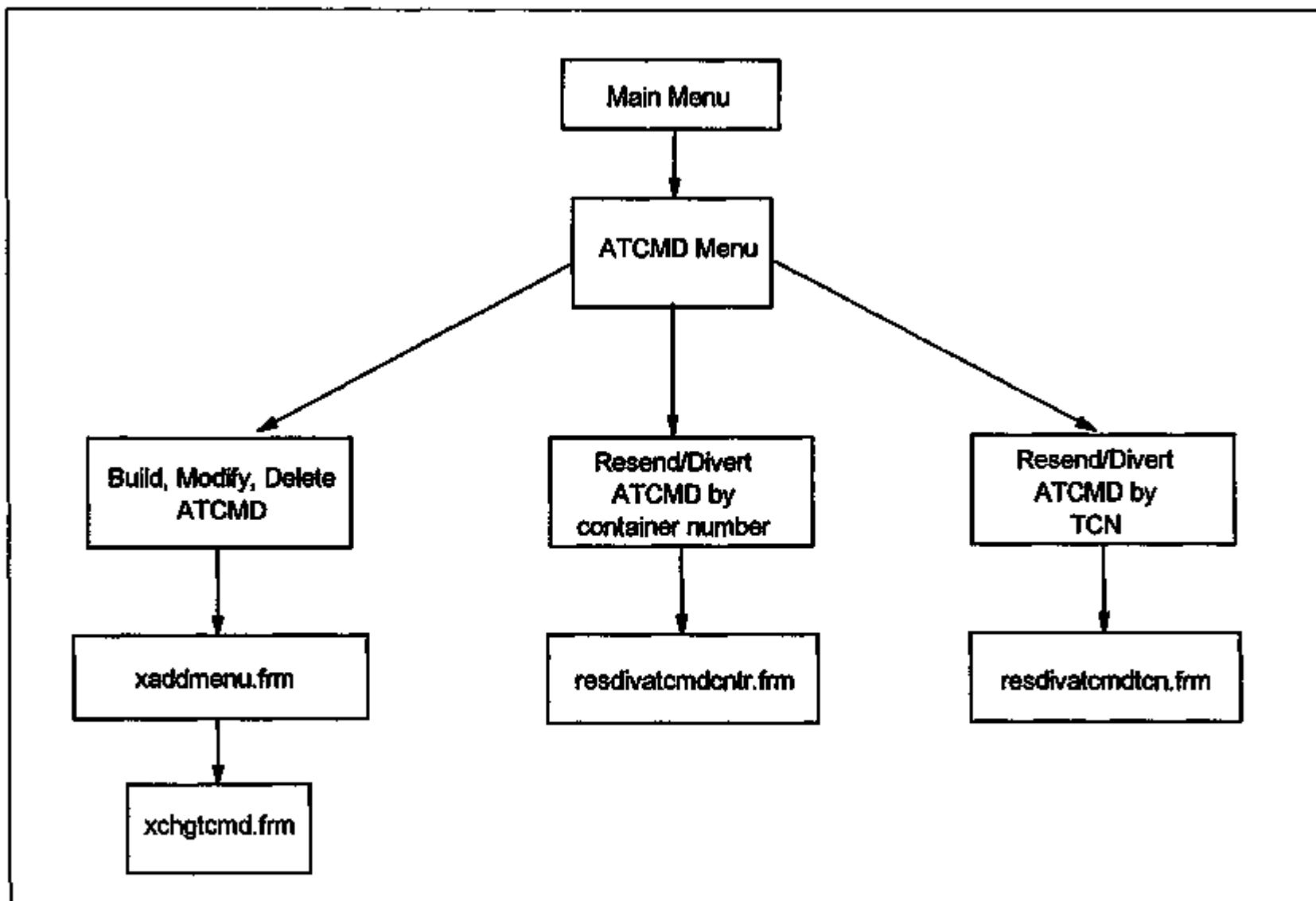


Figure 4.3. The ATCMD subsystem.

Software Unit Name: prbreak.sql
Software Unit Type: PL/SQL script
Software Unit Description: This module breaks out ATCMD prime records to the WPS format, linking containers to contents and rejecting records with invalid TCN, package code, POE, and contents without an associated prime record.

Software Unit Name: trbreak.sql
Software Unit Type: PL/SQL script
Software Unit Description: This module breaks out ATCMD trailers to WPS format, links trailer records to their prime records, and rejects trailers without associated prime records.

Software Unit Name: umove.sql
Software Unit Type: PL/SQL script
Software Unit Description: This module populates the WPS unit move table from the unit move trailers.

Software Unit Name: bldrej.sql
Software Unit Type: PL/SQL script
Software Unit Description: This module moves rejected ATCMD records to the reject table for possible correction.

Software Unit Name: areject.sql
Software Unit Type: SQL*PLUS script
Software Unit Description: This module prepares a report of rejected ATCMD records.

Software Unit Name: xaddmenu.frm
Software Unit Type: SQL*form
Software Unit Description: This is the menu form for xchgcmd.frm.

Software Unit Name: xchgcmd.frm
Software Unit Type: SQL*form
Software Unit Description: This is the WPS format build TCMD form.

Software Unit Name: resdivatcmdtcn.frm
Software Unit Type: SQL*form
Software Unit Description: This form allows the user to select and mark by TCN shipment_unit records to be resent or diverted.

Software Unit Name: resdivatcmdcntr.frm
Software Unit Type: SQL*form
Software Unit Description: This form allows the user to select and mark by container number shipment_unit records to be resent or diverted.

Software Unit Name: su_div_res_e
Software Unit Type: ORACLE stored procedure
Software Unit Description: The su_div_res_e stored procedure is executed from the 5-minute hub cron processing. The su_div_res_e procedure selects records from the su_div_res view. For each su_id selected, data is selected from the shipment_unit table and records are inserted into the hub_xshipment, hub_xoutsize, hub_xnsn, hub_xexplosive, hub_xremarks, hub_xunit_move, and the hub_xstopoff tables on the hub.

4.4.1.2 The Manifest Subsystem. The Manifest subsystem consists of a series of processing modules which preprocess flat file MILSTAMP manifest data, load it to ORACLE, and convert it from MILSTAMP multiple 80-column record format to WPS database format. Additional modules appropriately mark records placed in resend status or diverted by users. Figure 4.4 provides a block diagram of the user interface screen flow. Screens are provided for Area Command personnel to resend or divert previously received manifests to WPS.

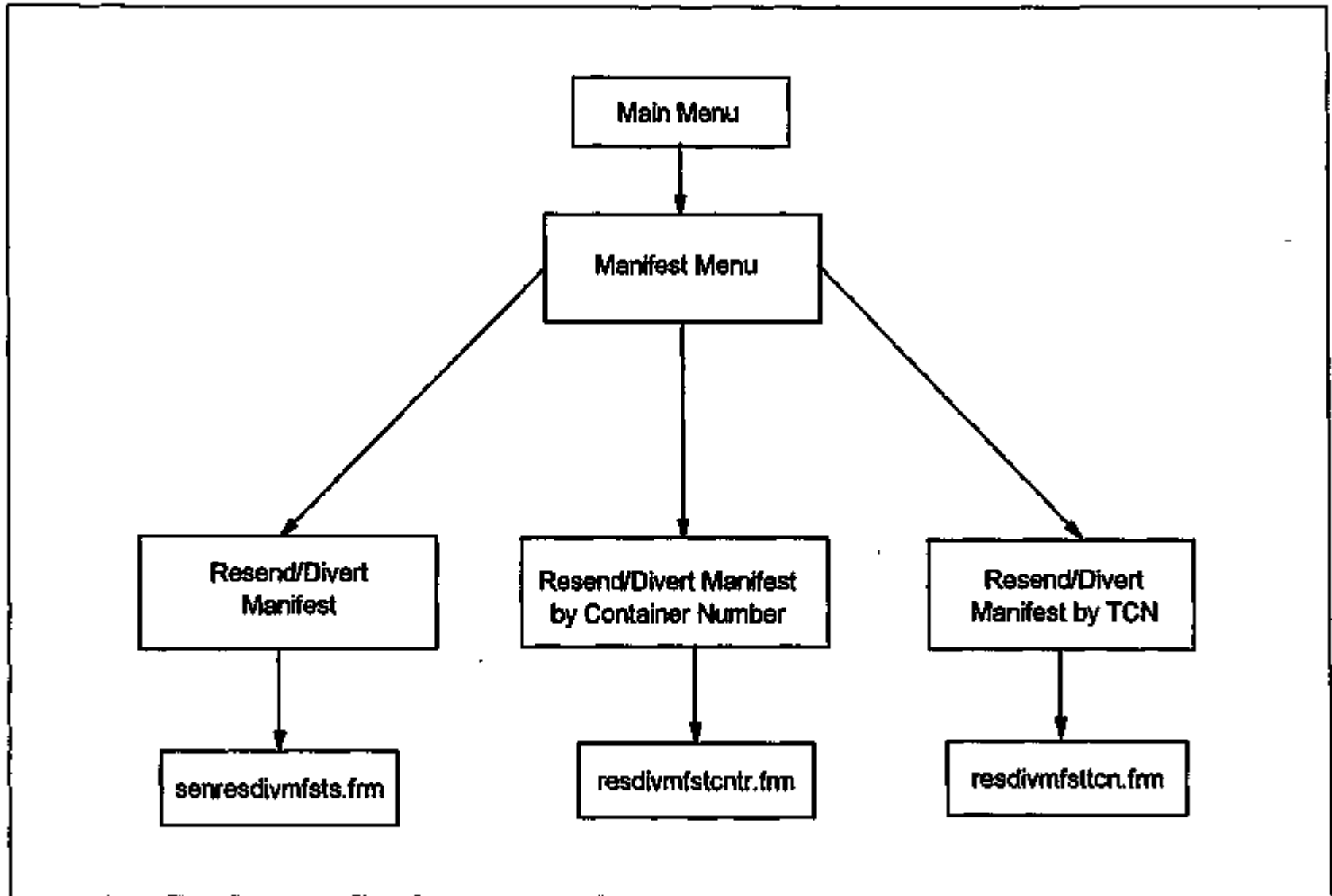


Figure 4.4. The Manifest subsystem.

Software Unit Name: `impreproc`
Software Unit Type: `Ada`
Software Unit Description: This program pre-processes flat file manifest data and validates numeric fields. The program also rejects records with unknown Document Identifier Codes and has a interim routine to shunt data to the TOLS system based on a user-maintained table.

Software Unit Name: `imprbreak.sql`
Software Unit Type: `L/SQL script`
Software Unit Description: This module breaks out manifest prime records to the WPS format, linking containers to contents and rejecting records with invalid TCN, package code, POD, and contents without primes.

Software Unit Name: `intrbreak.sql`
Software Unit Type: `PL/SQL script`
Software Unit Description: This module breaks out manifest trailers to WPS format, links trailers to their primes, and rejects trailers without primes.

Software Unit Name: `immove.sql`
Software Unit Type: `PL/SQL script`
Software Unit Description: This module populates the WPS unit move table from the unit move trailers.

Software Unit Name: `senresdiv.frm`
Software Unit Type: `SQL*form`
Software Unit Description: The form allows the user to enter resend or divert transactions for previously processed manifests.

Software Unit Name: `resdivmfstcn.frm`

Software Unit Type: SQL*form
Software Unit Description: This form allows the user to select and mark by TCN shipment_unit records to be resent or diverted.

Software Unit Name: resdivmfstcntr.frm
Software Unit Type: SQL*form
Software Unit Description: This form allows the user to select and mark by container number records to be resent or diverted.

Software Unit Name: su_div_res_i
Software Unit Type: ORACLE stored procedure
Software Unit Description: The su_div_res_i stored procedure is executed from the 5-minute hub cron processing. The su_div_res_i procedure selects records from the su_div_res view. For each su_id selected, data is selected from the shipment_unit table and records are inserted into the hub_shipment, hub_outsize, hub_nsn, hub_explosive, hub_remarks, hub_unit_move, and the hub_stopoff tables on the hub.

Software Unit Name: mfst_div_res
Software Unit Type: ORACLE stored procedure
Software Unit Description: The mfst_div_res stored procedure is executed from the 5-minute hub cron processing. The mfst_div_res procedure selects records from the mfst_div_resend view. For each voydoc, poe, pod, vstat combination selected, data is selected from the shipment_unit table and records are inserted into the hub_shipment, hub_outsize, hub_nsn, hub_explosive, hub_remarks, hub_unit_move, and the hub_stopoff tables on the hub.

4.4.1.3 The Inquiry Subsystem. The Inquiry subsystem consists of four subfunctions as shown in Figure 4.5: single shipment queries, multiple shipment - detail queries, multiple shipment - summary queries, and ad hoc queries.

Single Shipment Subfunction

Figure 4.6 shows the logical flow for the software units that are components of the single shipment subfunction. The following are the software units for this subfunction.

Software Unit Name:	cntssqry.frm
Software Unit Type:	SQL*Form
Software Unit Description:	This form displays three fields, TCON, CNTR_OWNER and CNTR_NBR, which will be used as user input for single shipment queries. The user must enter a value in at least one of the fields provided. The value entered can either be a partial value or a value encompassing the entire length of the field.
Software Unit Name:	cntssdtl.frm
Software Unit Type:	SQL*Form
Software Unit Description:	This is a display only form called by 'cntssqry'. It displays shipment unit information about containers and any contents. The data is retrieved based upon previous user specifications passed from the calling form. The values are used in building part of the where condition.
Software Unit Name:	tcnssqry.frm
Software Unit Type:	SQL*Form
Software Unit Description:	This form will accept user input to the TCN field, Transportation Control Number. Because this field is 17

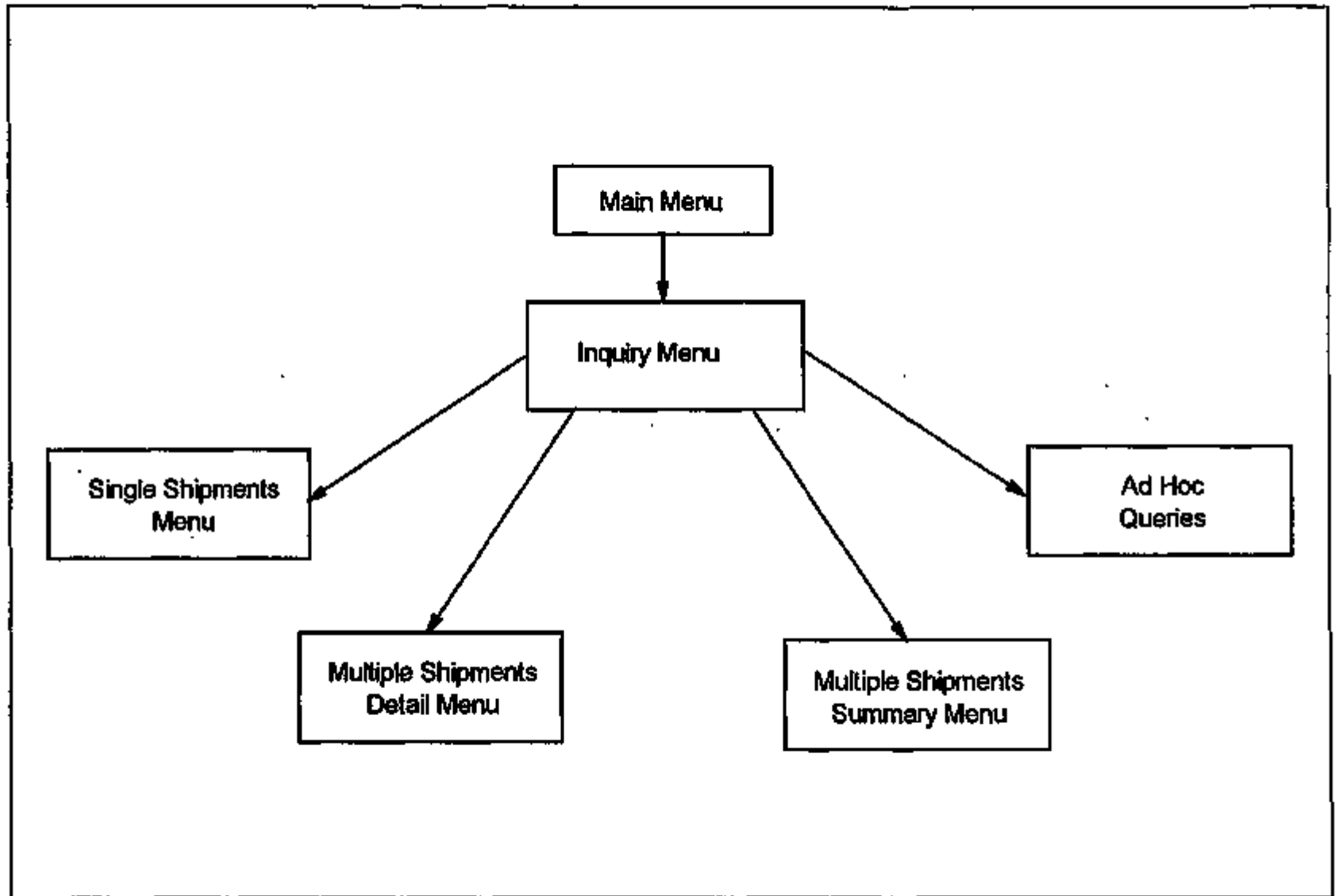


Figure 4.5. The Inquiry subsystem.

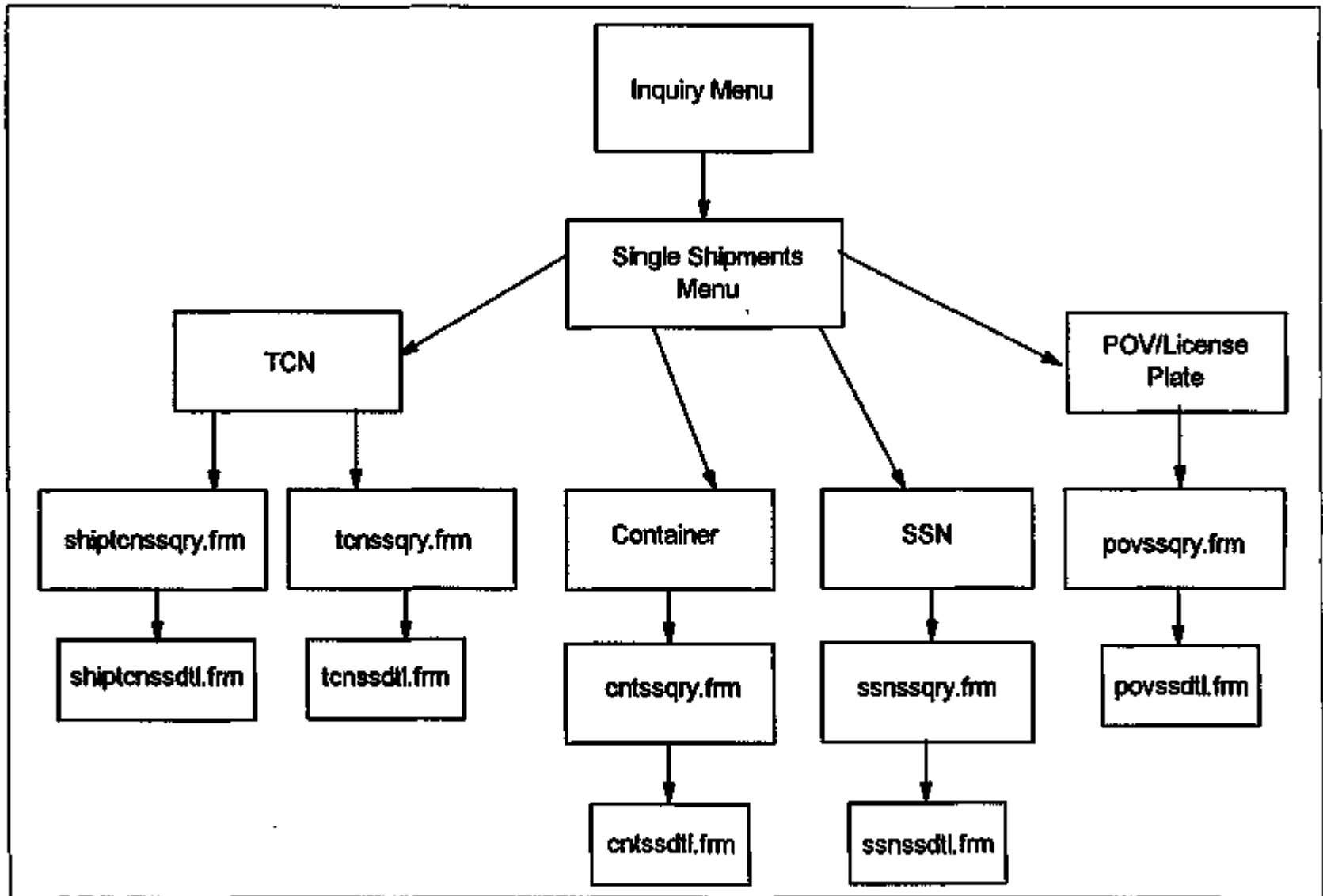


Figure 4.6. The Single Shipments subfunctions.

characters long, the user may enter a partial value as well as the entire field value.

Software Unit Name: tcnssdtl.frm
Software Unit Type: SQL*Form
Software Unit Description: This form is specifically designed for display only. It is called by 'tcnssqry' and retrieves single shipment data based upon a value passed from the calling form. This is done by dynamically building part of the where clause with the value.

Software Unit Name: shiptcnssqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form will be seen by shippers only. The user must enter at least 14 characters of the ten tcn field for query purposes.

Software Unit Name: shiptcnssdtl.frm
Software Unit Description: This form is a duplicate of the tcnssdtl.frm, but it is called for shippers only by the shiptcnssqry form.

Software Unit Name: ssnssqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form displays one field, SSN (social security number), which will be used as user input for single shipment queries. The user may enter either a partial value (the last four digits) or the entire value of a social security number.

Software Unit Name: ssnssdtl.frm
Software Unit Type: SQL*Form
Software Unit Description: This form is called by 'ssnssqry'. A value, SSN, is passed to this form, a where clause is built using the value, and a query is invoked to retrieve single shipment cargo data associated with the entered social security number.

Software Unit Name: povssqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form functions specifically for user input. The two values, LAST NAME and POV LICENSE PLATE NBR, entered either partially or entirely, will be used for single shipment data retrieval.

Software Unit Name: povssdf.frm
Software Unit Type: SQL*Form
Software Unit Description: This form is called by 'povssqry'. It displays personal property information associated with a particular person. The data is retrieved based upon the value passed from the calling form, which is used to build part of the where clause.

Multiple Shipment - Detail Subfunction

Figure 4.7 shows the logical flow for the software units that are components of the multiple shipment - detail subfunction. The following are the software units for this subfunction.

Software Unit Name: allmsqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to build a dynamic SQL select statement to retrieve shipment unit records regardless of the cargo type. From this form the user may enter values for building the "where" portion of a Select Statement. This "where" clause is passed to the detail display screen allmsdf.

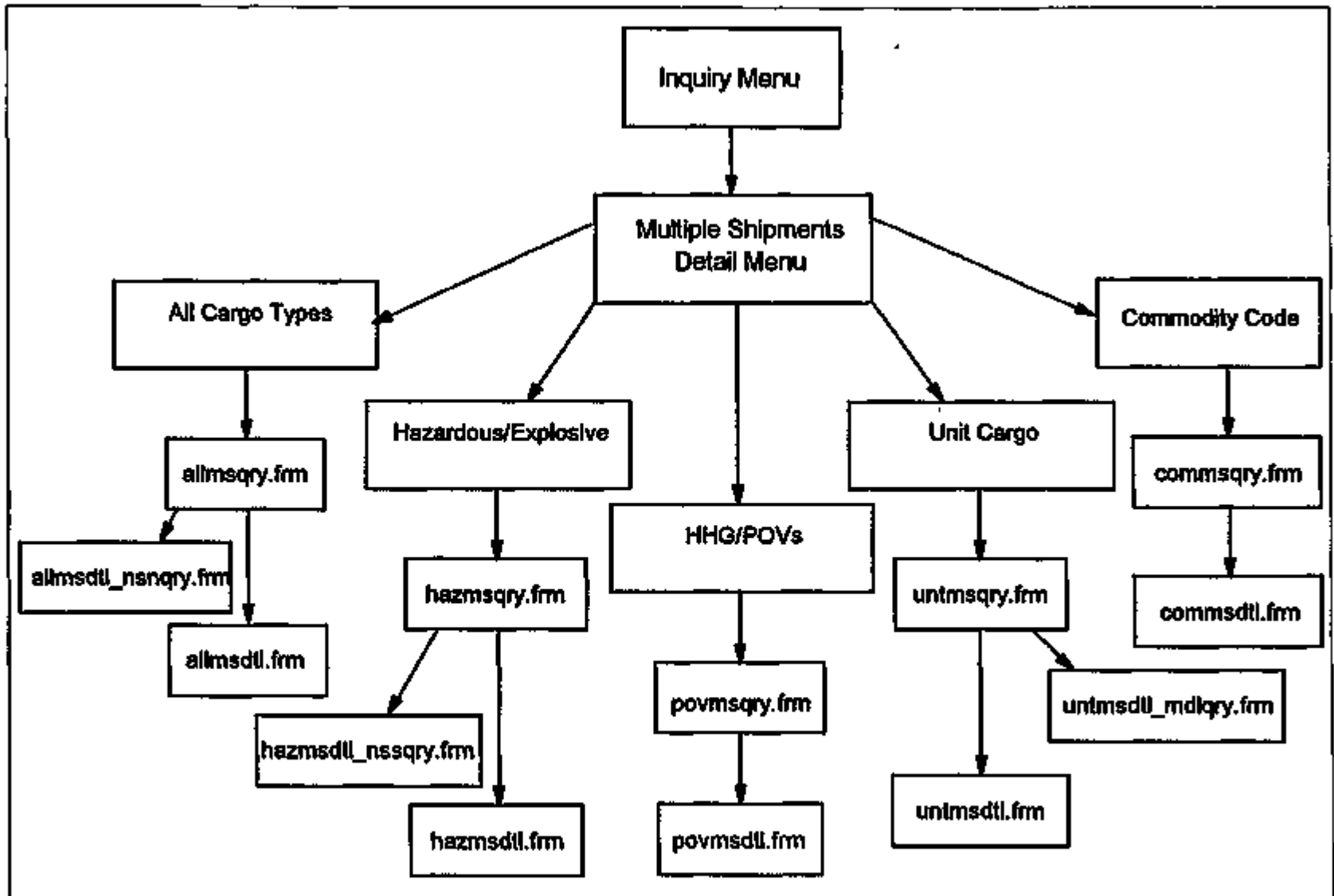


Figure 4.7. The Multiple Shipments - Detail subfunctions.

Software Unit Name: allmsdtl.frm
Software Unit Type: SQL*Form
Software Unit Description: This form displays the shipment unit record information based on the query criteria entered by the user in the allmsqry form.

Software Unit Name: allmsdtl_nsqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form is like the allmsdtl form except that it uses a different view for the base table. It is called if a user entered an nsn on the allmsqry form.

Software Unit Name: hazmsqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to build a dynamic SQL select statement to retrieve Hazardous/Explosive cargo shipment unit records. From this form the user may enter values for building the "where" portion of a Select Statement. This "where" clause is passed to the detail display screen hazmsdtl.

Software Unit Name: hazmsdtl.frm
Software Unit Type: SQL*Form
Software Unit Description: This form displays Hazardous/Explosive shipment unit information based on the query criteria entered by the user in the hazmsqry form.

Software Unit Name: hazmsdtl_nsqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form is like the hazmsdtl.frm

Software Unit Name: povmsqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to build a dynamic SQL select statement to retrieve Household Goods/Privatey Owned Vehicle cargo shipment unit records. From this form the user may enter values for building the "where" portion of a Select Statement. This "where" clause is passed to the detail display screen povmsdtl.

Software Unit Name: povmsdtl.frm
Software Unit Type: SQL*Form
Software Unit Description: This form displays Household Goods/Privatey Owned Vehicle shipment unit information based on the query criteria entered by the user in the povmsqry form.

Software Unit Name: untmsqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to build a dynamic SQL select statement to retrieve Unit Movement cargo shipment unit records. From this form the user may enter values for building the "where" portion of a Select Statement. This "where" clause is passed to the detail display screen untmsdtl.

Software Unit Name: untmsdtl.frm
Software Unit Type: SQL*Form
Software Unit Description: This form displays Unit Movement shipment unit information based on the query criteria entered by the user in the untmsqry form.

Software Unit Name: untmsdtl_mdlqry.frm
Software Unit Type: SQL*Form

Software Unit Description: This form is like the unmsdfl form except that it uses a different view for the base table. It is called if the user entered a model on the unmsqry form.

Software Unit Name: commsqry.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the user to build a dynamic SQL select statement to retrieve Commodities cargo shipment unit records. From this form the user may enter values for building the "where" portion of a Select Statement. This "where" clause is passed to the detail display screen commsdfl.

Software Unit Name: commsdfl.frm

Software Unit Type: SQL*Form

Software Unit Description: This form displays Commodity cargo shipment unit information based on the query criteria entered by the user in the commsqry form.

Multiple Shipments - Summary Subfunction

Figure 4.8 shows the logical flow for the software units that are components of the multiple shipment - summary subfunction. These units consist of forms that allow the user to enter query criteria for a report that may be sent to the screen, a printer, or a file.

Software Unit Name: allsumsel.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the user to select the sort order for the summary report.

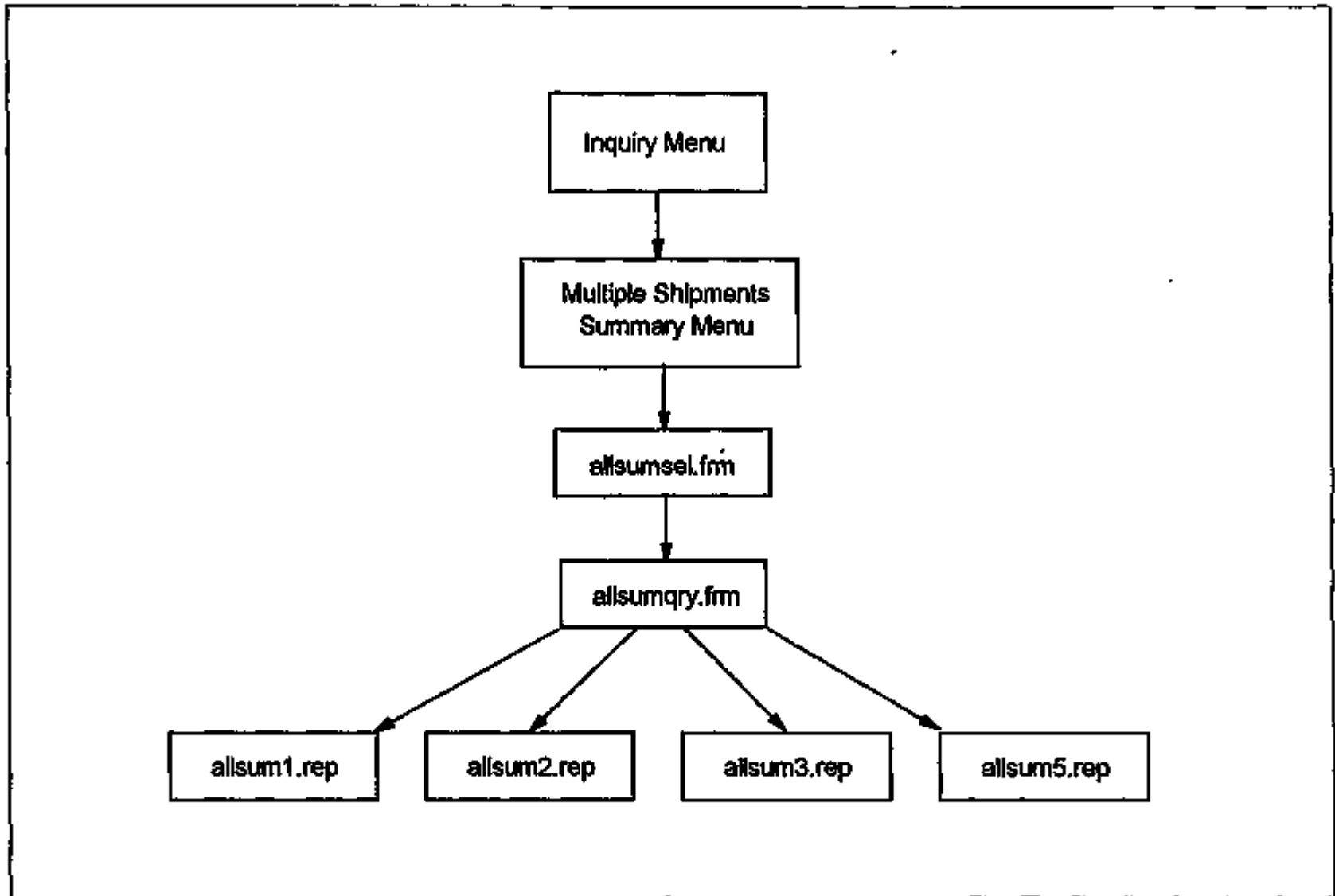


Figure 4.8. The Multiple Shipments Summary.

Software Unit Name: allsumqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to enter values for building the "where" portion of a select statement. This "where" clause is passed to the selected summary report by a SQL*Forms HOST command. It retrieves shipment unit records regardless of cargo type. These records are then used in building the selected report in SQL*ReportWriter.

Software Unit Name: allsum.rep
Software Unit Type: SQL*ReportWriter
Software Unit Description: This report provides the total number of pieces, weight, and cube of shipment units regardless of cargo type. These records are displayed by POE, POD, commodity code, RSTAT, and container number.

Software Unit Name: allsum2.rep
Software Unit Type: SQL*ReportWriter
Software Unit Description: This report provides the total number of pieces, weight, and cube of shipment units regardless of cargo type. These records are displayed by Voydoc, POE, POD, and RSTAT.

Software Unit Name: allsum3.rep
Software Unit Type: SQL*ReportWriter
Software Unit Description: This report provides the total number of pieces, weight, and cube of shipment units regardless of cargo type. These records are displayed by POE, POD, and RSTAT.

Software Unit Name: allsum5.rep
Software Unit Type: SQL*ReportWriter

Software Unit Description: This report provides the total number of pieces, weight, and cube of shipment units regardless of cargo type. These records are displayed by RSTAT, POE, and POD.

Ad Hoc Subfunction

Some ICDB users will be given the capability to form their own SQL queries using a third-party vendor product, Intelligent Query (IQ)¹. Figure 4.9 shows the logical flow for the ad hoc queries. Once a user requests to perform an Ad Hoc query, the user interface screen will call this product.

4.4.1.4 The Reports Subsystem. The Reports subsystem allows the user to see summary data. Users may view report data on the screen for those reports that fit on the screen, print the reports on a network printer, or send the report to a file. Additional functionality that allows users to send a report file to a floppy or to their PC printer will be added at a later date. Figure 4.10 shows the logical flow for the Reports subsystem.

Cargo Management and Documentation Subfunction

Figures 4.11 and 4.12 show the logical flow for the software units that are components of the Reports Cargo Management and Documentation subfunction. Reports for Export Cargo Container Tracking and Carrier Load List will be developed once IBS data is added to the Regional Database.

Software Unit Name: bkliftqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to choose whether they want to see the report for shippers or carriers and for breakbulk cargo or containers.

¹IQ is a product of Programmed Intelligence Corp., Norcross, Georgia.

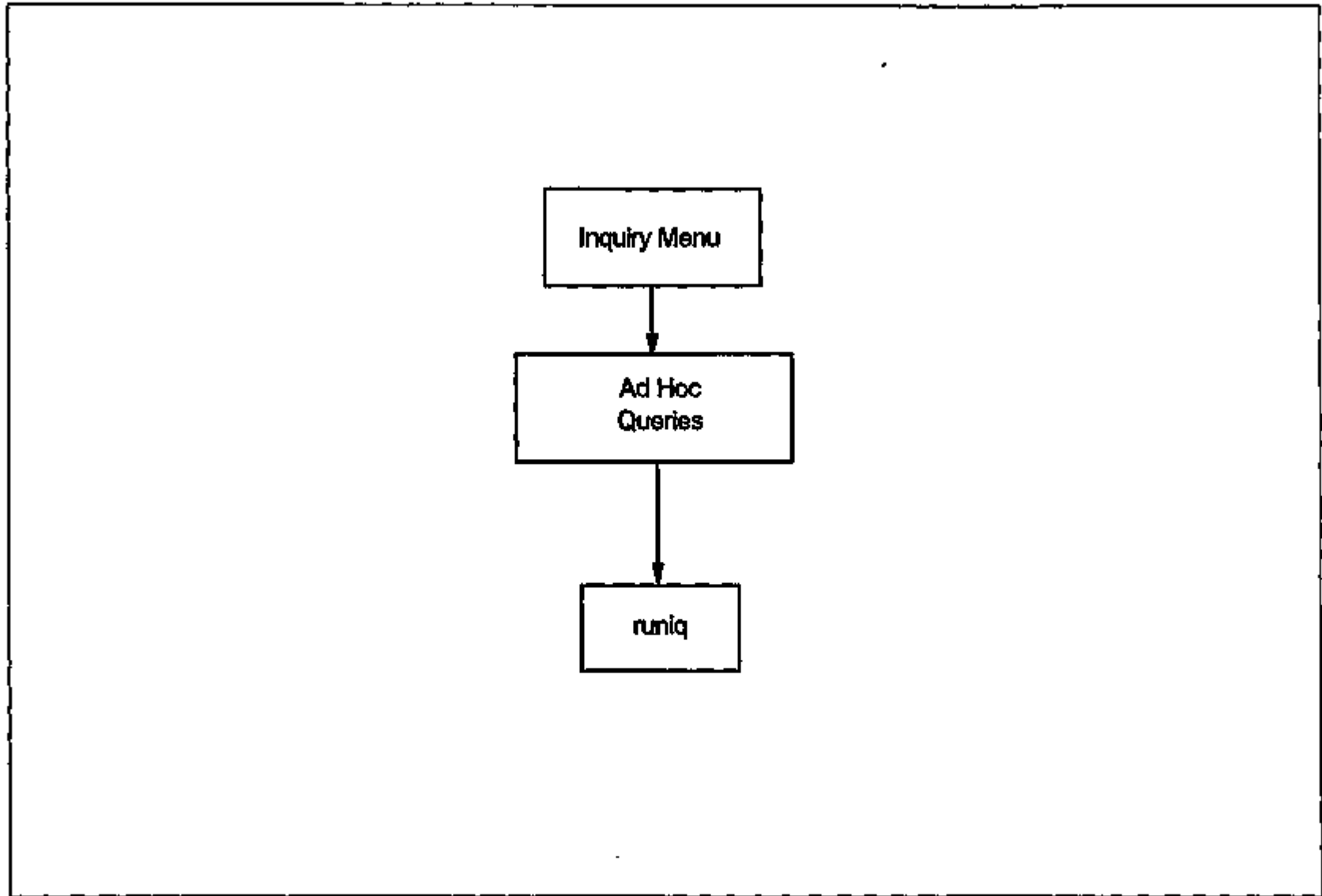


Figure 4.9. The Ad Hoc Queries subsystem.

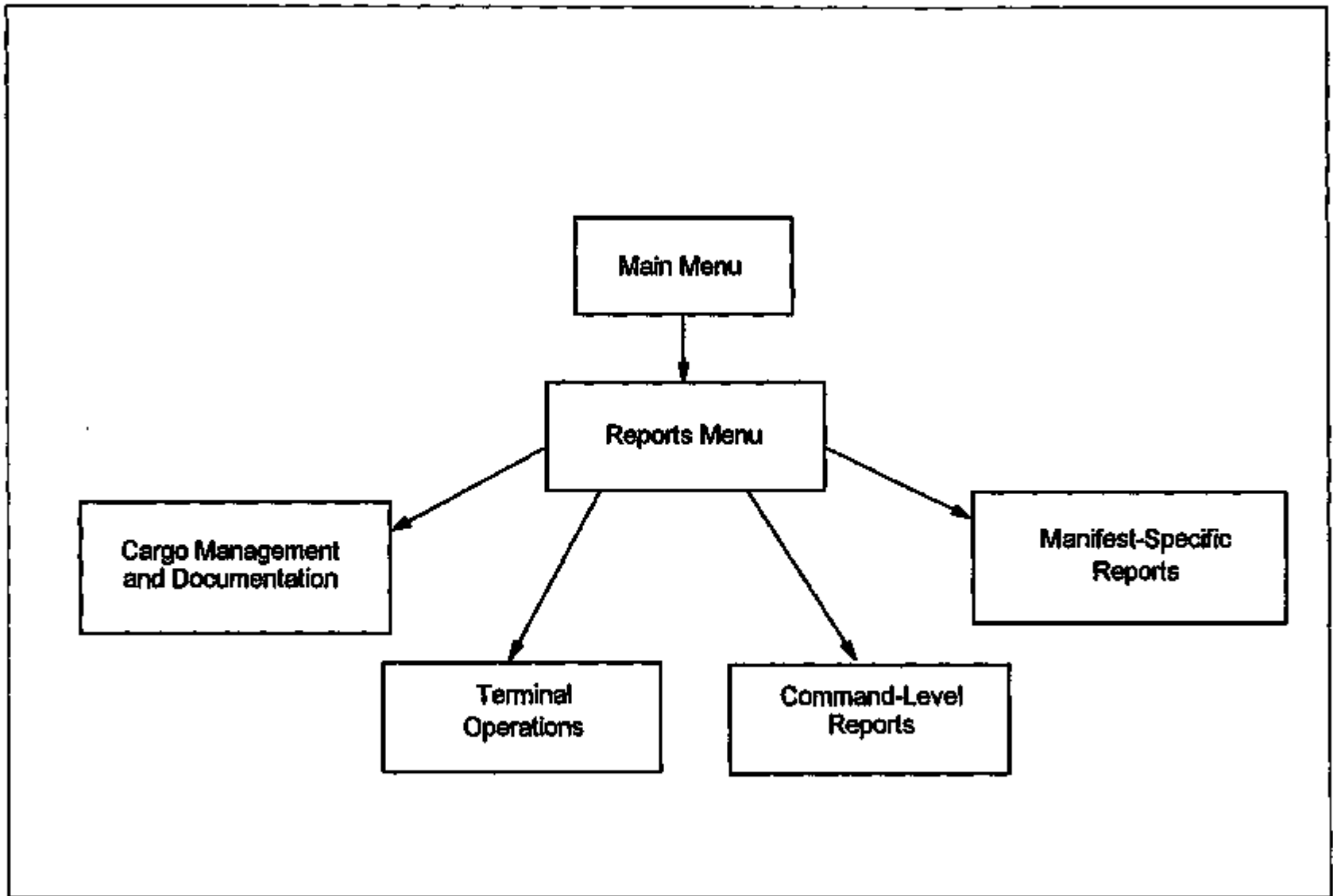


Figure 4.10. The Reports subsystem.

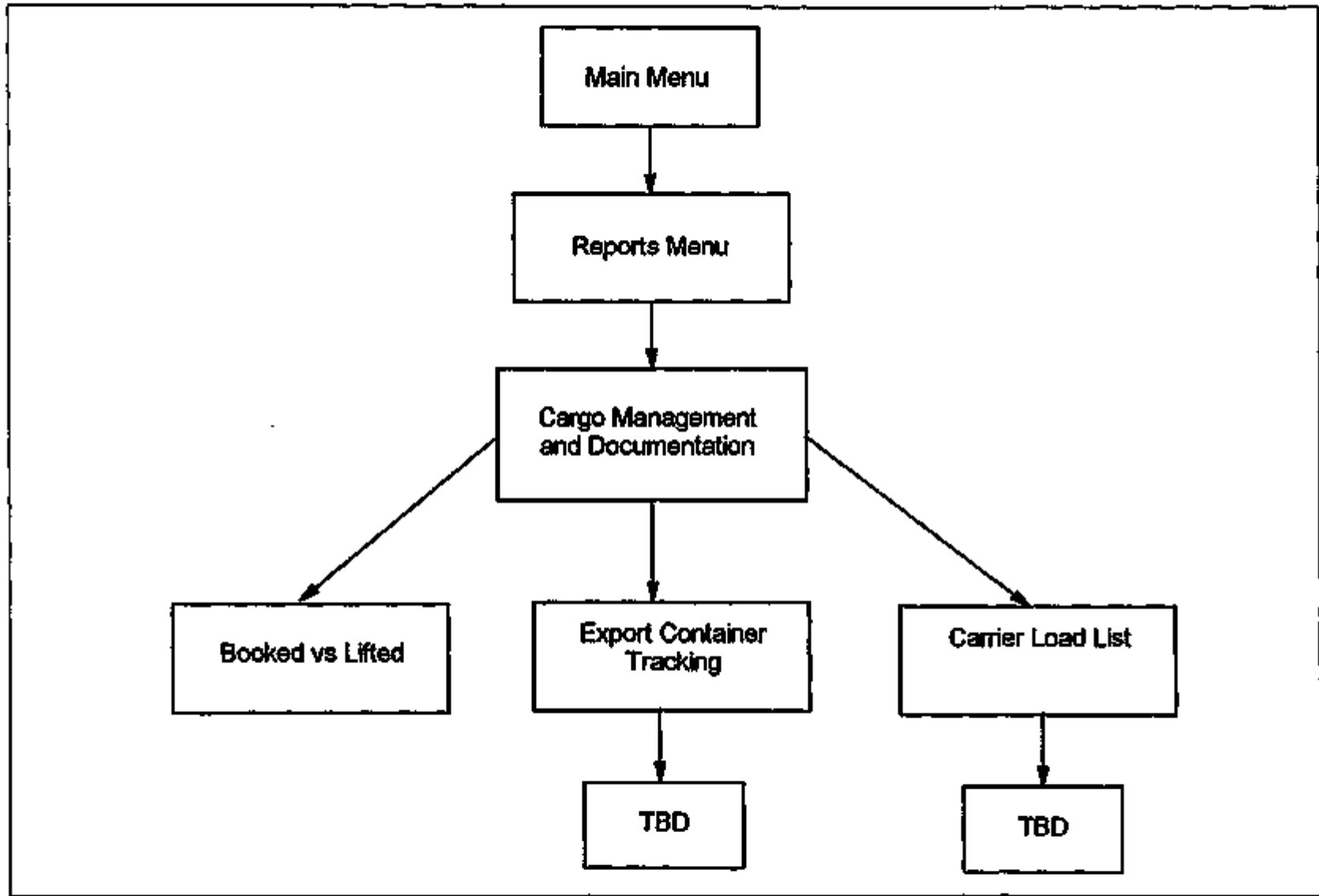


Figure 4.11. The Reports - Cargo Management and Documentation subfunction.

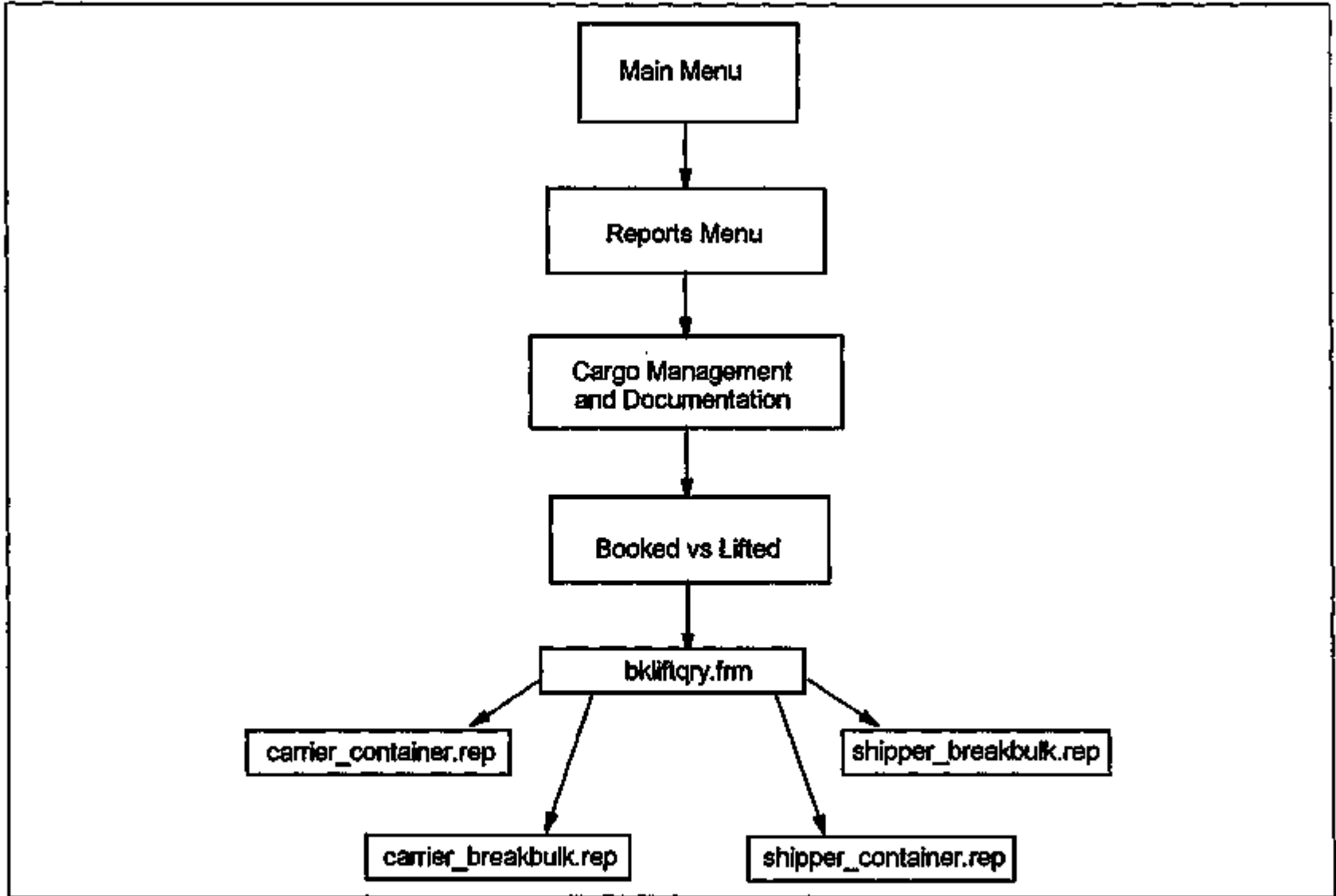


Figure 4.12. The Reports - Cargo Management and Documentation - Booked vs Lifted subfunction.

Software Unit Name: carrier_container.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows booked vs lifted shipments by carriers for container cargo.

Software Unit Name: carrier_breakbulk.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows booked vs lifted shipments by carrier for breakbulk cargo.

Software Unit Name: shipper_container.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows booked vs lifted shipments by shipper for container cargo.

Software Unit Name: shipper_breakbulk.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows booked vs lifted shipments by shipper for breakbulk cargo.

Terminal Operations Subfunction

Figures 4.13 - 4.20 show the logical flow for the software units that are components of the Reports Terminal Operations subfunction.

Software Unit Name: exportcargosumqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to limit the Export Cargo On-hand Shippable report to a region or port and to further limit the report to containerized cargo or breakbulk cargo.

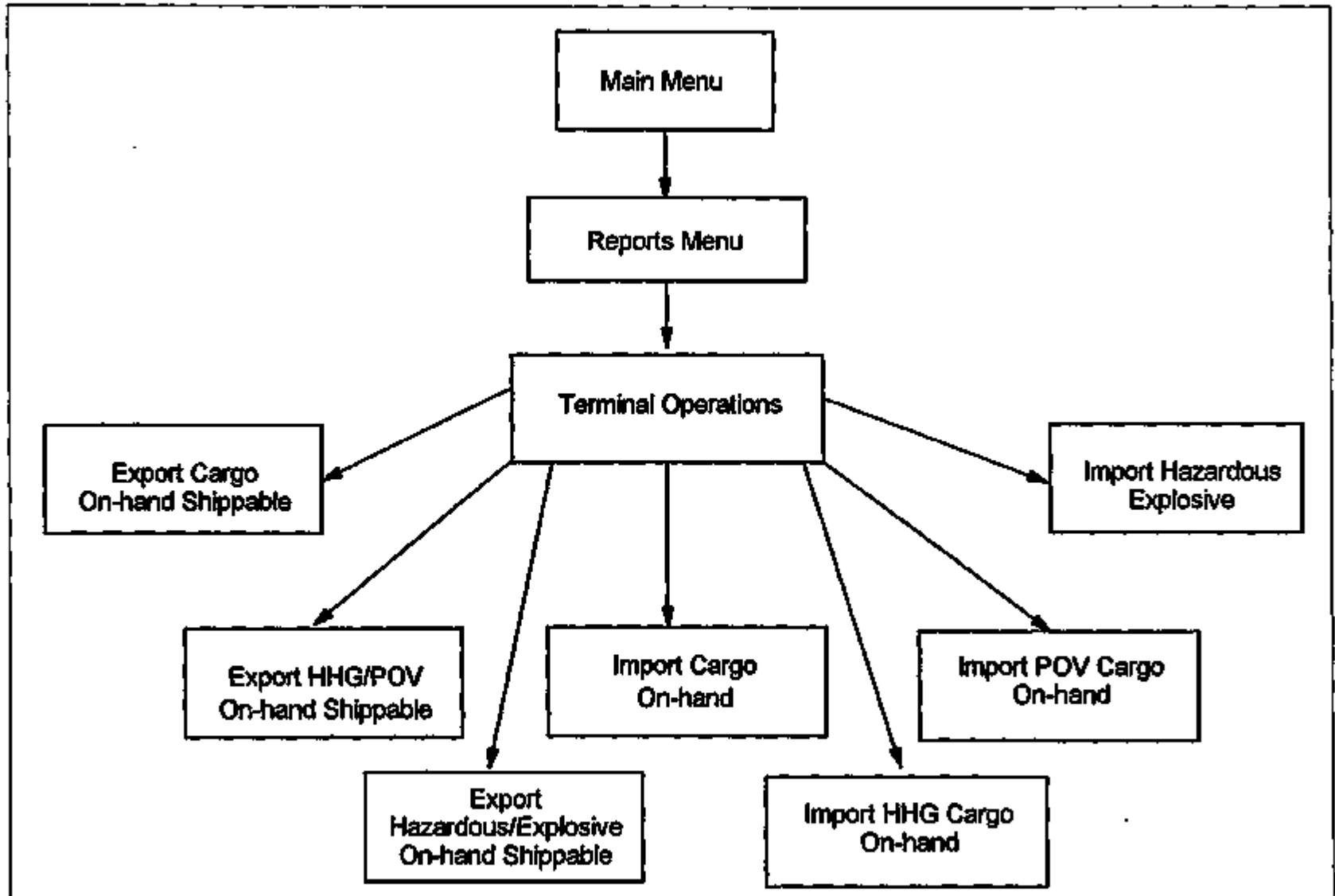


Figure 4.13. The Reports - Terminal Operations subfunction.

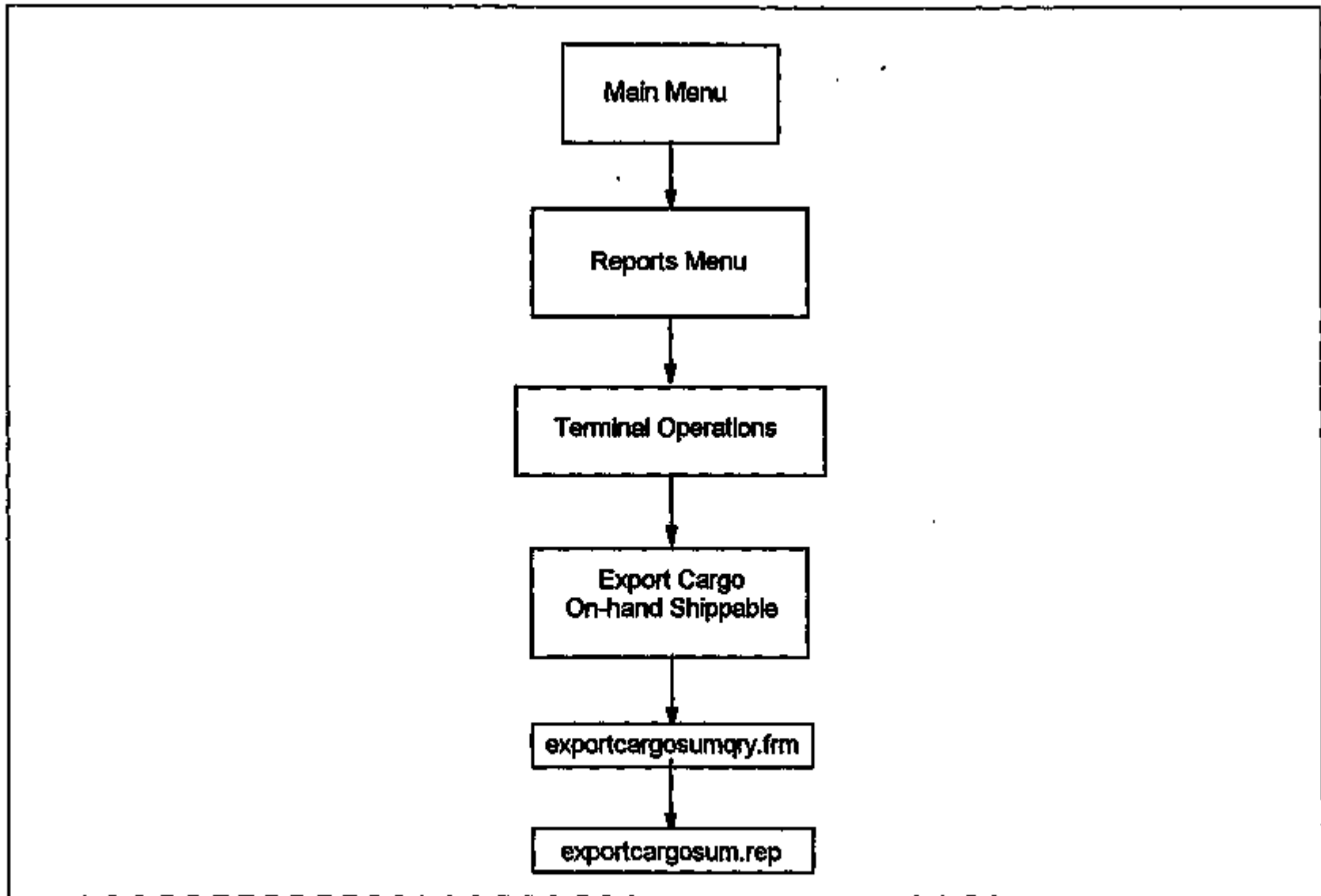


Figure 4.14. The Reports - Terminal Operations - Export Cargo On-hand Shippable subfunction.

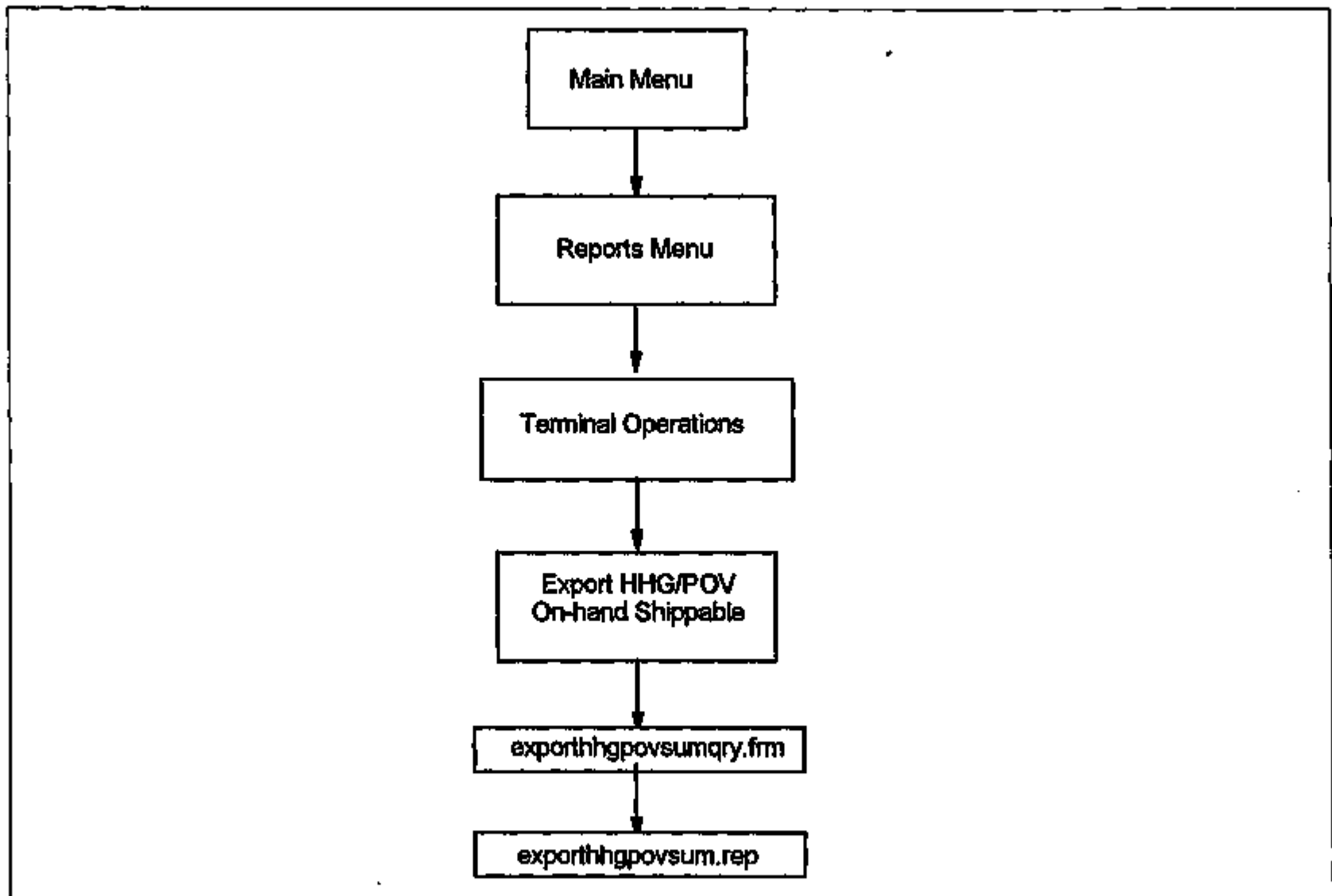


Figure 4.15. The Reports - Terminal Operations - Export HHG/POV On-hand Shippable subfunction.

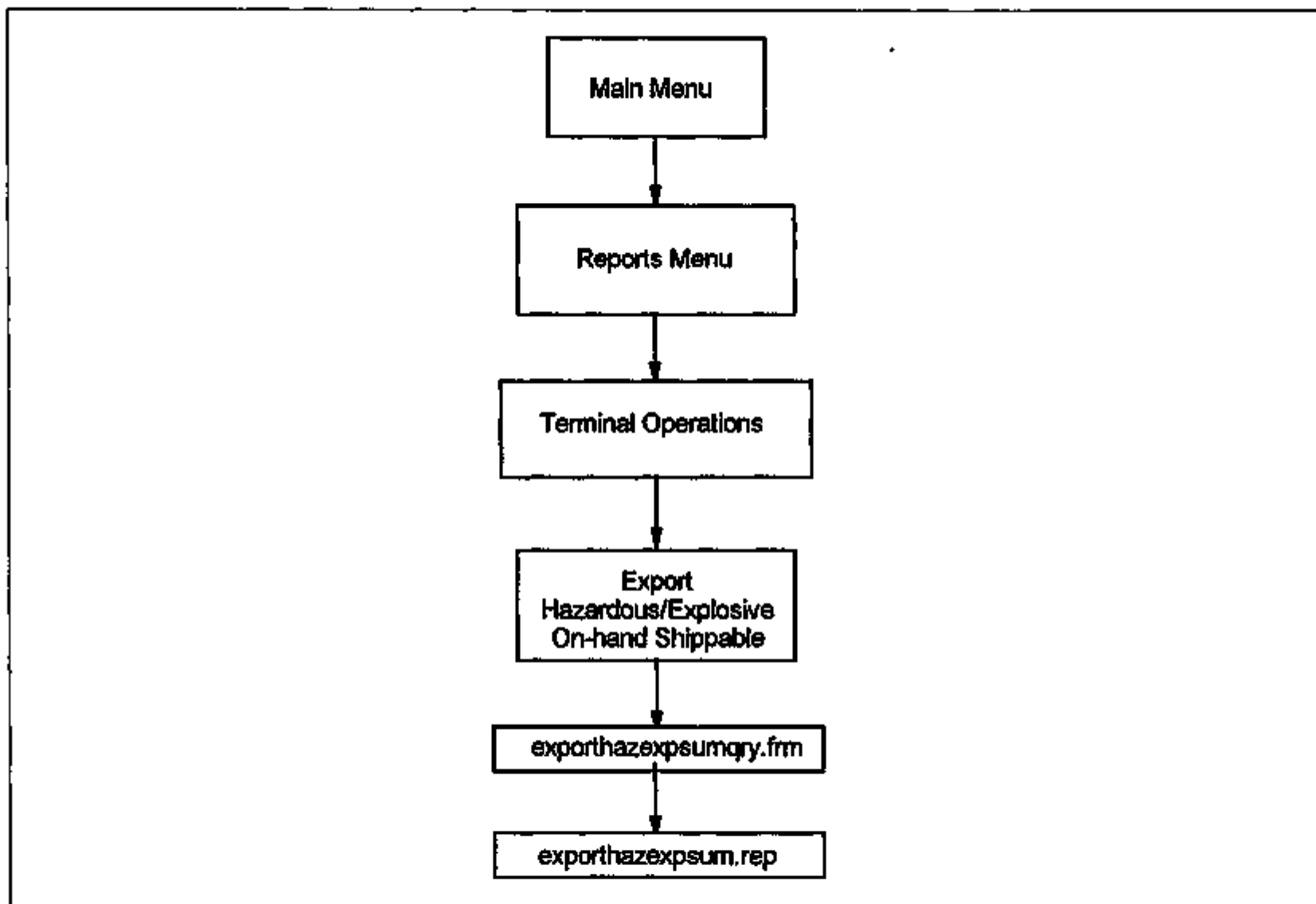


Figure 4.16. The Reports - Terminal Operations - Export Hazardous/Explosive On-hand Shippable subfunction.

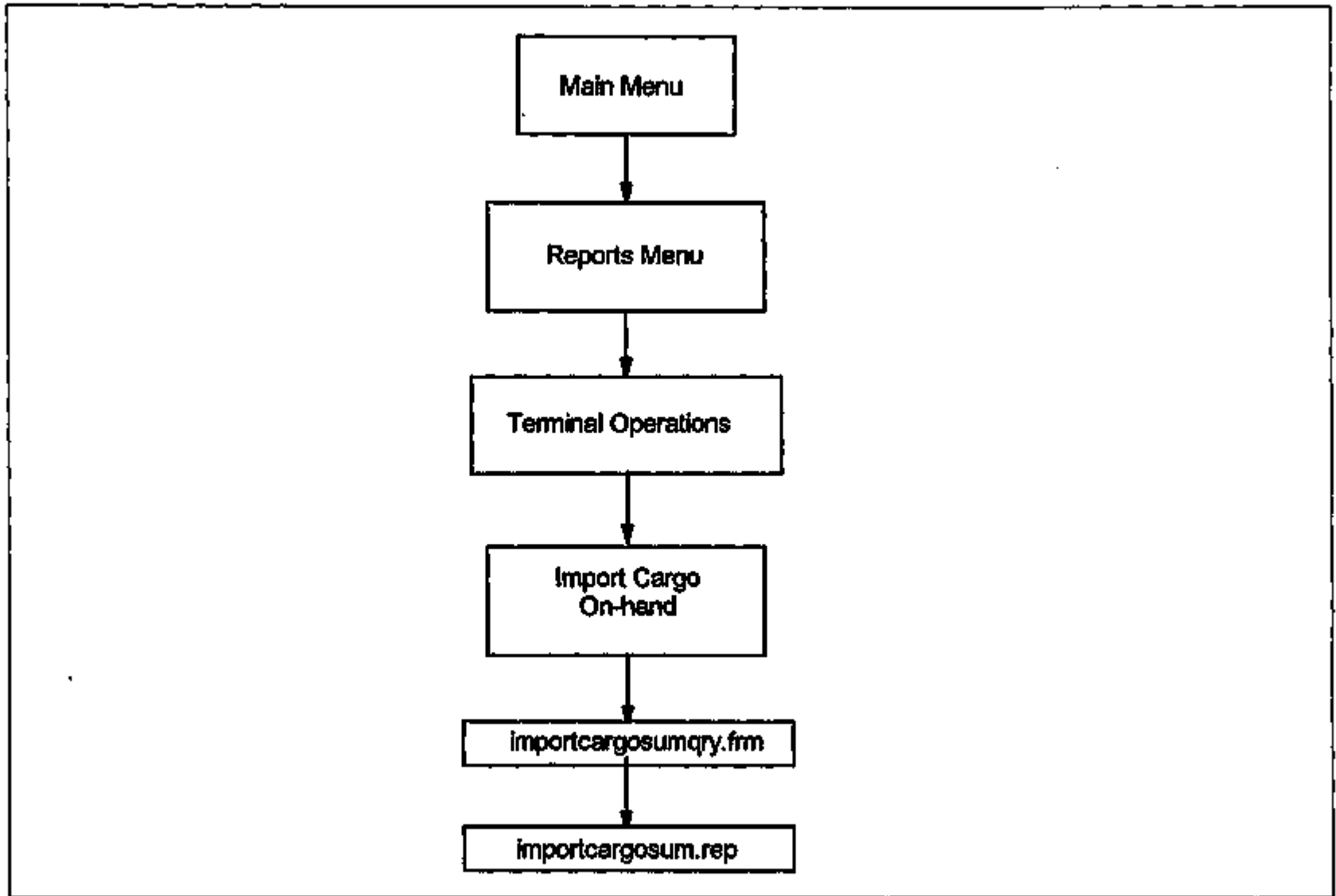


Figure 4.17. The Reports - Terminal Operations - Import Cargo On-hand subfunction.

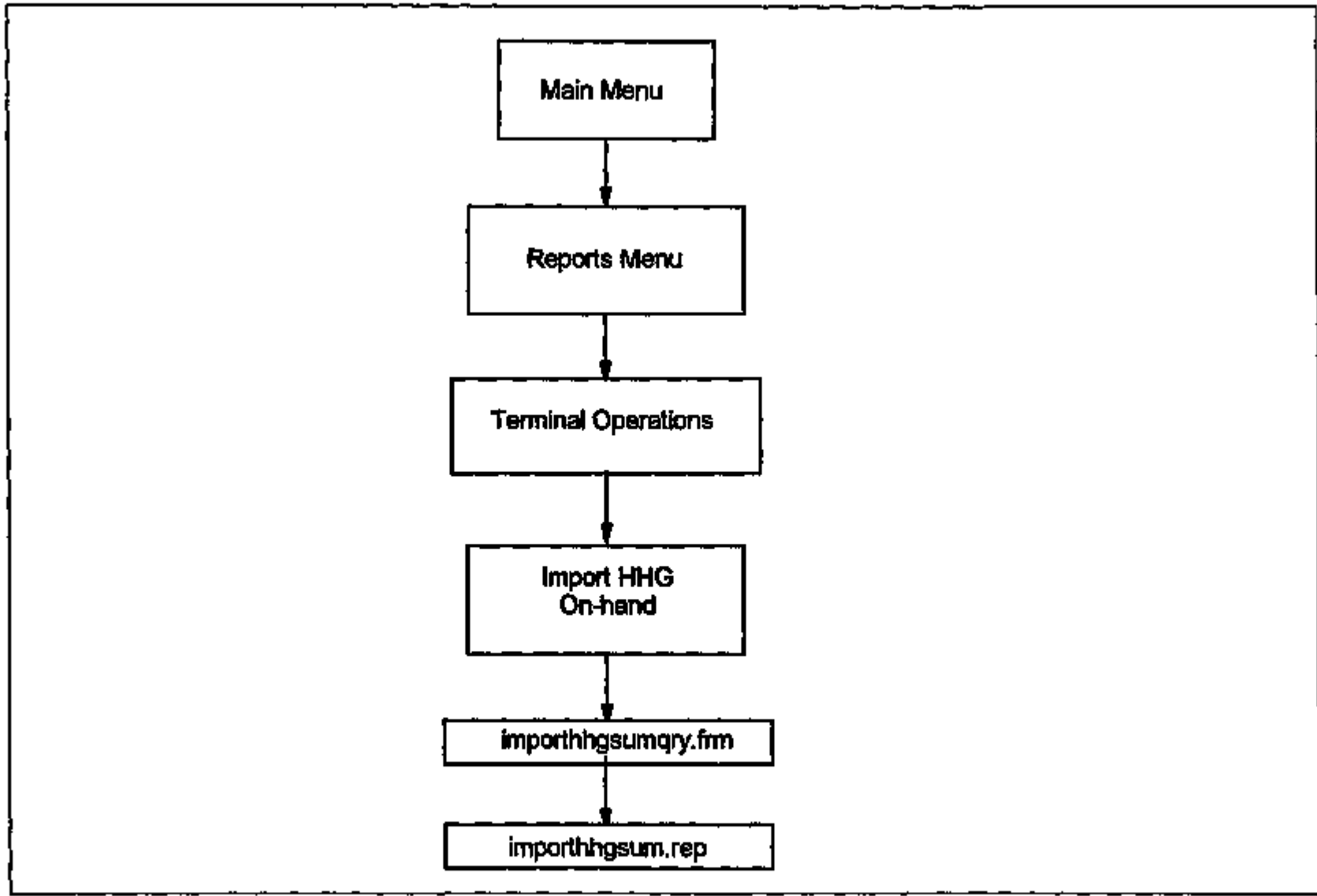


Figure 4.18. The Reports - Terminal Operations - Import HHG Cargo On-hand subfunction.

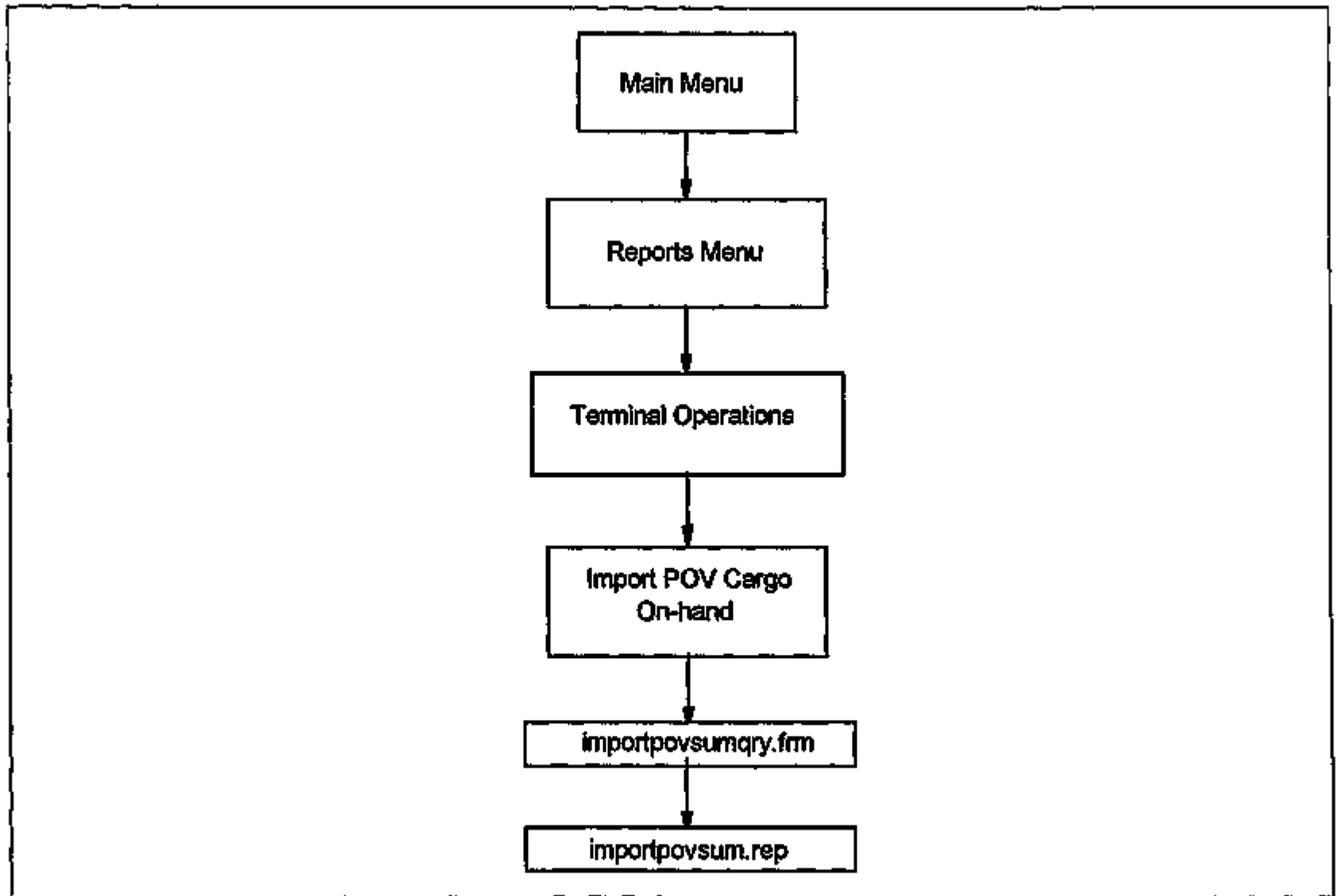


Figure 4.19. The Reports - Terminal Operations - Import POV Cargo On-hand subfunction.

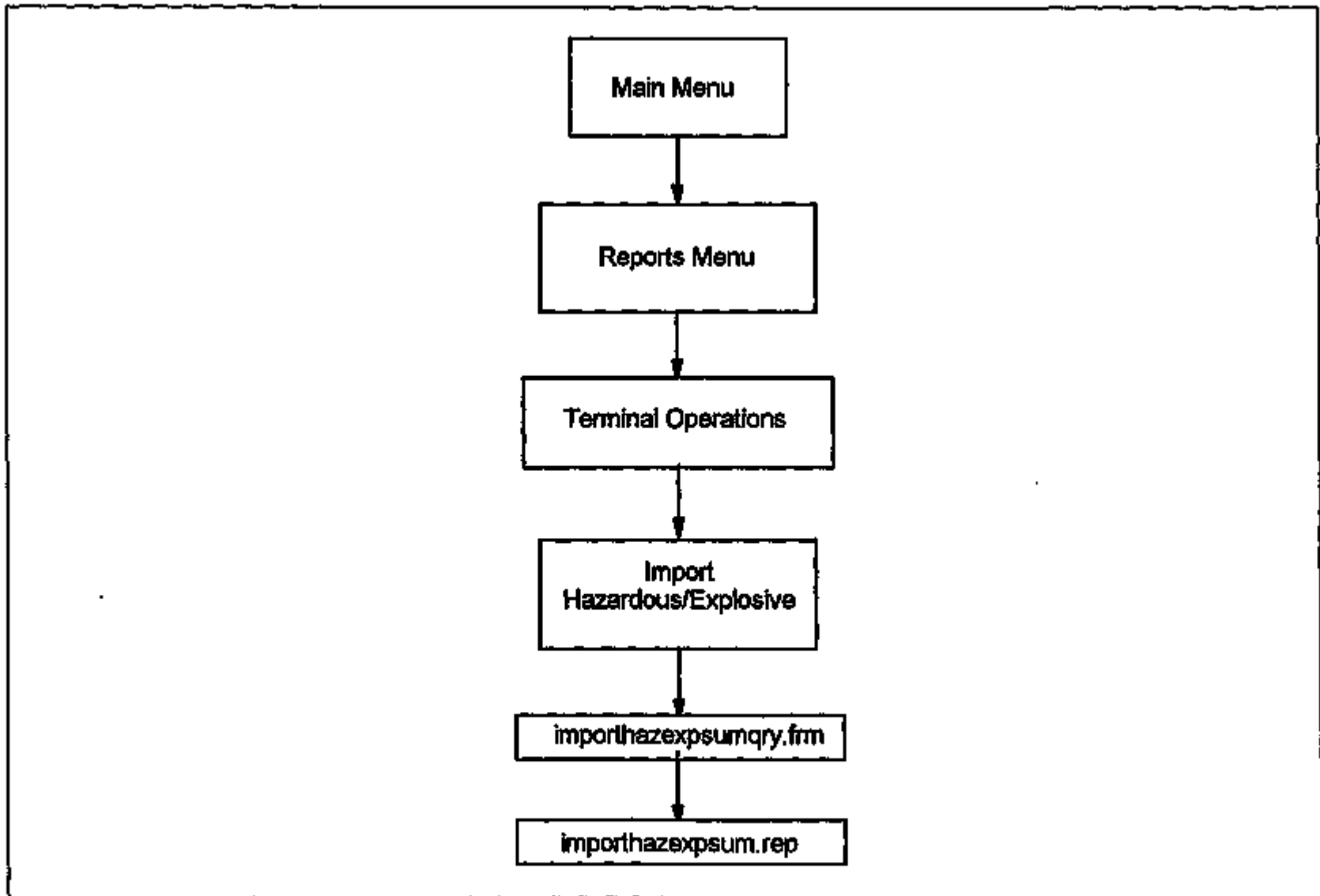


Figure 4.20. The Reports - Terminal Operations - Import Hazardous/Explosive subfunction.

Software Unit Name: exportcargosum.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows a summary of export cargo by port. It displays the number of shipments units on-hand, the number of shipments on-hand for number of days categories, and the average age for on-hand shipments.

Software Unit Name: exportihgpovsumqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to limit the Export HHG/POV On-hand Shippable report to a specific region or port.

Software Unit Name: exporthhgovsum.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows a summary of HHG/POV export cargo by port. It displays the number of shipment units on-hand, the number of shipment units on-hand for number of days categories, and the average age for on-hand shipments.

Software Unit Name: exporthazexpsumqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to limit the Export Hazardous/Explosive On-hand Shippable report to a specific region or port.

Software Unit Name: exporthazexpsumqry.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows a summary of hazardous/explosive export cargo by port. It displays the number of shipment units on-hand, the number of shipment units on-hand for number of days categories, and the average age for on-hand shipments.

Software Unit Name: imporccargosumqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to limit the Import Cargo On-hand report to a specific region or port and to further limit the report to container cargo or breakbulk cargo.

Software Unit Name: importcargosum.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows a summary of import cargo by port. It displays the number of shipment units on-hand, the number of shipment units on-hand for number-of-days categories, and the average age for on-hand shipments.

Software Unit Name: importhhgsumqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to limit the Import HHG Cargo On-hand report to a specific region or port.

Software Unit Name: importhhgsum.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows a summary of import HHG cargo by port. It displays the number of shipment units on-hand, the number of shipment units on-hand for number-of-days categories, and the average age for the on-hand shipments.

Software Unit Name: importpovsumqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to limit the Import POV Cargo On-hand report to a specific region or port.

Software Unit Name: importpovsum.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows a summary of import HHG cargo by port. It displays the number of shipment units on-hand, the number of shipment units on-hand for number-of-days categories, and the average age for the on-hand shipments.

Software Unit Name: importhazexpsumqry.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to limit the Import Hazardous/Explosive Cargo On-hand report to a specific region or port.

Software Unit Name: importhazexpsum.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows a summary of import Hazardous/Explosive cargo by port. It displays the number of shipment units on-hand, the number of shipment units on-hand for number-of-days categories, and the average age for the on-hand shipments.

Command-Level Reports Subfunction

Figure 4.21 shows the logical flow for software units that are components of the Command-Level reports subfunction. The POV/RDD Performance report will be added after system implementation.

Software Unit Name: prtretcmd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to choose among the three reports that are listed for Command-Level reports. Once the user has

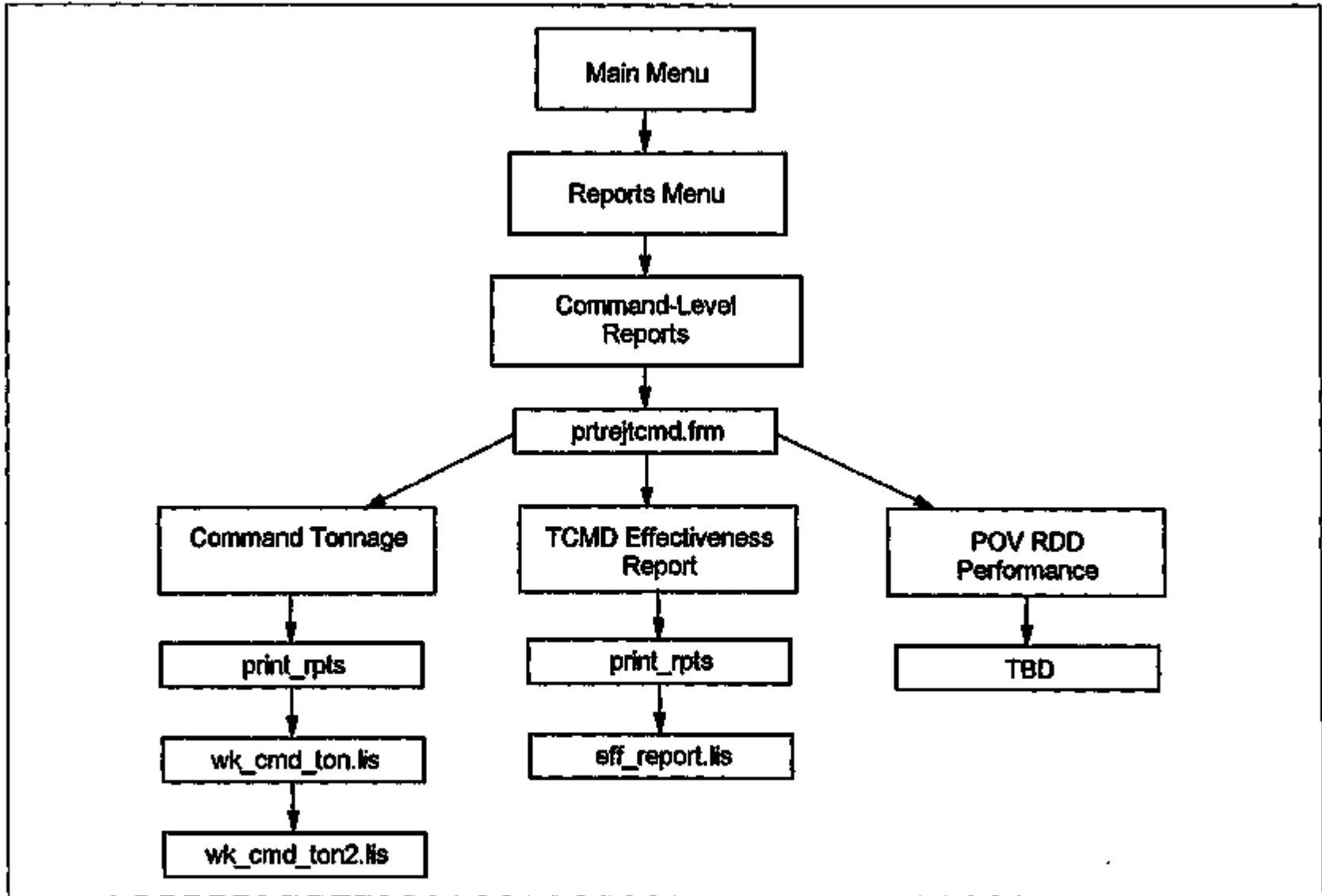


Figure 4.21. The Command-Level subfunction.

chosen a report, a pop-up window ask the user to select a printer and/or select a file to be printed.

Software Unit Name: wk_cmd_ton.lis
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows detailed summaries of shipment units received, consolidated, and shipped by POE. The data is sorted by range for each POE. It is produced weekly.

Software Unit Name: wk_cmd_ton2.lis
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows a higher-level summary of shipment units received, consolidated, and shipped by range. It is produced weekly.

Software Unit Name: print_rpts
Software Unit Type: Unix script file
Software Unit Description: This file is called by all three of the Command-level reports. It accepts the report, printer, and for the TCMD Effectiveness Report the date file. It is also used to check the sid of the user to point to the file in the correct directory path (e.g., PROD or TEST).

Software Unit Name: eff_report.lis
Software Unit Type: SQL*Plus
Software Unit Description: This report shows a listing of TCMDs that were identified as having one of 39 types of errors. These TCMDs did not have rejectable errors. Because this is a daily report, the user may choose to print any report produced during a seven-day period.

Software Unit Name: atcmdeff.sql
Software Unit Type: PL/SQL script
Software Unit Description: This module checks ATCMD records submitted by shippers for the 39 errors specified by MILSTAMP, indicating the errors on the ATCMD EFFECTIVENESS table.

Software Unit Name: error1.sql
Software Unit Type: PL/SQL script
Software Unit Description: This module builds prime records on the ATCMD EFFECTIVENESS table for records indicated as "No Advance Supplied."

Manifest-Specific Reports Subfunction

Figure 4.22 shows the logical flow for the software units that are components of the Reports Manifest Specific Reports subfunction. More information about the software units for the forms for Print Manifest, Manifest Distribution Eastern Area, and Manifest Distribution Western Area must be supplied by developers of the WPS terminal-level system.

Software Unit Name: print_rpts
Software Unit Type: Unix script file
Software Unit Description: This file is called by all three of the Manifest-Specific reports. It accepts the report, printer, and for the Manifest Recap by POE Report the date file. It is also used to check the sid of the user to point to the file in the correct directory path (e.g., PROD or TEST).

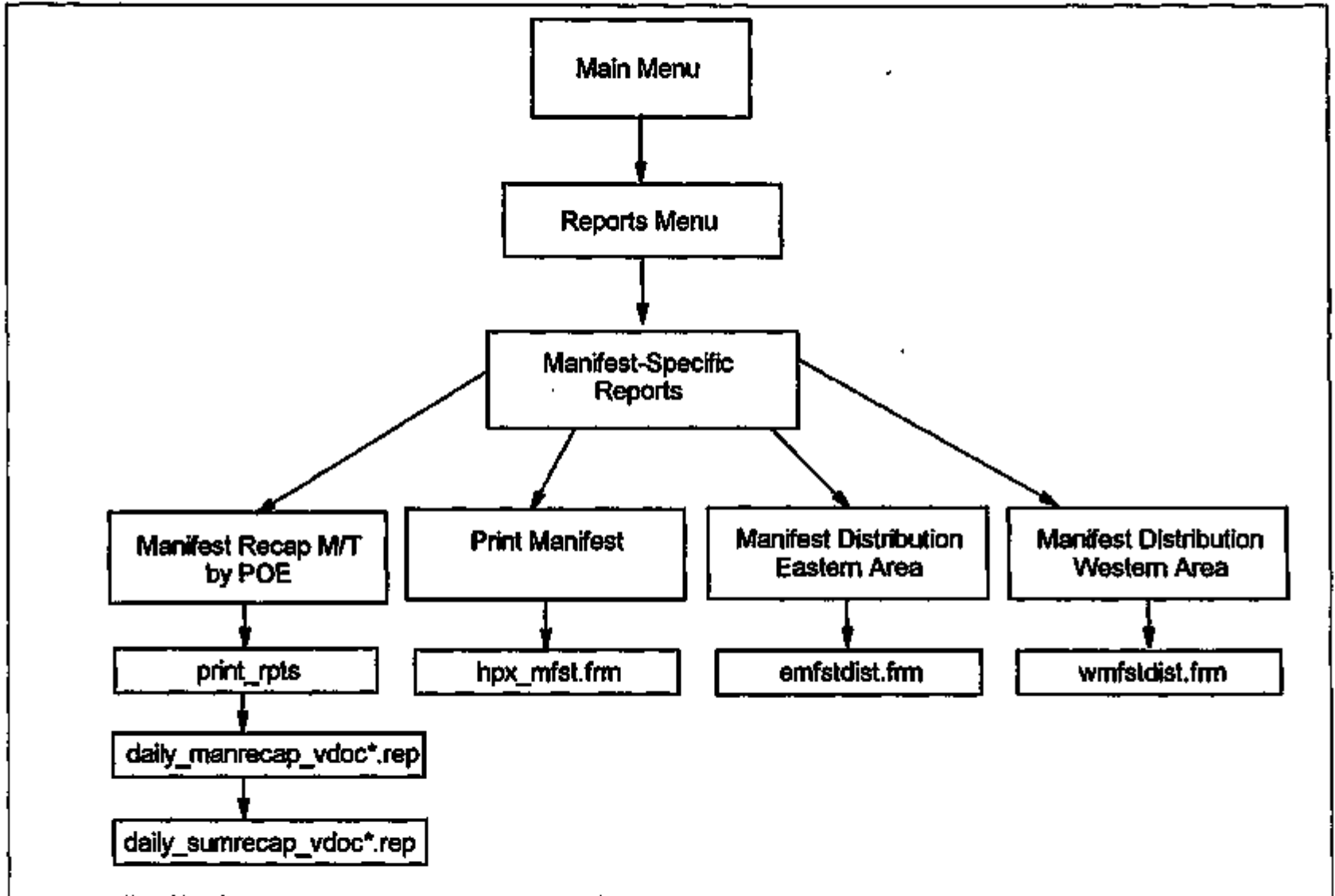


Figure 4.22. The Reports - Manifest-Specific Reports subfunction.

Software Unit Name: mfst_recap_byoe
Software Unit Type: Stored procedure
Software Unit Description: This stored procedure is executed daily from the con utility. It stores CONUS export manifest data into two tables, DAILY_MANIFEST_RECAP and DAILY_MANIFEST_SUMMARY. This data will then be utilized by Reportwriter to generate six reports (daily_manrecap_vdocA.rep; daily_manrecap_vdocG.rep; daily_manrecap_vdocP.rep; daily_sumrecap_vdocA.rep; daily_sumrecap_vdocG.rep; and daily_sumrecap_vdocP.rep) based upon the first letter of the CONUS VOYDOC for all manifests entered during a 24-hour period.

Software Unit Name: daily_manrecap_vdoc*.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report displays a detailed report of manifest cargo. It shows the number of containers, the weight of the containerized cargo, the number of break bulk cargo and the weight of the break bulk cargo.

Software Unit Name: daily_sumrecap_vdoc*.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report displays a summary report of all cargo for ports with the same first two characters of the port-id. For containers and break bulk cargo, the report displays the number and M/TON amount of basic and supplement cargo.

4.4.1.5 The Regional Processes Subsystem. Figure 4.23 shows the logical flow for the Regional Processes subsystem. This subsystem will allow data administrators and functional ICDB users to monitor, edit, and/or delete data in the ICDB database. Most users of ICDB will not see this subsystem.

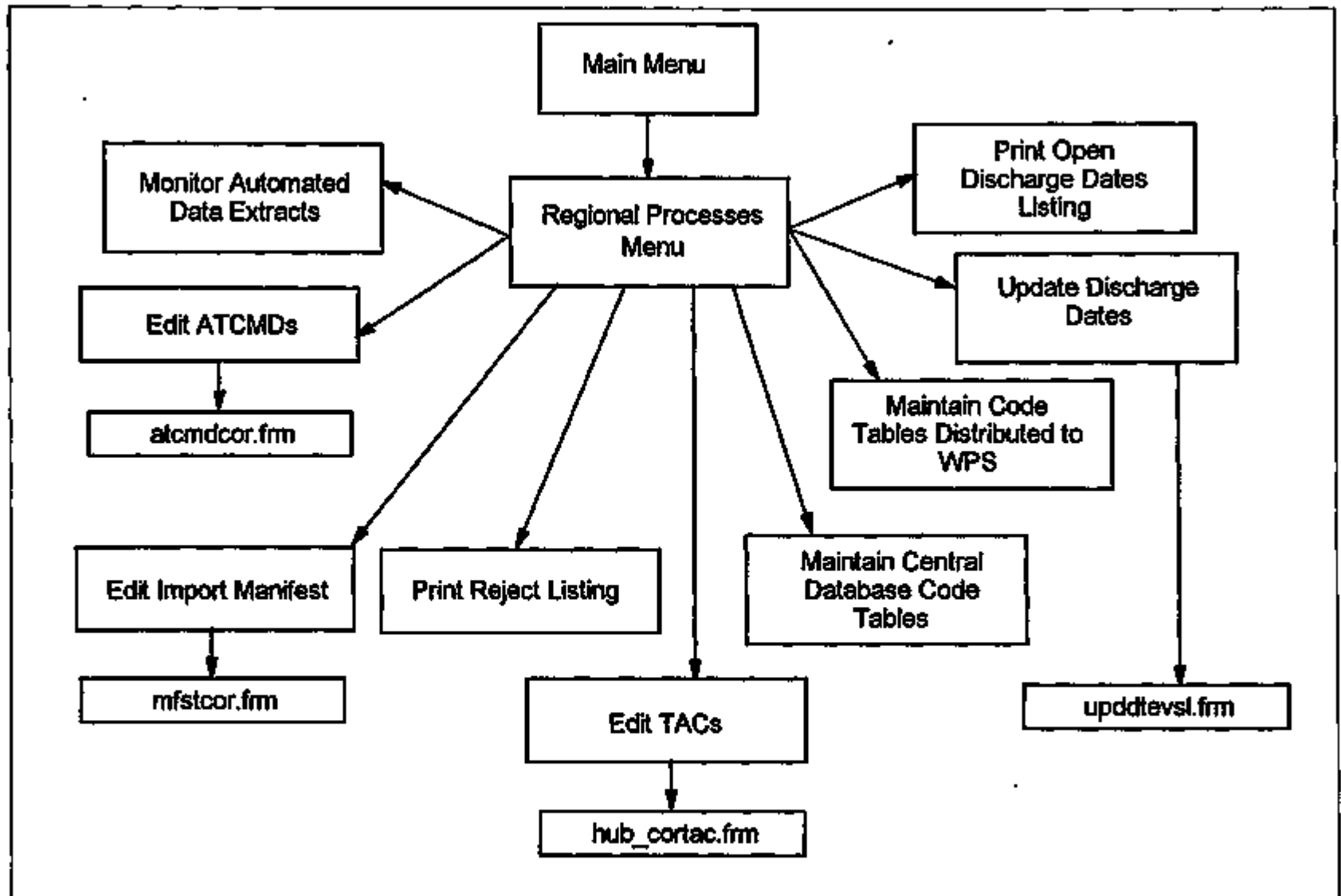


Figure 4.23. The Regional Processes subsystem.

Edit ATCMDs Subfunction

This subfunction allows data administrators or functional ICDB users to edit or delete ATCMDs that were submitted to ICDB but contained errors. The following are the software components for this subfunction.

Software Unit Name: atcmdcor.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to correct TCN/POE/PKG in rejected ATCMDs and mark them for reinput.

Software Unit Name: ireinput.sql
Software Unit Type: SQL script
Software Unit Description: This module removes records marked by atcmdcor.frm from the ATCMD reject table and reinputs them to the ATCMD process.

Edit Import Manifests Subfunction

This subfunction allows data administrators or functional ICDB users to edit or delete import manifests that were submitted to ICDB but contained errors. The following software units are components of this subfunction.

Software Unit Name: infstcor.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to correct TCN/POD/PKG in rejected manifests and mark them for reinput.

Software Unit Name: ibldrej.sql
Software Unit Type: PL/SQL script
Software Unit Description: This module moves reject manifest records to the reject table for possible correction.

Software Unit Name: hub_cortac.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to correct invalid TCN/POE/PKG in invalid TACs and mark them for reinput.

Print Reject Listing Subfunction

This subfunction allows data administrators or functional ICDB users to print a list of rejected TCMDs, rejected import manifests, or invalid TACs. Figure 4.24 shows the logical flow for software units that are components of this subfunction.

Software Unit Name: prtrejssel.frm
Software Unit Type: SQL*Form
Software Unit Description: This form is a selection form that allows the user to choose which reject listing to print: TCMDs, manifests, or TACs.

Software Unit Name: print_rpts
Software Unit Type: Unix Script File
Software Unit Description: This script accepts the report and printer. It is also used to check the sid (i.e., database thread) of the user to point to the correct directory path (e.g., PROD or TEST).

Software Unit Name: arej_report.lis
Software Unit Type: Unix File
Software Unit Description: This is the file that contains the records with rejectable TCMDs. It is created from a SQL*Plus script that is connected to the cron.

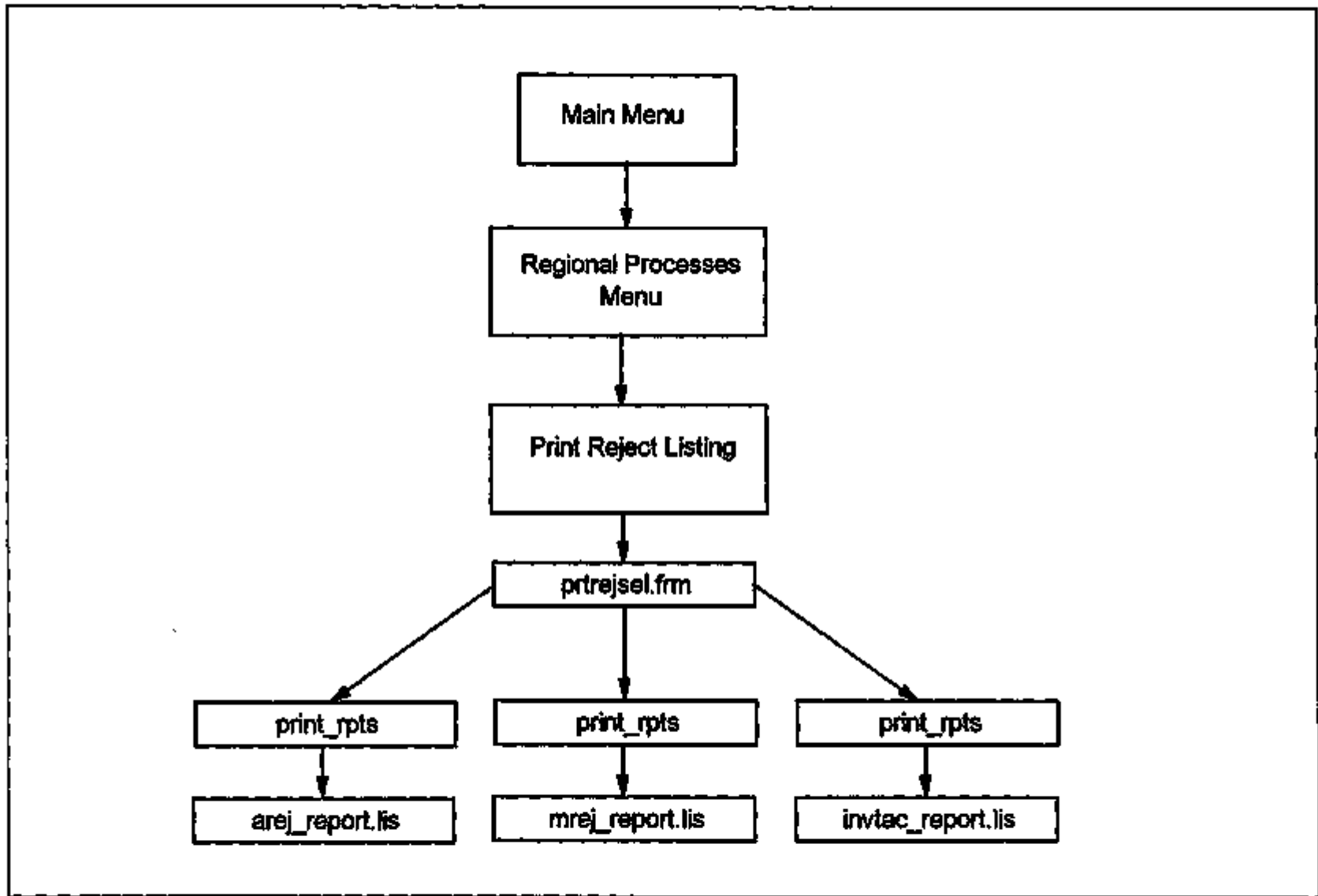


Figure 4.24. The Print Reject Listing subfunction.

Software Unit Name: mrej_report.lis
Software Unit Type: Unix File
Software Unit Description: This is the file that contains the records with rejectable manifests. It is created from a SQL*Plus script that is connected to the cron.

Software Unit Name: invtac_report.lis
Software Unit Type: Unix File
Software Unit Description: This is the file that contains the records with invalid TACs. It is created from a SQL*Plus script that is connected to the cron.

Software Unit Name: mreject.sql
Software Unit Type: SQL*PLUS script
Software Unit Description: This module prepares a report of rejected manifests records.

Software Unit Name: areject.sql
Software Unit Type: SQL*Plus script
Software Unit Description: This script prepares a report of rejected ATCMD records.

Software Unit Name: val_tac.sql
Software Unit Type: SQL*Plus script
Software Unit Description: This script prepares a report of invalid TAC records.

Maintain Central Database Code Tables Subfunction

This subfunction allows data administrators or functional users to insert, update, or delete records in the central database code tables. The changes made to these code tables will be distributed to the Hubs but will not be distributed to WPS sites. Figure 4.25 shows the logical flow for the software units that are components of this subfunction. The following software units are components of this subfunction.

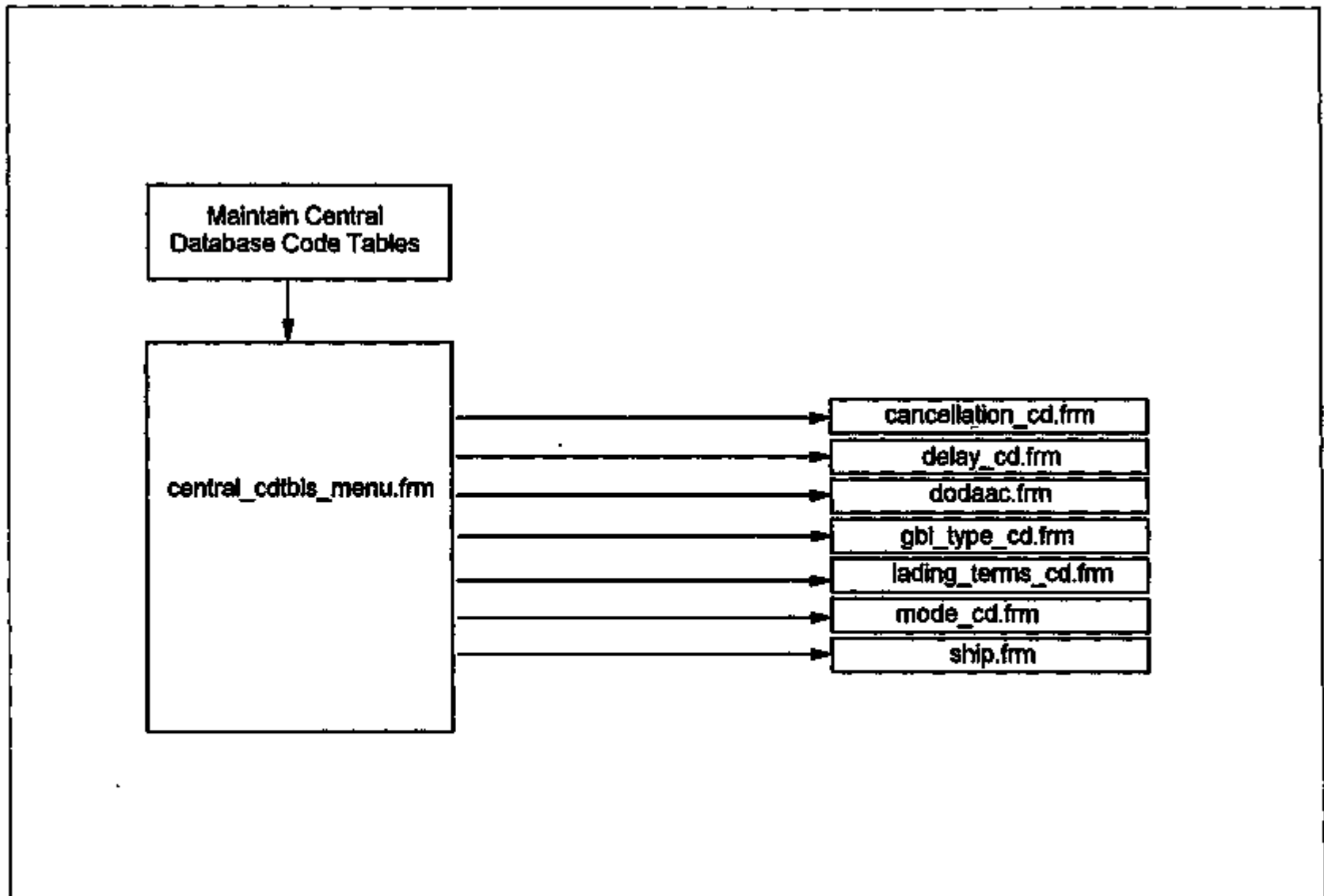


Figure 4.25. The Maintain Central Database Code Tables subfunction.

Software Unit Name: central_cdthis_menu.frm
Software Unit Type: SQL*Form
Software Unit Description: This form acts as a calling form for the code tables associated with ICDB. The Data Administrator will be able to navigate to the desired table name, press EXECUTE QUERY and call a form representing the selected code table.

Software Unit Name: cancellation_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the CANCELLATION CODE table.

Software Unit Name: delay_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records or columns for the DELAY CODE table.

Software Unit Name: dodacc.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the dodacc table.

Software Unit Name: gbl_type_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the GOVERNMENT BILL OF LADING TYPE CODE table.

Software Unit Name: lading_terms_cd.frm
Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the LADING TERMS CODE table.

Software Unit Name: mode_cd.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the MODE CODE table.

Software Unit Name: ship.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the ship table.

Maintain Code Tables Distributed to WPS

This subfunction allows ICDB data administrators or functional users to insert, update, or delete records in the ICDB code tables. Changes to these code tables will be distributed to WPS sites. Figure 4.26 shows the logical flow for the software units that are components of this subfunction. The following are the software units that are components of this subfunction.

Software Unit Name: commodity_cd.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the COMMODITY CODE table. A Unix script is then called which distributes the changes to the WPS sites.

Software Unit Name: commodity_cd_grp1.frm

Software Unit Type: SQL*Form

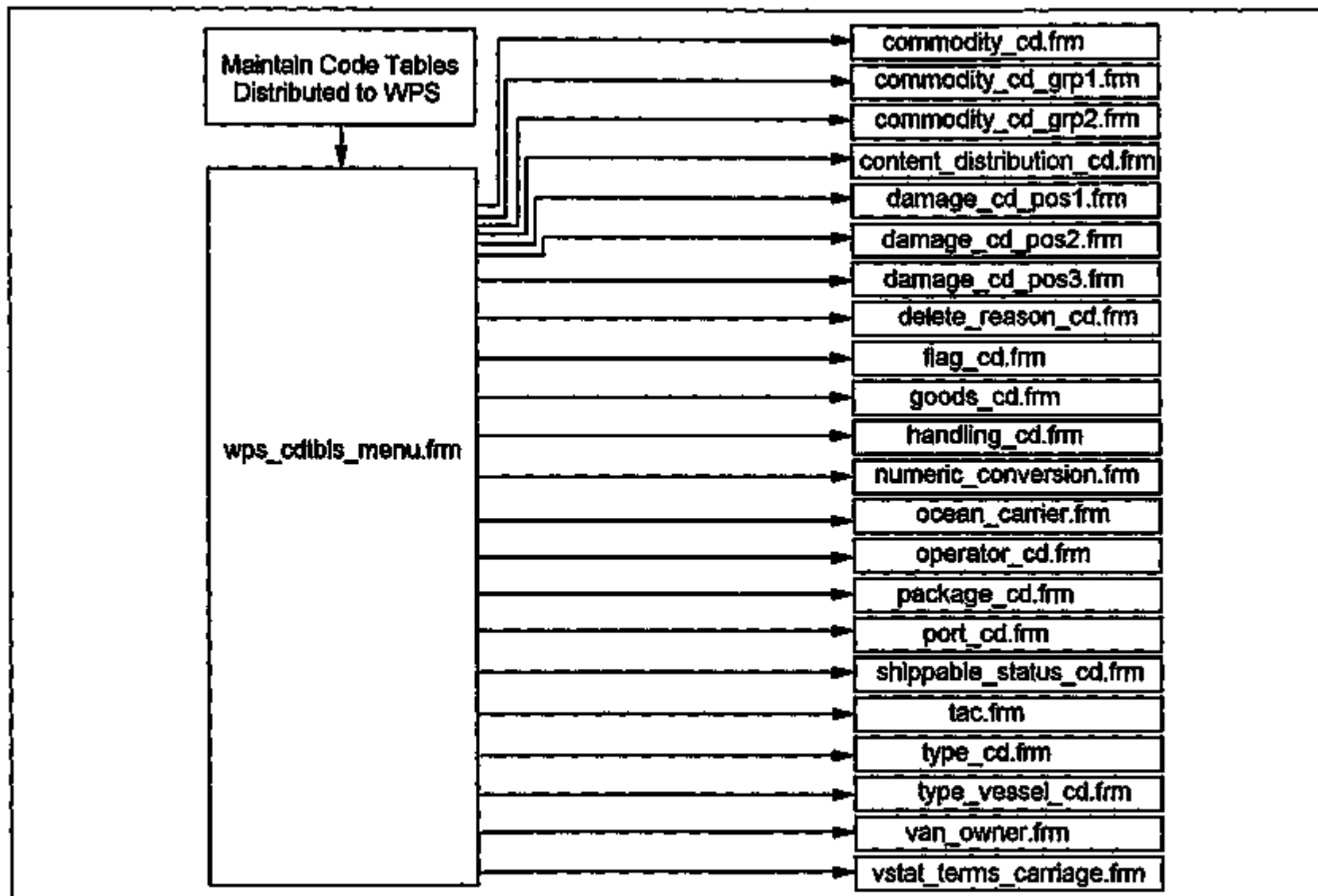


Figure 4.26. The Maintain Code Tables Distributed to WPS.

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the COMMODITY CODE GROUP 1 table. A Unix script is then called which distributes the changes to the WPS sites.

Software Unit Name: commodity_cd_grp2.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the COMMODITY CODE GROUP 2 table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: content_distribution_cd.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the CONTENT DISTRIBUTION table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: damage_cd_pos1.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the DAMAGE CODE POSITION 1 table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: damage_cd_pos2.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the DAMAGE CODE POSITION 2 table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: damage_cd_pos3.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the DAMAGE CODE POSITION 3 table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: delete_reason_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the DELETE REASON CODE table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: flag_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the FLAG CODE table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: goods_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the GOODS CODE table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: handling_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the HANDLING CODE table.

A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: numeric_conversion.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the NUMERIC CONVERSION table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: ocean_carrier.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the OCEAN CARRIER table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: operator_cd.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the OPERATOR CODE table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: package_cd.frm

Software Unit Type: SQL*Form

Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the PACKAGE CODE table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: port.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the PORT table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: shippable_status_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the SHIPPABLE STATUS CODE table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: tac.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the TRANSPORTATION ACCOUNT CODE table.

Software Unit Name: type_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the TYPE CODE table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: type_vessel_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the TYPE VESSEL CODE

table. A Unix script is then called which distributes the changes to the WPS sites.

Software Unit Name: van_owner.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the VAN OWNER table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: vstat_terms_carriage.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the VESSEL STATUS TERMS CARRIAGE CODE table. A UNIX script is then called which distributes the changes to the WPS sites.

Monitor Automated Data Extracts Subfunction

This function allows selected users (data administrators or functional ICDB users) to change the frequency of data extracts from ICDB for GTN recipients or to resend data file extracts to various regional database recipients. Figure 4.27 shows the logical flow for this subfunction. The following are the software units that are components of this subfunction.

Software Unit Name: reptdata.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to resend any ICDB recipient a data extract file. The user marks the appropriate recipient(s). In the background, the recipient_data_files table is updated with the system date, and a Unix host command executes the Unix script "testhost" that resends data to the selected recipient(s).

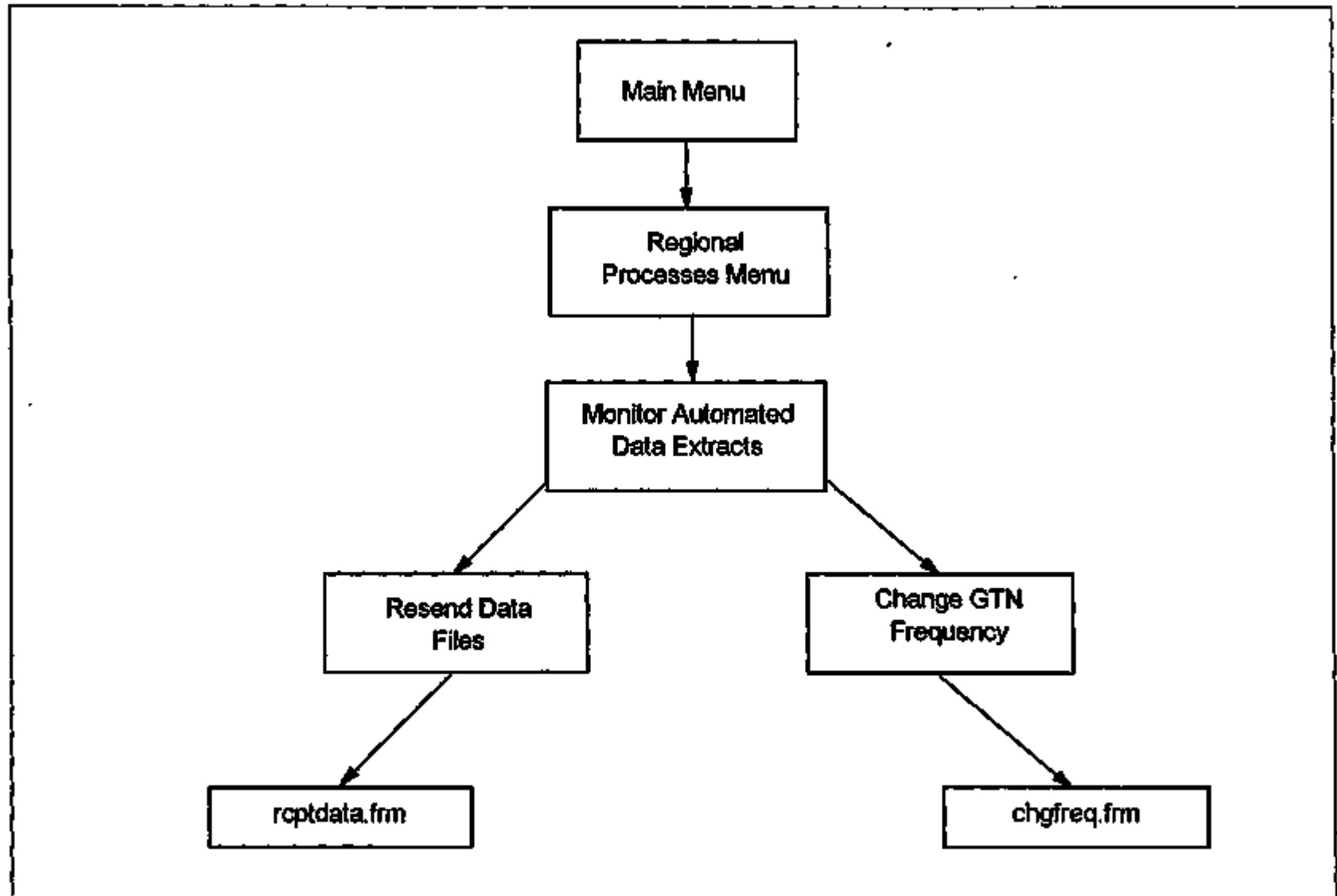


Figure 4.27. The Monitor Automated Data Extracts subfunction.

Software Unit Name: chgfreq.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows users to change the frequency (based on hours) that GTN data extract files are sent by the ICDB system. It updates the cron_table with the system date and calls a Unix script "cronjob" that updates the Unix cron file.

Software Unit Name: cronjob
Software Unit Type: Unix script file
Software Unit Description: The cronjob script file is called via the host command from the Change GTN Frequency screen and performs the following functions:

1. Executes a SQL*Plus command to select data from the cron_table table into a flat file.
2. Executes the Unix utility 'crontab' to delete any existing cron entry for the user.
3. Executes the Unix utility 'crontab' to create an existing cron entry for the user using the flat file selected from the cron_table table.
4. Removes the flat file selected from the cron_table table.

Update Discharge Dates Subfunction

The Update Discharge Dates Subfunction allows certain users the ability to update discharge dates for VOYDOCs. The following software units are components of the update Discharge Dates Subfunction.

Software Unit Name: updtvsl.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to enter, for a VOYDOC/POD combination, a discharge date if the discharge date is null for

that combination. The discharge date will be updated in the VOYDOC port pair table as well as in the shipment unit table for those records with null discharge dates that belong to that VOYDOC port pair.

Print Open Discharge Dates Listing Subfunction

The Print Open Discharge Dates Listing Subfunction allows the user to print VOYDOC port pair records that have a null discharge date. Figure 4.28 shows the logical flow for the software units that are components of this subfunction. The following are the software units that are components of this subfunction.

Software Unit Name: disch_dte.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to select a printer for printing the Open Discharge Dates Listing.

Software Unit Name: voydoc_lacking_disch_dates.rep
Software Unit Type: SQL*Reportwriter
Software Unit Description: This report shows the VOYDOC port pairs that have open discharge dates.

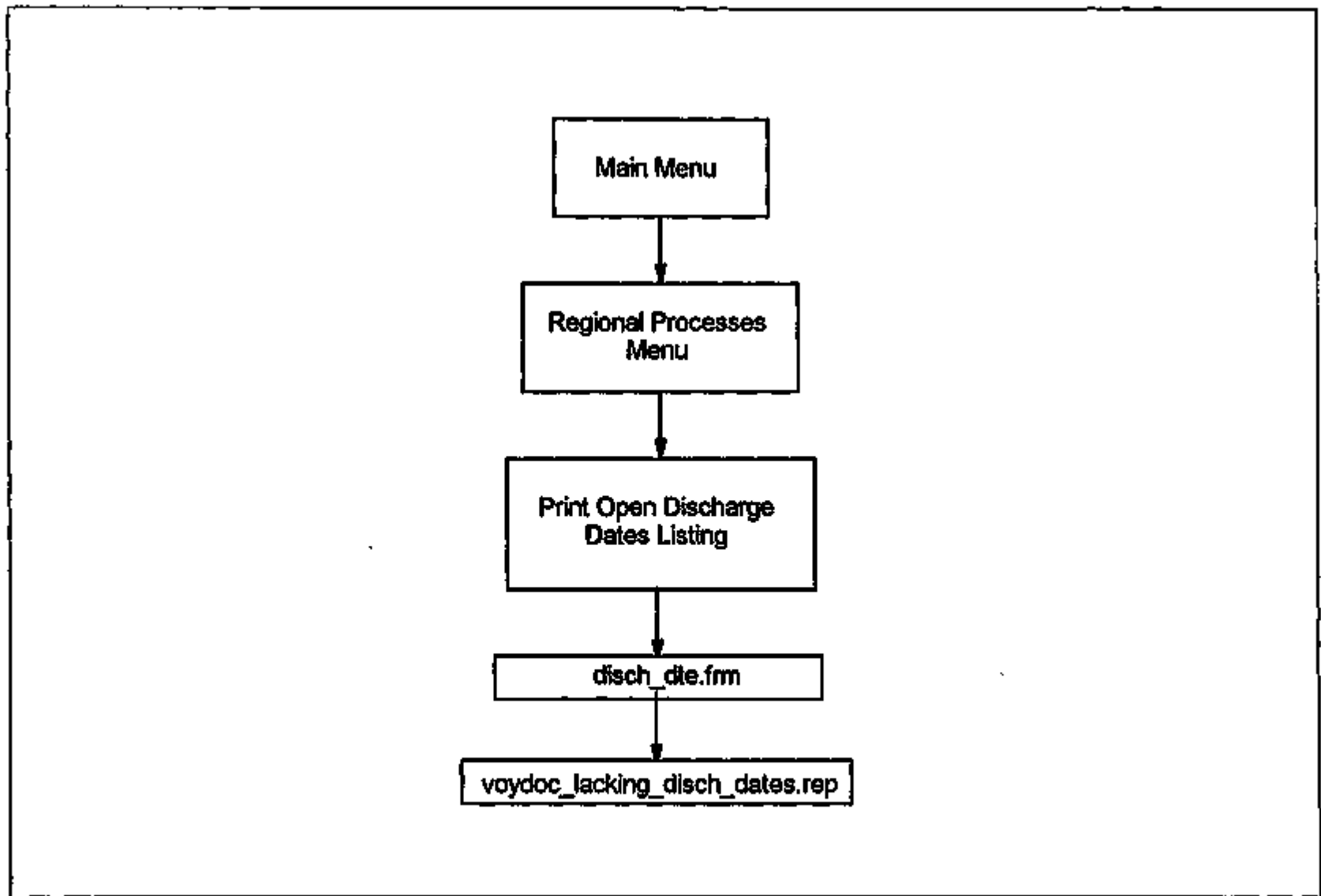


Figure 4.28. The Print Open Discharge Dates subfunction.

4.4.1.5 The Utilities Subsystem. Figure 4.29 shows the logical flow for the Utilities Subsystem. This subsystem will allow users to print files they have created in IQ or through other user interface screens. This subfunction consists of the following software units.

Software Unit Name: iq_util.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to print or download to floppy disk any created file.

Software Unit Name: other_util.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the user to print or download to floppy disk those files that were not created using I.Q.

4.4.2 Automated Processes

Automated processes are those processes that are run without user intervention. They may occur at predefined time intervals, or they may be initiated by a specific event or condition. The automated ICDB processes consist of four subsystems: processes for data transfer, for data conversion, for data distribution, and for system operational integrity checks (see Fig. 4.30).

4.4.2.1 Data Transfer Subsystem. The ICDB system automatically exchanges data with other systems to which it is connected. The Hubs pull data from the WPS port sites and push data to the central server.

The exchanges are driven by shell scripts that are started by the system clock. These scripts first check to see whether data can be exchanged, then check to see if data needs to be exchanged, then exchanges the data. The shell scripts control Oracle7 SQL scripts that utilize Oracle7 SQL*Net V2. Oracle7 SQL*Net V2 allows SQL statements to span databases on geographically separated machines. Figure 4.31 shows the logical flow for the Data Transfer subsystem. The following are the software units for this subsystem.

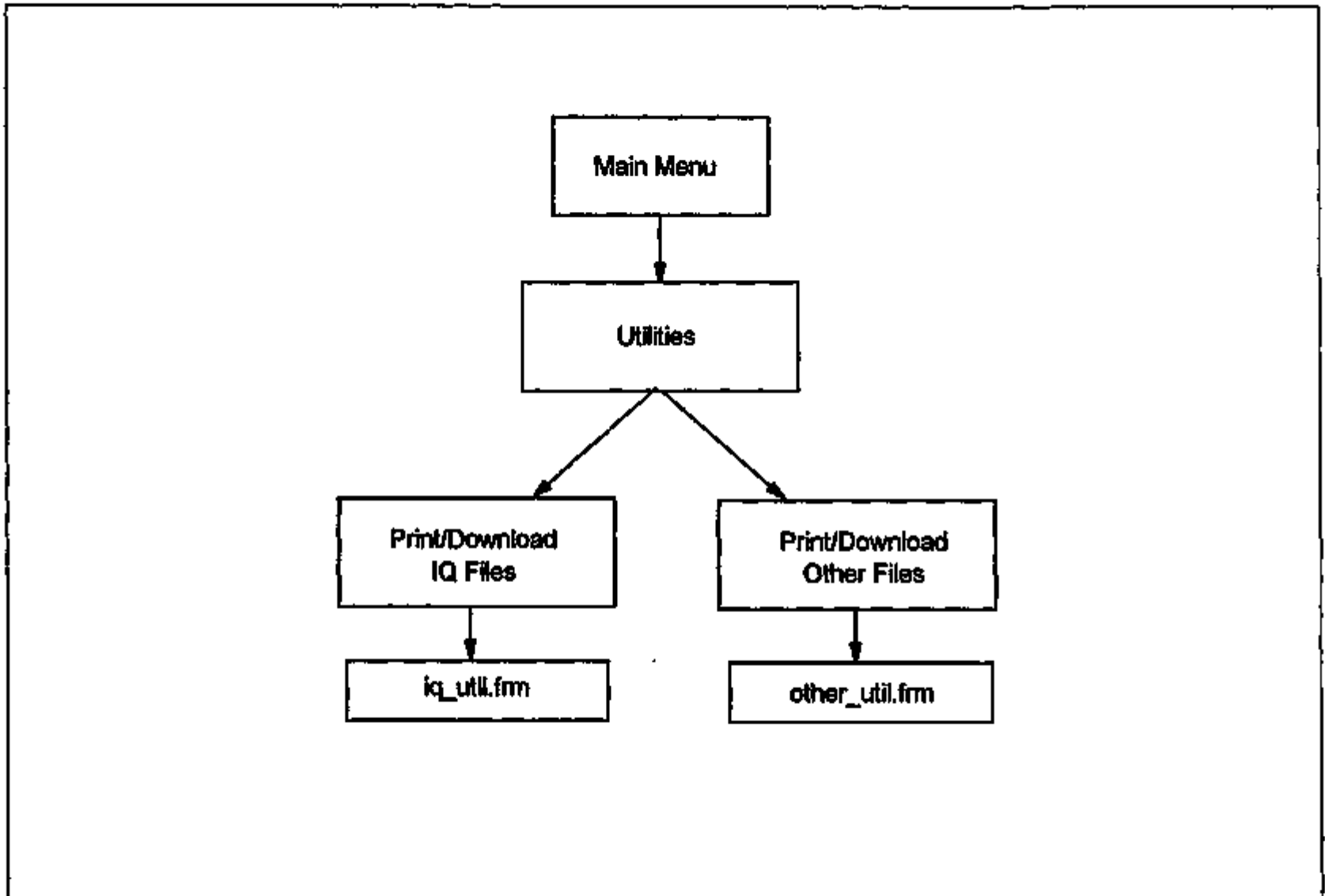


Figure 4.29. The Utilities subsystem.

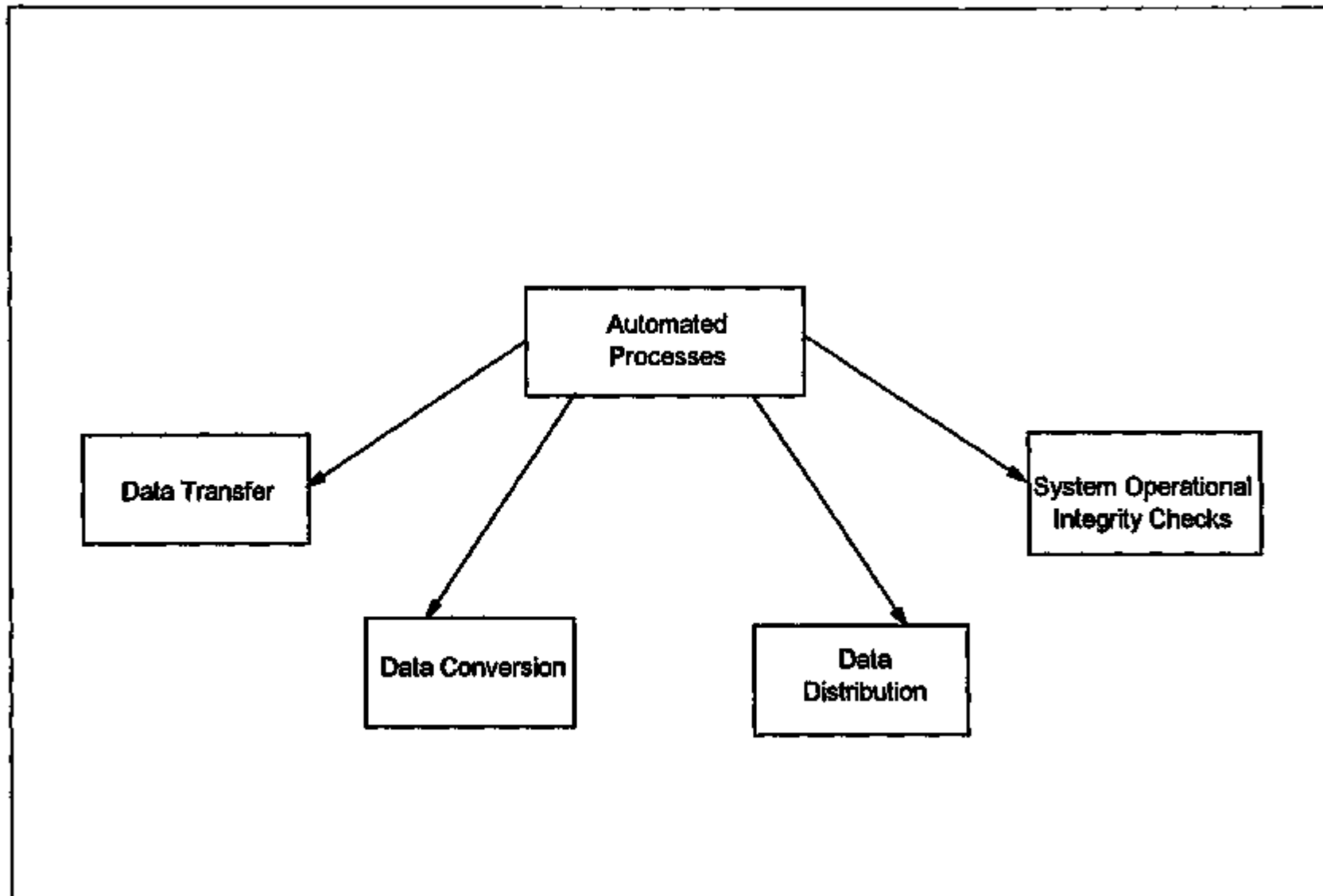


Figure 4.30. The Automated Processes subsystem.

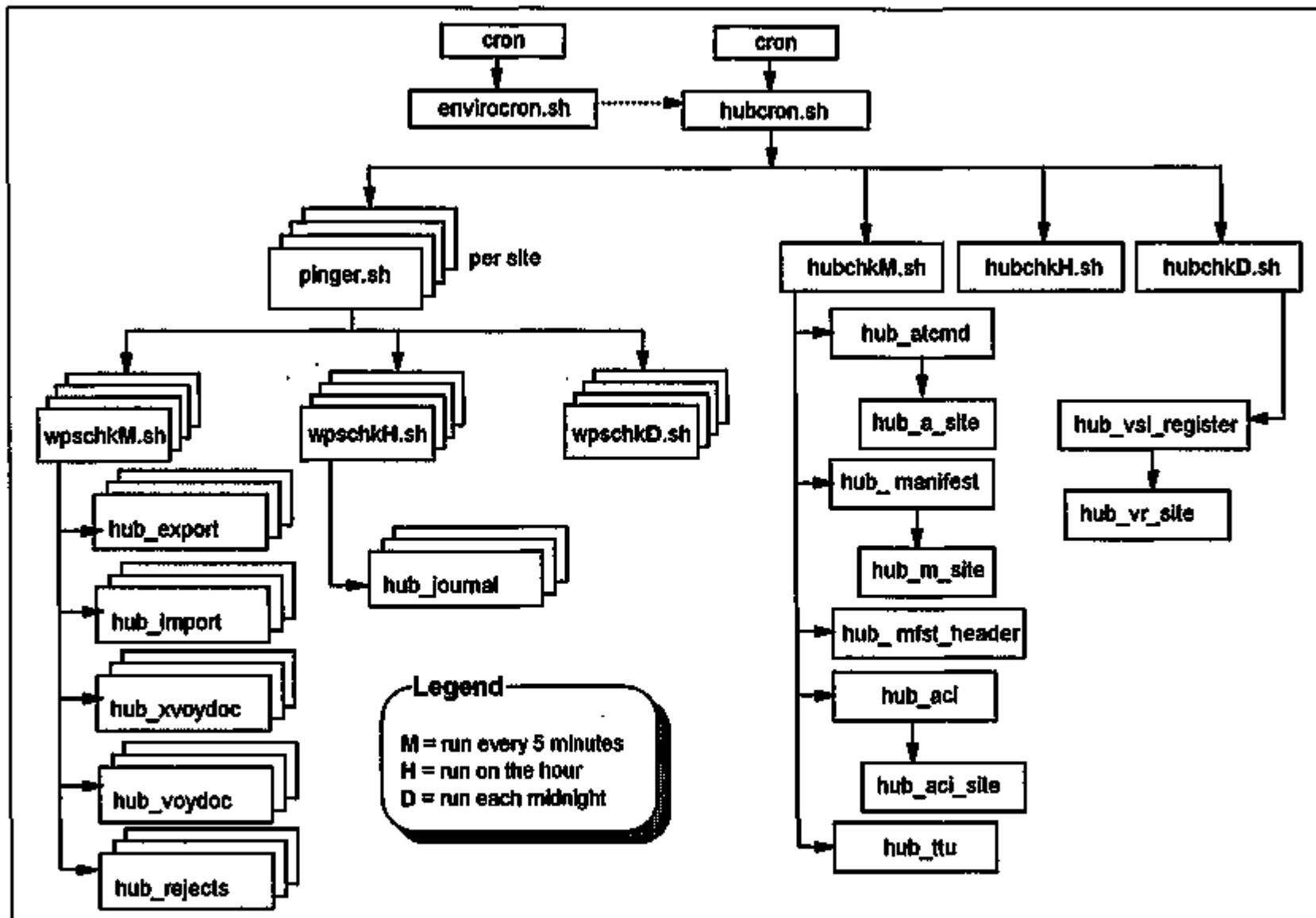


Figure 4.31. The Data Transfer subsystem.

Software Unit Name: envirocron.sh
Software Unit Type: Bourne shell script
Software Unit Description: This shell script is called by the system clock through crontab every 5 minutes but 1 minute before the call to hubcron.sh. It checks the data exchange environment to make sure all the environment variables are available and correct before hubcron.sh runs. It also checks to see if the Central server is down for maintenance.

Software Unit Name: hubcron.sh
Software Unit Type: Bourne shell script
Software Unit Description: This shell script is called by the system clock through crontab every 5 minutes but 1 minute after the call to envirocron.sh. It establishes the communications method to the Central server, checks for ORACLE availability on the Central server, and starts any Hub processes. Using a list of WPS and Hub sites created by the Oracle7 site maintenance utilities, it establishes processes to extract data from each of those sites.

Software Unit Name: pinger.sh
Software Unit Type: Bourne shell script
Software Unit Description: Pinger is called for each WPS site in the site list (Hubs are checked for ATCMDs and manifest data). It uses the Unix utility "ping" and attempts to "ping" the WPS site to see if the site can be reached. If the ping fails, the software attempts to dial-up the site to establish contact. If the dial-up is successful, the fault lies with the communications and the system administrator is notified; the data extraction processing continues. If the dial-up fails, the fault lies with the site and the system administrator is notified; no data can be extracted from that site.

Software Unit Name: wpschkm.sh
Software Unit Type: Bourne shell script
Software Unit Description: Wpschkm.sh is called for 5-minute processes for each WPS site that can be communicated with to see if there is any data that needs to be extracted and sent to the ICDB Server. It checks two tables at the site for any unread data: drecno and a journal table. This script calls hub_export, hub_import, hub_voydoc, hub_xvoydoc, and hub_rejects.

Software Unit Name: wpschkh.sh
Software Unit Type: Bourne shell script
Software Unit Description: wpschkh.sh is called for hourly processes for each WPS site that can be communicated with to see if there is any data that needs to be extracted and sent to the ICDB server. It checks two tables at the site for any unread data: drecno and a journal table. This script calls hub_journal.

Software Unit Name: wpschkd.sh
Software Unit Type: Bourne shell script
Software Unit Description: wpschkd.sh is called for daily processes for each WPS site that can be communicated with to see if there is any data that needs to be extracted and sent to the ICDB server. It checks two tables at the site for any unread data: drecno and a journal table.

Software Unit Name: hubchkm.sh
Software Unit Type: Bourne shell script
Software Unit Description: Hubchkm.sh is called for 5-minute processes to check the named Hub for the presence of unread ATCMD or manifest data. This script calls hub_srvr_dist, hub_atcmd, hub_manifest, hub_mfst_header, and hub_aci.

Software Unit Name: hubchkh.sh
Software Unit Type: Bourne shell script
Software Unit Description: hubchkh.sh is called for hourly processes to check the named Hub.

Software Unit Name: hubchkd.sh
Software Unit Type: Bourne shell script
Software Unit Description: hubchkd.sh is called for daily processes to check the named Hub. It calls hub_vsl_register.

Software Unit Name: hub_export
Software Unit Type: Bourne shell script
Software Unit Description: Hub_export transfers unread export data from the WPS site to the Hub and then to the Server if the Server is available.

Software Unit Name: hub_import
Software Unit Type: Bourne shell script
Software Unit Description: Hub_import transfers unread import data from the WPS site to the Hub and then to the Server if the Server is available.

Software Unit Name: hub_journal
Software Unit Type: Bourne shell script
Software Unit Description: Hub_journal transfers unread journal data from the WPS site to the Hub and then to the Server if the Server is available.

Software Unit Name: hub_atcmd
Software Unit Type: Bourne shell script
Software Unit Description: Hub_atcmd distributes ATCMD data to the Server and determines if the data should be transferred to the WPS site.

Software Unit Name: hub_manifest
Software Unit Type: Bourne shell script

Software Unit Description: Hub_manifest distributes manifest data to the Server and determines if the data should be transferred to the WPS site.

Software Unit Name: hub_ttu

Software Unit Type: Bourne shell script

Software Unit Description: Hub_ttu transfers TTU data (i.e., transactions on shipment unit discharge and disposition) to the Server.

Software Unit Name: xfer_site_hub_e

Software Unit Type: Stored Procedure

Software Unit Description: Xfer_site_Hub_e transfers export data from WPS tables to transfer tables at the WPS site.

Software Unit Name: xfer_site_hub_i

Software Unit Type: Stored Procedure

Software Unit Description: Xfer_site_hub_i transfers import data from WPS tables to transfer tables at the WPS site.

Software Unit Name: xfer_site_hub_j

Software Unit Type: Stored Procedure

Software Unit Description: Xfer_site_hub_j transfers journal data from WPS tables to transfer tables at the WPS site.

Software Unit Name: xfer_atcmd_bkg

Software Unit Type: Stored Procedure

Software Unit Description: Xfer_atcmd_bkg transfers booking and ATCMD data from working tables to transfer tables and checks to see if there is an existing record for the same shipment on the Server.

Software Unit Name: hub_a_site

Software Unit Type: Bourne shell script

Software Unit Description: Hub_a_site transfers ATCMD data to the WPS sites.

Software Unit Name: xfer_mfst
Software Unit Type: Stored Procedure
Software Unit Description: Xfer_mfst transfers manifest data from working tables to transfer tables.

Software Unit Name: hub_m_site
Software Unit Type: Bourne shell script
Software Unit Description: Hub_m_site transfers manifest data to the WPS sites.

Software Unit Name: hub_aci
Software Unit Type: Bourne shell script
Software Unit Description: Transfers ACI data from the Hub to the server if the server is available.

Software Unit Name: hub_aci_site
Software Unit Type: Bourne shell script
Software Unit Description: Transfers ACI data to a WPS port site.

Software Unit Name: hub_mfst_header
Software Unit Type: Bourne shell script
Software Unit Description: Transfers manifest header data to the server if the server is available.

Software Unit Name: hub_rejects
Software Unit Type: Bourne shell script
Software Unit Description: Transfers reject data from a WPS port site to the Hub and on to the Central server if it is available.

Software Unit Name: hub_srvr_dist
Software Unit Type: Bourne shell script

Software Unit Description: Transfers data from the server if it is available back to the Hub. This data is then subsequently transferred to the WPS port sites.

Software Unit Name: hub_voydoc

Software Unit Type: Bourne shell script

Software Unit Description: Transfer import voydoc data from WPS port site to the Hub and on to the Central server if it is available.

Software Unit Name: hub_vr_site

Software Unit Type: Bourne shell script

Software Unit Description: Transfer vessel register data to the WPS port sites from the Hub.

Software Unit Name: hub_vsl_register

Software Unit Type: Bourne shell script

Software Unit Description: Transfer vessel register data from the Server, if it is available, to the Hub.

Software Unit Name: hub_xvoydoc

Software Unit Type: Bourne shell script

Software Unit Description: Transfer xvoydoc data from WPS to server.

4.4.2.2 System Integrity Checks Subsystem. In order to assure that the system exhibits the highest availability possible, each machine is equipped with software that can check on the integrity of the other machines in the system. Every minute, the ICDB Server "pings" all the Hubs connected to it. If there is no response, it attempts to dial-up the unresponsive Hub. If it reaches the unresponsive Hub by using dial-up, the system administrator is notified of the communications problem. If the Hub is unreachable by dial-up, the workload for that Hub is transferred to the nearest (or least utilized) Hub and data extraction continues. The system administrator is notified of the automatic change in configuration.

4.4.2.3 Data Conversion Subsystem. The data conversion software units convert data from non-ICDB database format and insert rows into or update rows of the appropriate tables. The data that is converted is data that has been transferred from WPS or other systems to ICDB. This data is stored in ICDB in its original format until it is successfully converted and inserted. Figure 4.32 shows the logical flow of the Data Conversion subsystem. The following software units are components of the data conversion subsystem.

Software Unit Name: server.sh
Software Unit Type: Bourne shell script
Software Unit Description: This script checks whether there is any export, import, or journal data that has not yet been processed, and if there is, it starts the appropriate conversion process. It continues in this loop until there is no more data that is unprocessed.

Software Unit Name: hubchk.sh
Software Unit Type: Bourne shell script
Software Unit Description: This script pings each hub known in the site_registration table one at a time. If the ping fails, it attempts to dial into that Hub. If that fails, the Hub's workload is transferred to an alternate Hub identified in a list of alternates for that Hub. The Hub is continually monitored. If the Hub comes back alive, the workload is transferred back.

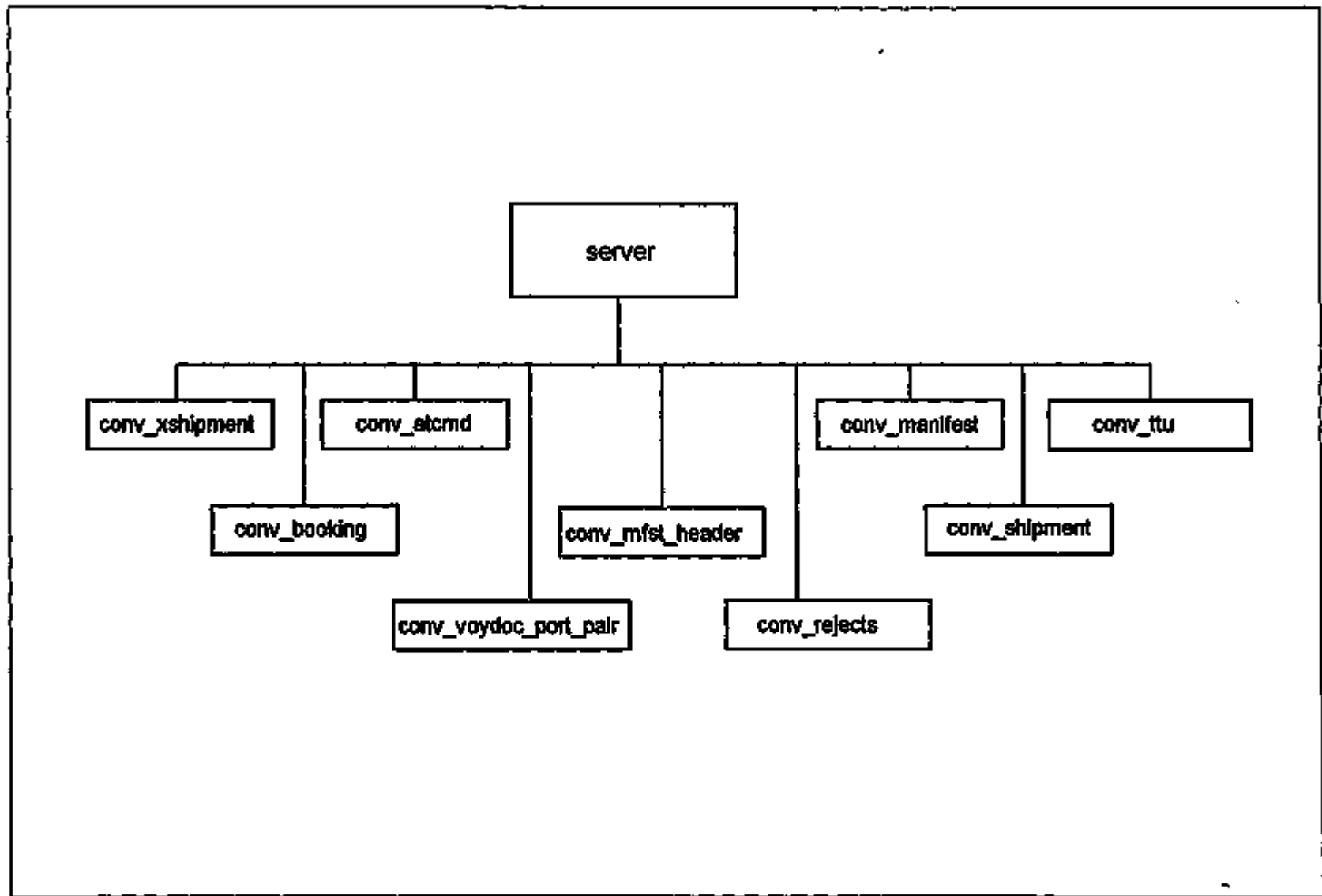


Figure 4.32. The Data Conversion subsystem.

Software Unit Name: broken_hub.sh
Software Unit Type: Bourne shell script
Software Unit Description: This script transfers the workload of a bad Hub to a good Hub. All the WPS sites that were serviced by the bad Hub are notified that they are now being serviced by the good Hub.

Software Unit Name: repaired_hub.sh
Software Unit Type: Bourne shell script
Software Unit Description: This script transfers a Hubs workload from the surrogate Hub back to the original Hub. All the WPS sites that were transferred are notified that they are now being serviced by their original Hub.

Software Unit Name: conv_xshipment
Software Unit Type: stored procedure
Software Unit Description: This procedure converts WPS export data and, if the record is for a new shipment unit, it inserts the data into ICDB shipment_unit and related tables. If the record is for an existing shipment unit, the existing records are updated.

Software Unit Name: conv_booking
Software Unit Type: stored procedure
Software Unit Description: This procedure converts booking data from the ATCMD processing module booking system and, if the record is for a new shipment unit, it inserts the data into ICDB shipment_unit and related tables. If the record is for an existing shipment unit, the existing records are updated.

Software Unit Name: conv_atcmd
Software Unit Type: stored procedure

Software Unit Description: This procedure converts ATCMD data from the ATCMD processing module and, if the record is for a new shipment unit, it inserts it into ICDB shipment_unit and related tables. If the record is for an existing shipment unit, the existing records are updated.

Software Unit Name: conv_ttu

Software Unit Type: Stored procedure

Software Unit Description: This procedure updates the discharge and disposition columns of the shipment unit table for records with a matching TCN and VOYDOC.

Software Unit Name: conv_manifest

Software Unit Type: stored procedure

Software Unit Description: This procedure converts manifest data and, if the record is for a new shipment unit, it inserts it into ICDB shipment_unit and related tables. If the record is for an existing shipment unit, the existing records are updated.

Software Unit Name: conv_shipment

Software Unit Type: stored procedure

Software Unit Description: This procedure converts WPS import data and, if the record is for a new shipment unit, it inserts it into ICDB shipment_unit and related tables. If the record is for an existing shipment unit, the existing records are updated.

Software Unit Name: conv_voydoc_port_pair

Software Unit Type: stored procedure

Software Unit Description: This procedure updates or inserts data into voydoc_port pair from xfer_xvoydoc and xfer_voydoc.

Software Unit Name: conv_mfst_header

Software Unit Type:	stored procedure
Software Unit Description:	This procedure inserts data from <code>xfer_mfst_header</code> into <code>voydoc_port_pair</code> .
Software Unit Name:	<code>conv_rejects</code>
Software Unit Type:	stored procedure
Software Unit Description:	This procedure processes records from <code>xfer_rejects</code> which are rejected because they are duplicates. It finds the record with matching <code>su_id</code> , updates it with booking data from the record with the matching <code>rejected_su_id</code> , and deletes the record in <code>shipment_unit</code> with <code>su_id</code> matching <code>rejected_su_id</code> .

4.4.2.4 Data Distribution Subsystem. As the central repository for cargo movement data, ICDB will serve as a source of pre-defined data files for other MTMC and non-MTMC systems. Pre-specified data files will include data elements as specified by interface agreements with systems receiving pre-defined data files. Data distribution processes will perform automated data extracts to Unix files for transfer to these systems.

Figure 4.33 indicates the various interfacing systems and the Unix script files used to generate data files for each system. Figure 4.34 shows the logical flow for the programs that are components of this subsystem.

The data distribution subsystem consists of two subfunctions: data file generation and data file transmission.

Data File Generation Subfunction.

Generation of pre-defined data files will be an automated process as described in the following software units. Figure 4.35 shows the logical flow for the programs that are components of this subsystem.

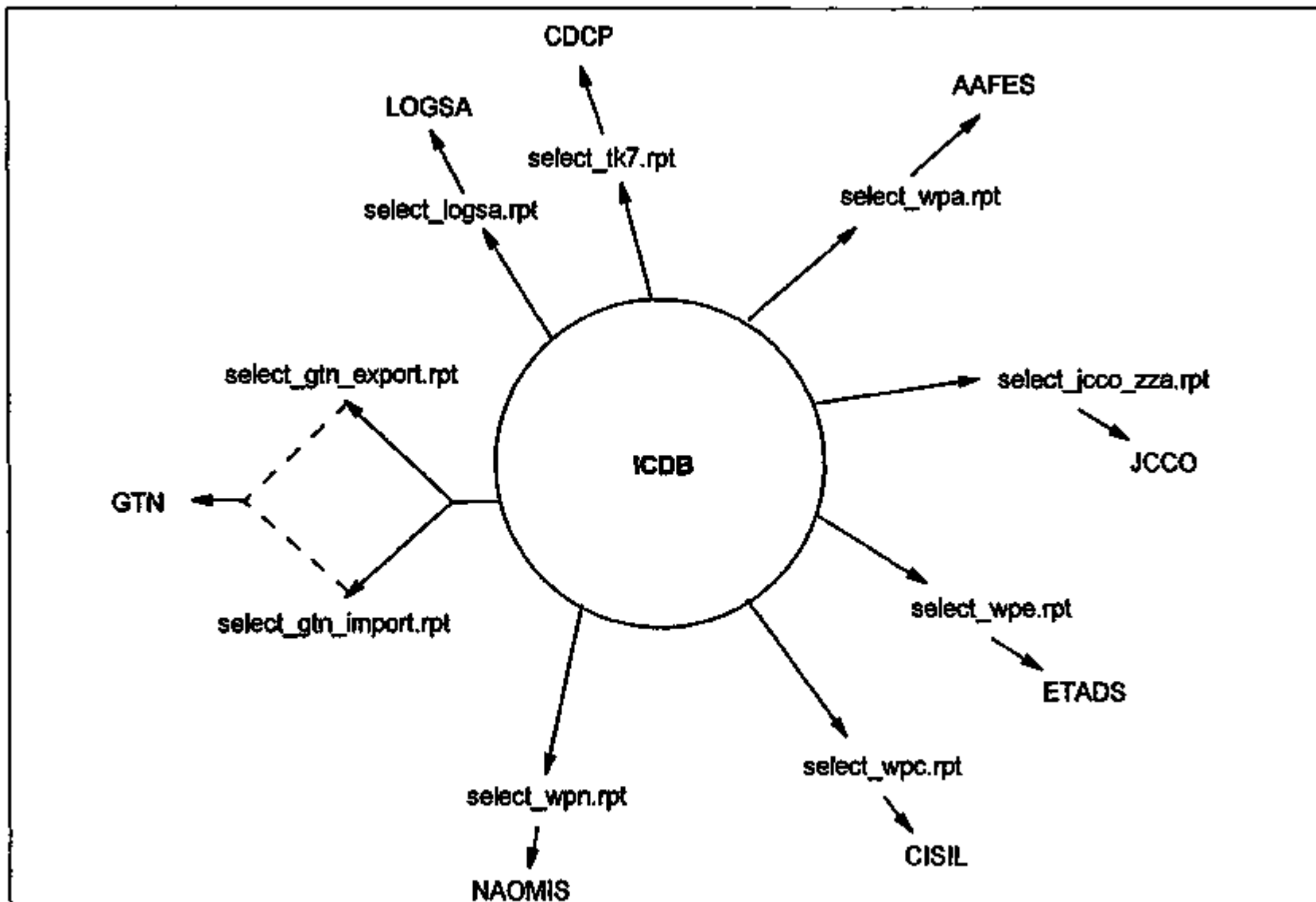


Figure 4.33. ICDB scripts for data recipients.

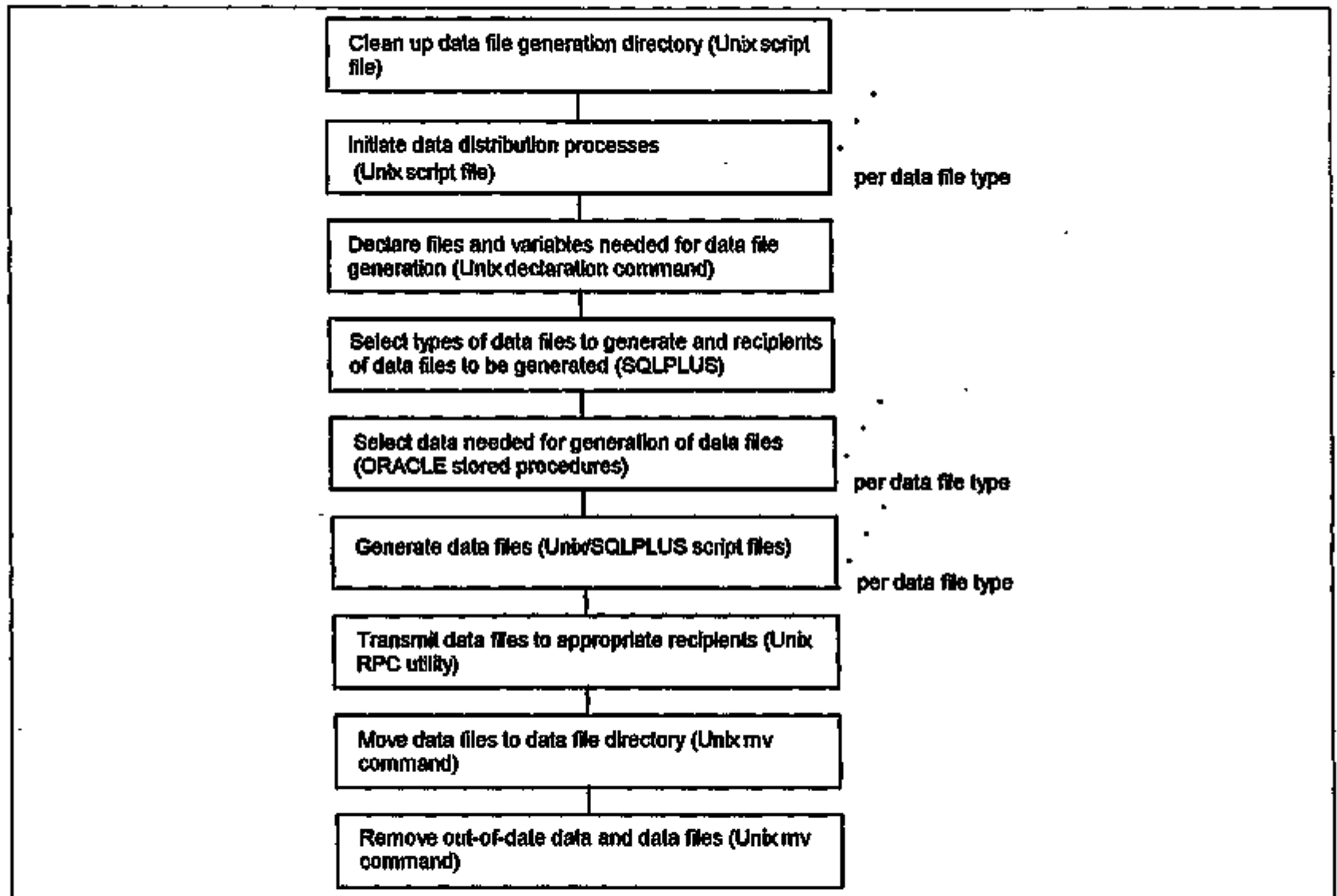


Figure 4.34. The Data Distribution subsystem.

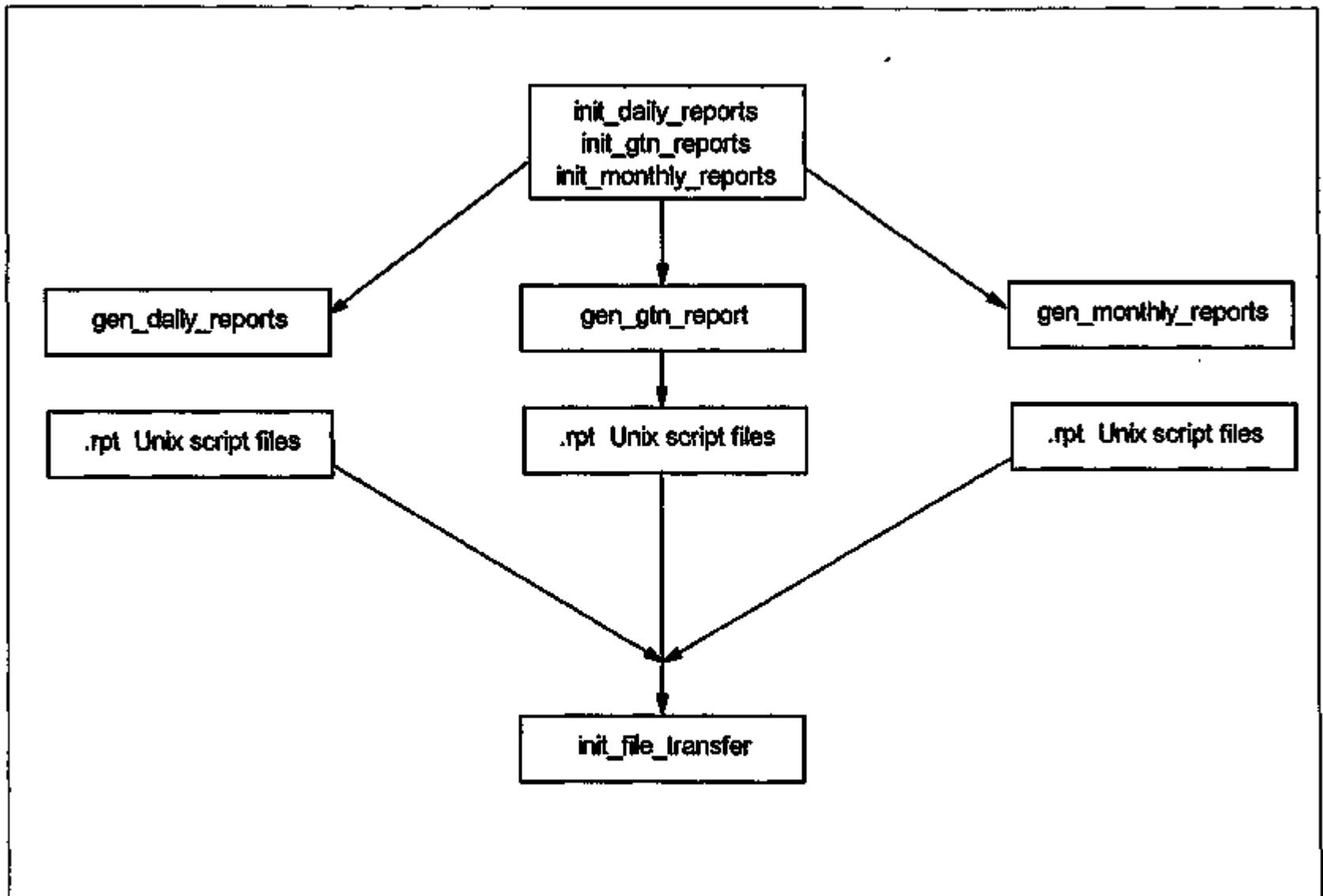


Figure 4.35. The Data File Generation subfunction.

Software Unit Name: `init_gtn_reports, init_monthly_reports, init_daily_reports`

Software Unit Type: Unix script files

Software Unit Description: The execution of applications to initiate the extraction of data and generation of data files, and transmission of data for recipients is controlled by the Unix 'cron' utility. At times specified by the cron utility the execution of the Unix script files listed above is begun. These Unix script files will be the driving applications for the Data Distribution Subsystem and will execute Unix commands and embedded SQLPLUS commands. The functionality of each of the Unix script files conforms to the following basic pattern:

1. File and variable declaration and variable value assignment.
2. Determination of types of data files to generate.
3. Determination of recipients of data files.
4. Selection of data appropriate to type of files to generate.
5. Creation of flat files according to specified data selection criteria.
6. Movement of data files to data files directory.
7. Removal of out-of-date data files, data, and temporary files.

Software Unit Name: `init_file_transfer`

Software Unit Type: Unix script files

Software Unit Description: The `init_file_transfer` will initiate the transmission of data files generated on a daily or monthly basis with the exception of data files for GTN. The `init_file_transfer` script file will accept the code name for the recipient and the file name to be transmitted as arguments. A SQL*Plus command will be executed to select information needed for the automatic transmission of the file to the recipient system. Based on the

recipient code, an appropriate Unix script file will be called to attempt the transmission of the file to the recipient system. The `init_file_transfer` script file is called from either the `init_daily_reports` script file or the `init_monthly_reports` script file.

Software Unit Name: `init_gtn_file_transfer`
Software Unit Type: Unix script files
Software Unit Description: The `init_file_transfer` will initiate the transmission of data files generated for GTN. the `init_gtn_file_transfer` will accept the code name for GTN and the two files to be transmitted to GTN as arguments. A SQL*Plus command will be executed to select.

Software Unit Name: `gen_gtn_reports`
Software Unit Type: Unix script file
Software Unit Description: The `gen_gtn_reports` script file is called from the `init_gtn_reports` script file. The `gen_gtn_reports` script file calls the following Unix script files to select data into flat files: `select_gtn_export.rpt` and `select_gtn_import.rpt`.

Software Unit Name: `gen_monthly_report`
Software Unit Type: Unix script file
Software Unit Description: The `gen_monthly_reports` script file is called from the `init_monthly_reports` script file and calls an appropriate Unix script file to select data into a flat file for either CDCP or CISIL.

Software Unit Name: `gen_daily_reports`
Software Unit Type: Unix script file
Software Unit Description: The `gen_daily_reports` script file is called from the `init_daily_reports` script file. It calls an appropriate Unix

script file to select data into a flat file based on the "file recipient" passed as a parameter from the init_daily_reports script file.

Software Unit Name: capture_journal
Software Unit Type: ORACLE stored procedure
Software Unit Description: The capture_journal stored procedure is executed every hour as part of the cron entry for hub processing. The capture_journal procedure inserts transaction records into the hub.xfer_journal table.

Software Unit Name: set_type_update
Software Unit Type: ORACLE stored procedure
Software Unit Description: The set_type_update stored procedure is executed from the capture_journal stored procedure. The set_type_update procedure inserts records into the daily_suids and monthly_report_fields table as transaction records are inserted into the hub.xfer_journal table.

Software Unit Name: delete_rpt_records
Software Unit Type: ORACLE stored procedure
Software Unit Description: The delete_rpt_records stored procedure is executed from the init_daily_reports, init_gtm_reports, and init_monthly_reports script files. The delete_rpt_records procedure deletes records from the daily_suids, daily_voydocs, hub.xfer_journal, and monthly_report_fields tables that are over seven days old.

Software Unit Name: insert_daily_suids
Software Unit Type: ORACLE stored procedure
Software Unit Description: The insert_daily_suids stored procedure is executed from from the set_type_update stored procedure. The set_type_update inserts records into the daily_suids table.

Software Unit Name: insert_monthly_report_fields
Software Unit Type: ORACLE stored procedure
Software Unit Description: The insert_monthly_report_fields stored procedure is executed from the capture_journal stored procedure. The insert_monthly_report_fields procedure inserts records into the monthly_report_fields table.

Software Unit Name: insert_report_fields
Software Unit Type: ORACLE stored procedure
Software Unit Description: The insert_report_fields stored procedure is executed from the select_daily_report_data and select_gtn_report_data stored procedures. The insert_report_fields procedure inserts records into the report_fields table.

Software Unit Name: select_daily_report_data
Software Unit Type: ORACLE stored procedure
Software Unit Description: The select_daily_report_data stored procedure is executed from the init_daily_reports script file. The select_daily_report_data selects cargo data which will be used in data files generated on a daily basis.

Software Unit Name: select_daily_tv_data
Software Unit Type: ORACLE stored procedure
Software Unit Description: The select_daily_tv_data stored procedure is executed from the init_daily_reports script file. The select_daily_tv_data selects vessel data which will be used in data files generated on a daily basis.

Software Unit Name: select_gtn_report_data
Software Unit Type: ORACLE stored procedure
Software Unit Description: The select_gtn_report_data stored procedure is executed from the init_gtn_reports script file. The select_gtn_report_data

selects cargo data which will be used in data files generated for GTN.

Software Unit Name: select_gtn_voydoc_data
Software Unit Type: ORACLE stored procedure
Software Unit Description: The select_gtn_voydoc_data stored procedure is executed from the init_gtn_reports script file. The select_gtn_voydoc_data selects vessel data which will be used in data files generated for GTN.

Software Unit Name: select_logsa.rpt
Software Unit Type: Unix script file
Software Unit Description: The select_logsa.rpt script file will select the data needed for the data file for LOGSA into a flat file and will update those records in the daily_suids and daily_voydocs tables which were used in creation of the flat file.

Software Unit Name: select_wpc.rpt
Software Unit Type: Unix script file
Software Unit Description: The select_wpc.rpt script file will select the data needed for the data file for CISIL into a flat file and will update those records in the monthly_report_fields table which were used in creation of the flat file.

Software Unit Name: select_wpa.rpt
Software Unit Type: Unix script file
Software Unit Description: The select_wpa.rpt script file will select the data needed for the data file for AAFES into a flat file and will update those records in the daily_suids and daily_voydocs tables which were used in creation of the flat file.

Software Unit Name: select_wpe.rpt

Software Unit Type: Unix script file
Software Unit Description: The select_wpe.rpt script file will select the data needed for the data file for ETADS into a flat file and will update those records in the daily_suids and daily_voydocs tables which were used in creation of the flat file.

Software Unit Name: select_wpn.rpt
Software Unit Type: Unix script file
Software Unit Description: The select_wpn.rpt script file will select the data needed for the data file for NAOMIS into a flat file and will update those records in the daily_suids and daily_voydocs tables which were used in creation of the flat file.

Software Unit Name: select_jcco_zza.rpt
Software Unit Type: Unix script file
Software Unit Description: The select_jcco_zza.rpt script file will select the data needed for the data file for JCCO into a flat file and will update those records in the daily_suids and daily_voydocs tables which were used in creation of the flat file.

Software Unit Name: select_tk7.rpt
Software Unit Type: Unix script file
Software Unit Description: The select_tk7.rpt script file will select the data needed for the data file for CDCP into a flat file and will update those records in the monthly_report_fields table which were used in creation of the flat file.

Software Unit Name: select_gtn_export.rpt, select_gtn_import.rpt
Software Unit Type: Unix script files
Software Unit Description: The select_gtn_export.rpt and select_gtn_import.rpt script files will select the data needed for the export and import data

files for GTN into flat files and will update those records in the `daily_suids` and `daily_voydocs` tables which were used in creation of the flat file.

Data Distribution - Data File Transmission Subfunction

Transmission of generated data files to appropriate recipient systems will be an automated process as described in the following software units. Figure 4.36 shows the logical flow for the programs that are components of this subfunction.

Software Unit Name:	<code>init_file_transfer</code>
Software Unit Type:	Unix Script File
Software Unit Description:	The <code>init_file_transfer</code> will initiate the transmission of data files generated on a daily or monthly basis with the exception of data files for GTN. The <code>init_file_transfer</code> script file will accept the code name for the recipient and the file name to be transmitted as arguments. A SQL*Plus command will be executed to select information needed for the automatic transmission of the file to the recipient system. Based on the recipient code, an appropriate Unix script file will be called to attempt the transmission of the file to the recipient system.

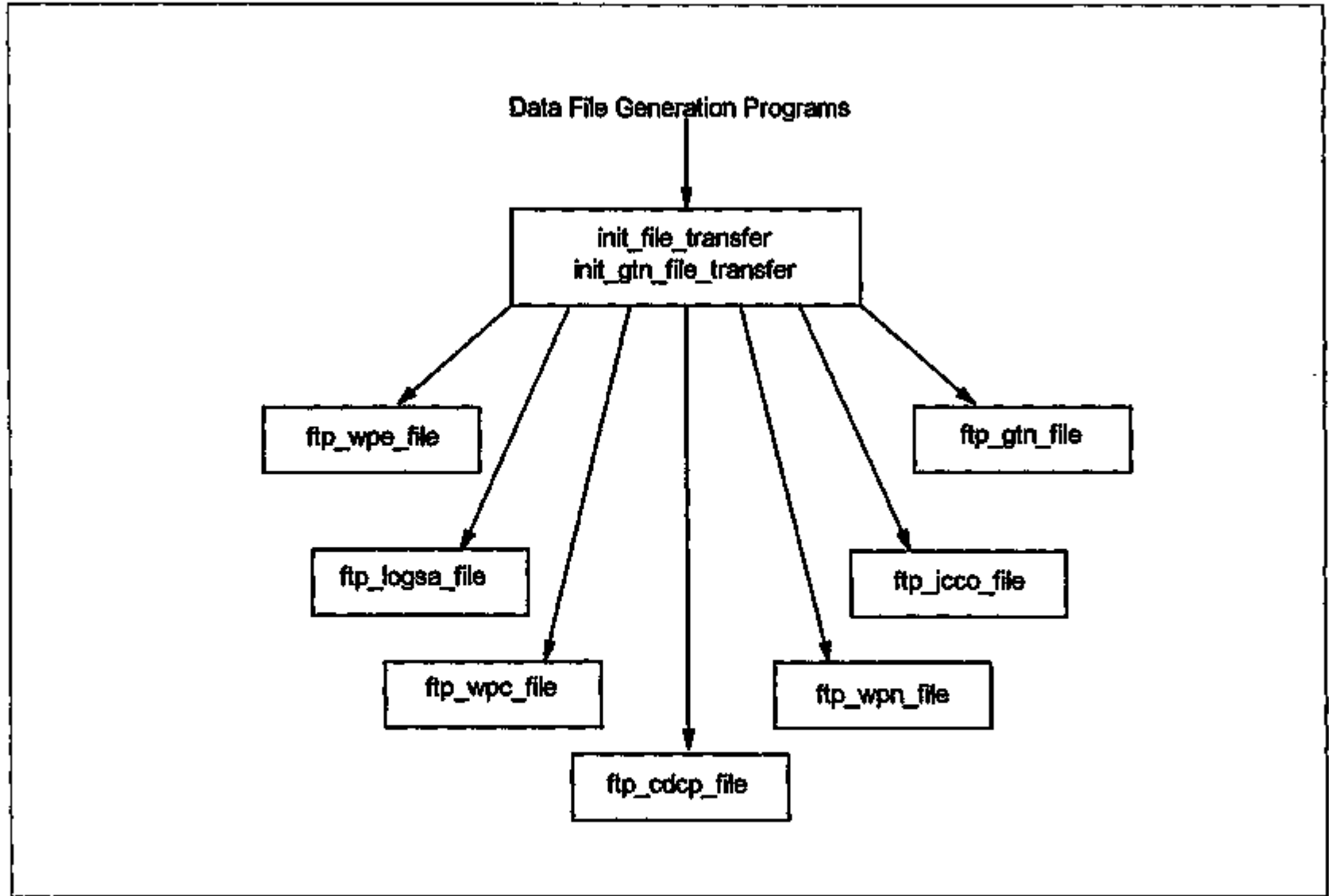


Figure 4.36. The Data File Transmission subfunction.

The `init_file_transfer` script file is called from either the `init_daily_reports` script file or the `init_monthly_reports` script file.

Software Unit Name: `cron_init_file_transfer`
Software Unit Type: Unix Script File
Software Unit Description: The `cron_init_file_transfer` will initiate the transmission of data files generated on a daily or monthly basis. The `cron_init_file_transfer` script file will accept the code name for the recipient and the file name to be transmitted as arguments. A SQL*Plus command will be executed to select information needed for the automatic transmission of the file to the recipient system. Based on the recipient code, an appropriate Unix script file will be called to attempt the transmission of the file to the recipient system. The `cron_init_file_transfer` script file is called from the `cron_xmit_data_file` script file.

Software Unit Name: `init_gtn_file_transfer`
Software Unit Type: Unix Script File
Software Unit Description: The `init_gtn_file_transfer` will initiate the transmission of data files generated for GTN. The `init_gtn_file_transfer` will accept the code name for GTN and the two files to be transmitted to GTN as arguments. A SQL*Plus command will be executed to select information needed for the automatic transmission of the file to GTN. The Unix script file, `ftp_gtn_file`, will be called to attempt the transmission of the files to GTN. The `init_gtn_file_transfer` is called from the `init_gtn_reports` script file.

Software Unit Name: `ftp_wpe_file`
Software Unit Type: Unix Script File

Software Unit Description: The ftp_wpe_file will attempt the transmission of the data file to ETADS and is called from the init_file_transfer script file.

Software Unit Name: cron_ftp_wpc_file

Software Unit Type: Unix Script File

Software Unit Description: The cron_ftp_wpc_file will attempt the transmission of the data file to ETADS and is called from the cron_init_file_transfer script file.

Software Unit Name: ftp_wpc_file

Software Unit Type: Unix Script File

Software Unit Description: The ftp_wpc_file will attempt the transmission of the data file to CISIL and is called from the init_file_transfer script file.

Software Unit Name: cron_ftp_wpc_file

Software Unit Type: Unix Script File

Software Unit Description: The cron_ftp_wpc_file will attempt the transmission of the data file to CISIL and is called from the cron_init_file_transfer script file.

Software Unit Name: ftp_wpn_file

Software Unit Type: Unix Script File

Software Unit Description: The ftp_wpn_file will attempt the transmission of the data file to NAOMIS and is called from the init_file_transfer script file.

Software Unit Name: cron_ftp_wpn_file

Software Unit Type: Unix Script File

Software Unit Description: The cron_ftp_wpn_file will attempt the transmission of the data file to NAOMIS and is called from the cron_init_file_transfer script file.

Software Unit Name: ftp_logsa_file
Software Unit Type: Unix Script File
Software Unit Description: The ftp_logsa_file will attempt the transmission of the data file to LOGSA and is called from the init_file_transfer script file.

Software Unit Name: cron_ftp_logsa_file
Software Unit Type: Unix Script File
Software Unit Description: The cron_ftp_logsa_file will attempt the transmission of the data file to LOGSA and is called from the cron_init_file_transfer script file.

Software Unit Name: ftp_jcco_file
Software Unit Type: Unix Script File
Software Unit Description: The ftp_jcco_file will attempt the transmission of the data file to JCCO and is called from the init_file_transfer script file.

Software Unit Name: cron_ftp_jcco_file
Software Unit Type: Unix Script File
Software Unit Description: The cron_ftp_jcco_file will attempt the transmission of the data file to JCCO and is called from the cron_init_file_transfer script file.

Software Unit Name: ftp_cdcg_file
Software Unit Type: Unix Script File
Software Unit Description: The ftp_cdcg_file will attempt the transmission of the data file to DAASC and is called from the init_file_transfer script file.

Software Unit Name: cron_ftp_cdcg_file
Software Unit Type: Unix Script File

Software Unit Description: The `cron_ftp_cdcop_file` will attempt the transmission of the data file to DAASC and is called from the `cron_init_file_transfer` script file.

Software Unit Name: `ftp_gtn_file`

Software Unit Type: Unix Script File

Software Unit Description: The `ftp_gtn_file` will attempt the transmission of the data files to GTN and is called from the `init_file_transfer` script file.

Software Unit Name: `cron_ftp_single_gtn_file`

Software Unit Type: Unix Script File

Software Unit Description: The `cron_ftp_single_gtn_file` will attempt the transmission of the data files to GTN and is called from the `cron_init_file_transfer` script file.

Data Distribution - Miscellaneous Script Files

Software Unit Name: `del_old_data_files`

Software Unit Type: Unix Script File

Software Unit Description: This script file will get a list of data files whose generation date is 7 days prior to the current date for the recipient `cd` passed as an argument. Each of these files will be removed from the `$HOME/rpt_files` directory if it has been transmitted to its recipient system. The record for the file will be deleted from the `recipient_data_files` table if it has been transmitted. The `del_old_data_files` script file is called from either the `init_daily_reports` or the `init_gtn_reports` script file.

Software Unit Name: `print_daily_log`

Software Unit Type: Unix Script File

Software Unit Description: The `print_daily_log` script file will print the `daily_log` file which is generated daily. The `daily_log` file will be deleted

after printing. The `print_daily_log` script file is called from the `datax` cron utility.

Software Unit Name: `reset_daily_data, reset_gtm_data`
Software Unit Type: Unix Script Files
Software Unit Description: The `reset_daily_data` and `reset_gtm_data` script files reset the `daily_report_flag` and `variable_report_flags` in the `daily_voydocs` and `daily_suids` tables. These script files are called from the `init_daily_reports` and `init_gtm_reports` script files.

Software Unit Name: `split_tk7_file`
Software Unit Type: Unit Script File
Software Unit Description: The `split_tk7_file` script file will split the `tk7` file into segments of 498 records each. Each segment will be appended to a segment header. The various segments will then be appended to a file header.

Software Unit Name: `cleanup`
Software Unit Type: Unix script file
Software Unit Description: The `cleanup` script file will be executed. The `cleanup` script file will remove any extraneous files still existing as a result of previous data file generation.

4.4.3 System Administration/Database Administration Processes

The ICDB system and database administration scripts have been developed to help maintain the ICDB system. Currently these scripts fall into eight subsystems:

1. Unix reports, which involve scripts that generate Unix file systems reports;
2. database administration reports, which are scripts that generate reports about the database structure and contents;

3. audit reports processes, which are scripts for the managing of audit functions for Oracle;
4. registration processes, which involve scripts for registering users, sites, modems, and data recipients;
5. export manifest communications processes;
6. maintenance processes, which are scripts that help to configure the database;
7. code table distribution processes, which distribute the remac and cargo record states code tables; and
8. help information for the system administrator.

The scope of these procedures is not to provide a complete administration toolset for the ICDB system. There are tools already available on the HP computer that perform many system functions. In addition, `sqldba` is an Oracle tool which performs Oracle startup and process management functions.

The scope of this code is not to duplicate existing capabilities but to provide a set of Unix and Oracle procedures that complement existing toolsets to provide the capability of managing the structure and configuration of the ICDB system.

These scripts are accessible at both the Unix prompt and via a menu interface. For complete installation instructions, see the file `/usr/home/icdb/sys_admin/sys_admin.readme` or see the ICDB Maintenance Manual. Figure 4.37 shows the logical flow for the System Administration/Database Administration subsystem.

4.4.3.1 Audit Reports Subsystem. The Audit Reports subsystem involves the managing of audit functions for Oracle. In order for auditing functions to take effect, however, the database must be shutdown and auditing turned on in the initialization file. Figure 4.38 shows the logical flow for the Audit Reports subsystem. The following software units are components of this subsystem.

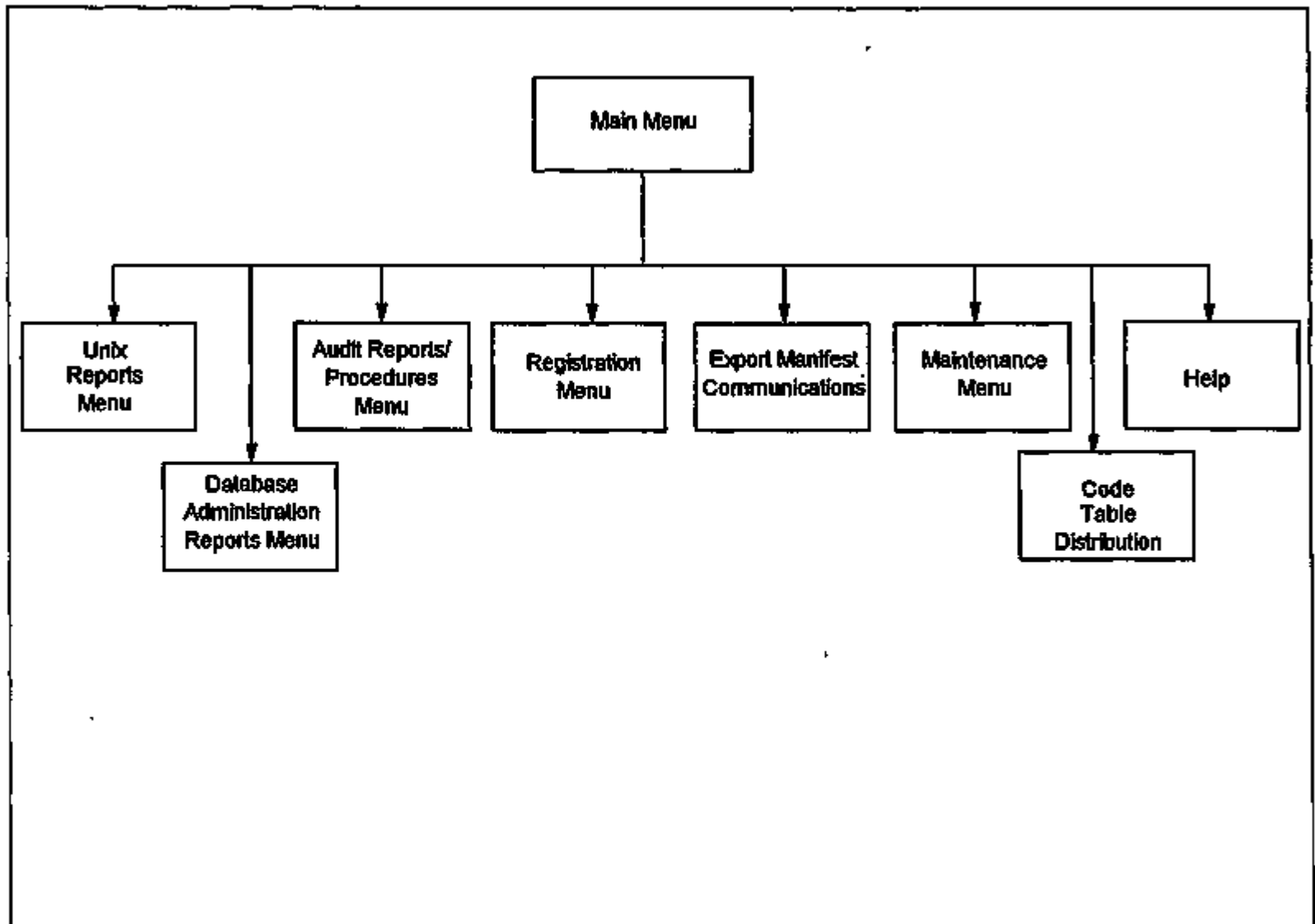


Figure 4.37. The System Administration/Database Administration subsystem.

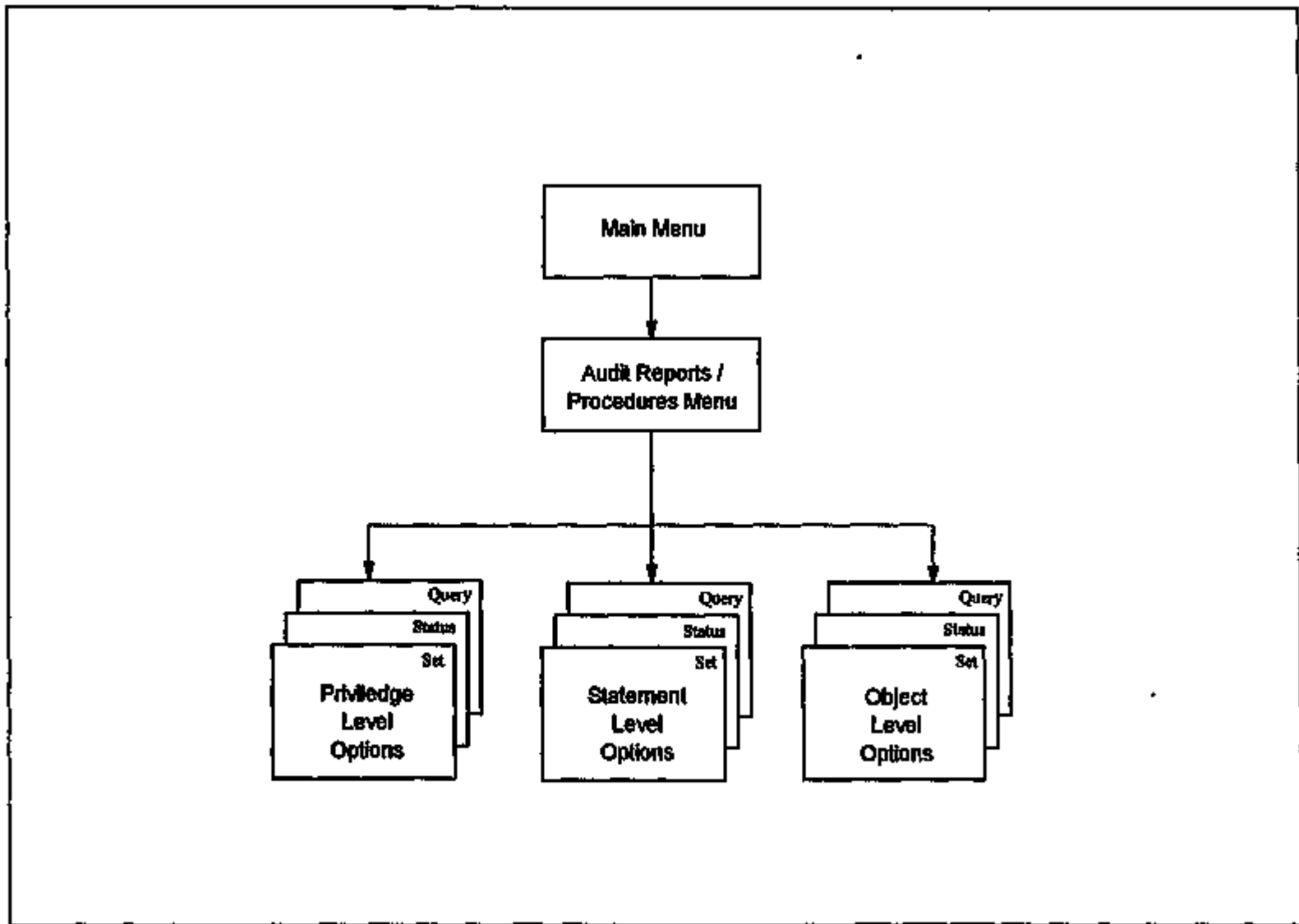


Figure 4.38. Audit Reports subsystem.

Software Unit Name: audit_object_query
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays auditing results from the audit trail due to object-level auditing statements.

Software Unit Name: audit_object_set
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script sets object-level auditing options.

Software Unit Name: audit_object_status
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all object-level auditing options which are currently turned on.

Software Unit Name: audit_privilege_query
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays auditing results from the audit trail due to privilege-level auditing statements.

Software Unit Name: audit_privilege_set
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script sets privilege-level auditing options.

Software Unit Name: audit_privilege_status
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all privilege-level auditing options which are currently turned on.

Software Unit Name: audit_statement_query
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays auditing results from the audit trail due to statement-level auditing statements.

Software Unit Name: audit_statement_set
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script sets statement-level auditing options.

Software Unit Name: audit_statement_status
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all statement-level auditing options which are currently turned on.

4.4.3.2 Database Administration Reports Subsystem. The Database Administration Reports subsystem generates reports about the database structure and contents. Figure 4.39 shows the high-level logical flow for the DBA Scripts subfunction. Figures 4.39-4.43 show the logical flow for subfunctions of the Database Administration Reports subsystem. The following software units are components of this subfunction.

Software Unit Name: dba_check_constraints
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays check constraints owned by a given owner.

Software Unit Name: dba_code_transfer
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays the CODE_TRANSFER file.

Software Unit Name: dba_col_privs
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays grants on columns of objects which are owned by a given user.

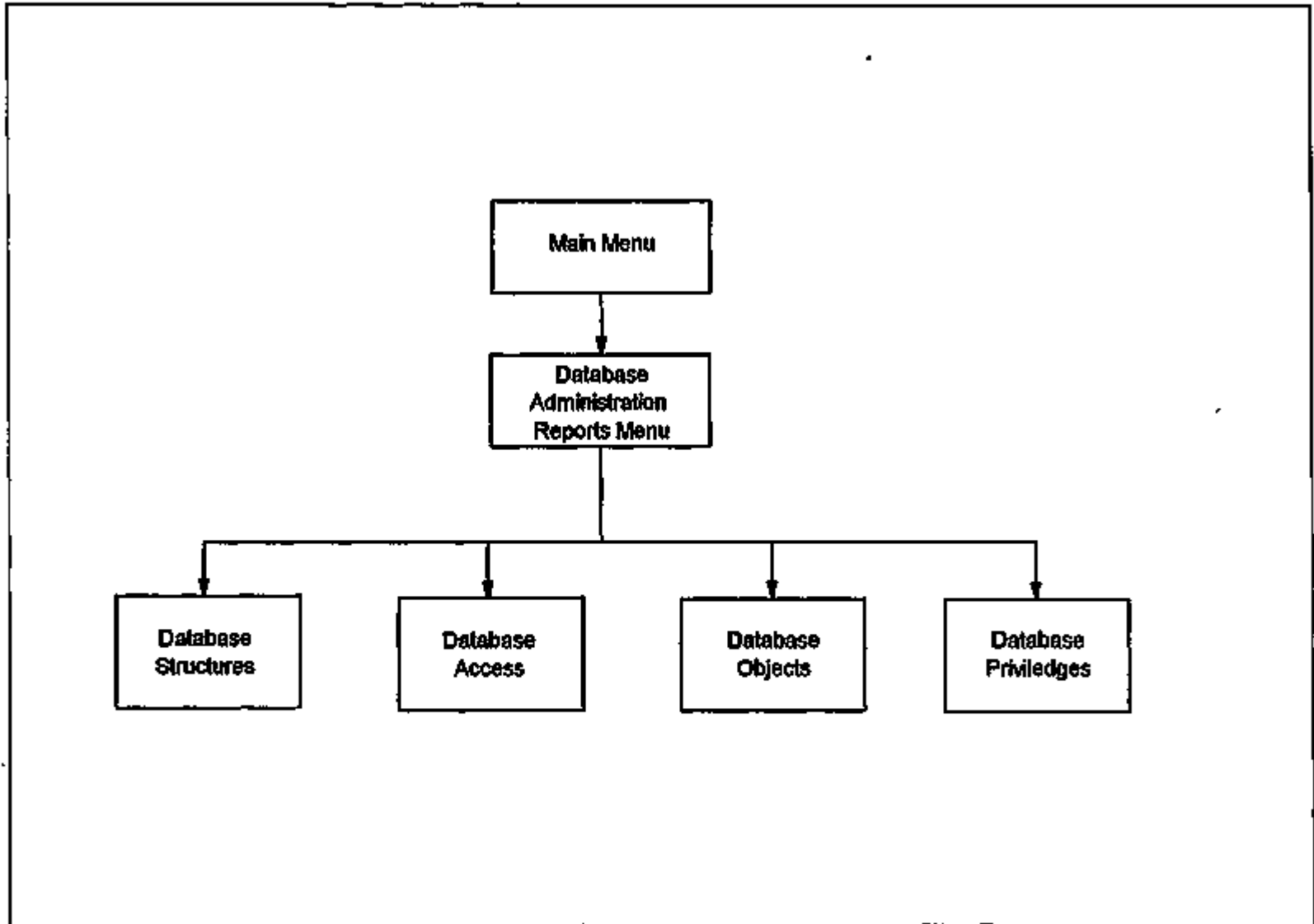


Figure 4.39. The Database Administration Reports subsystem.

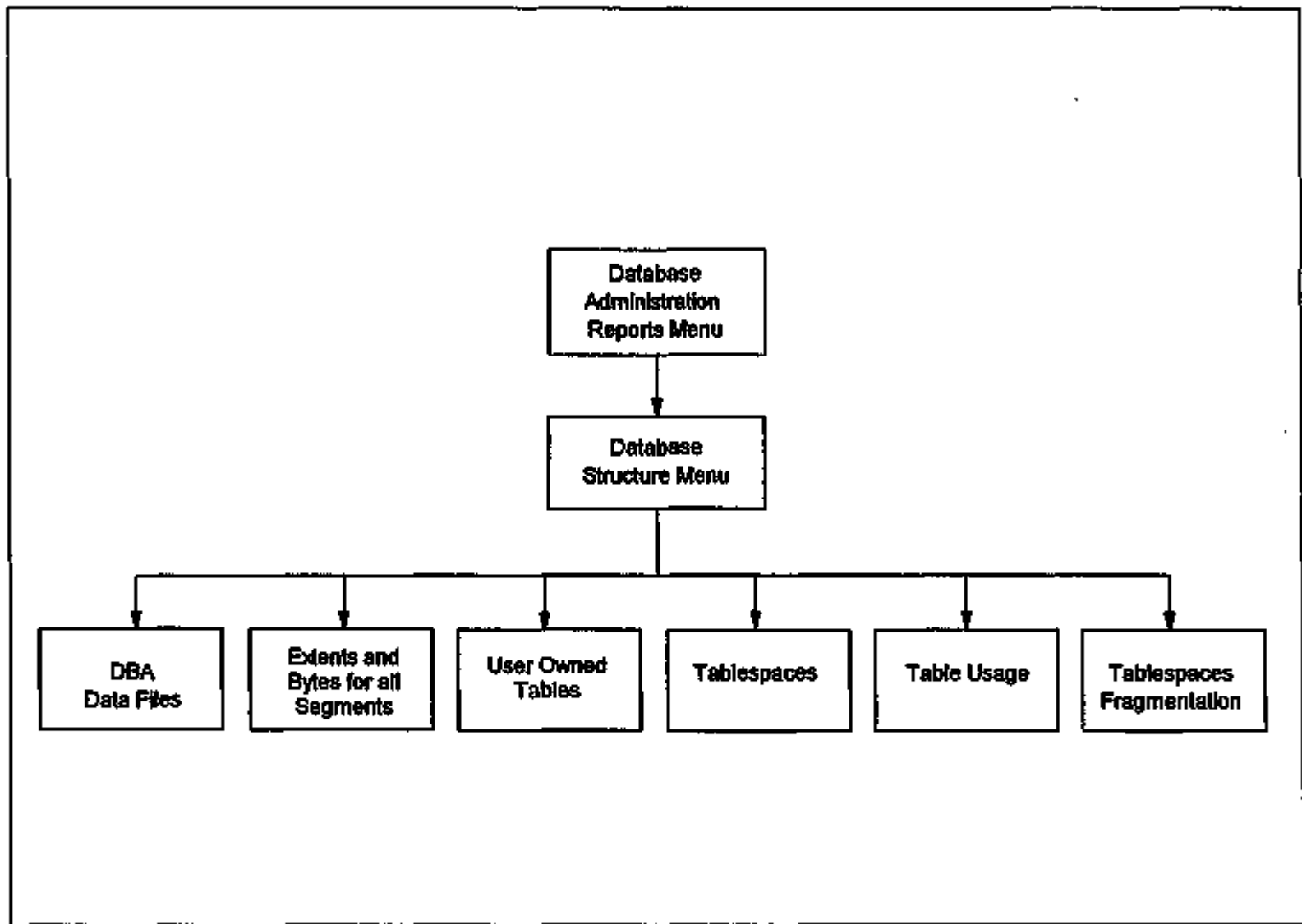


Figure 4.40. The Database Structure subfunctions.

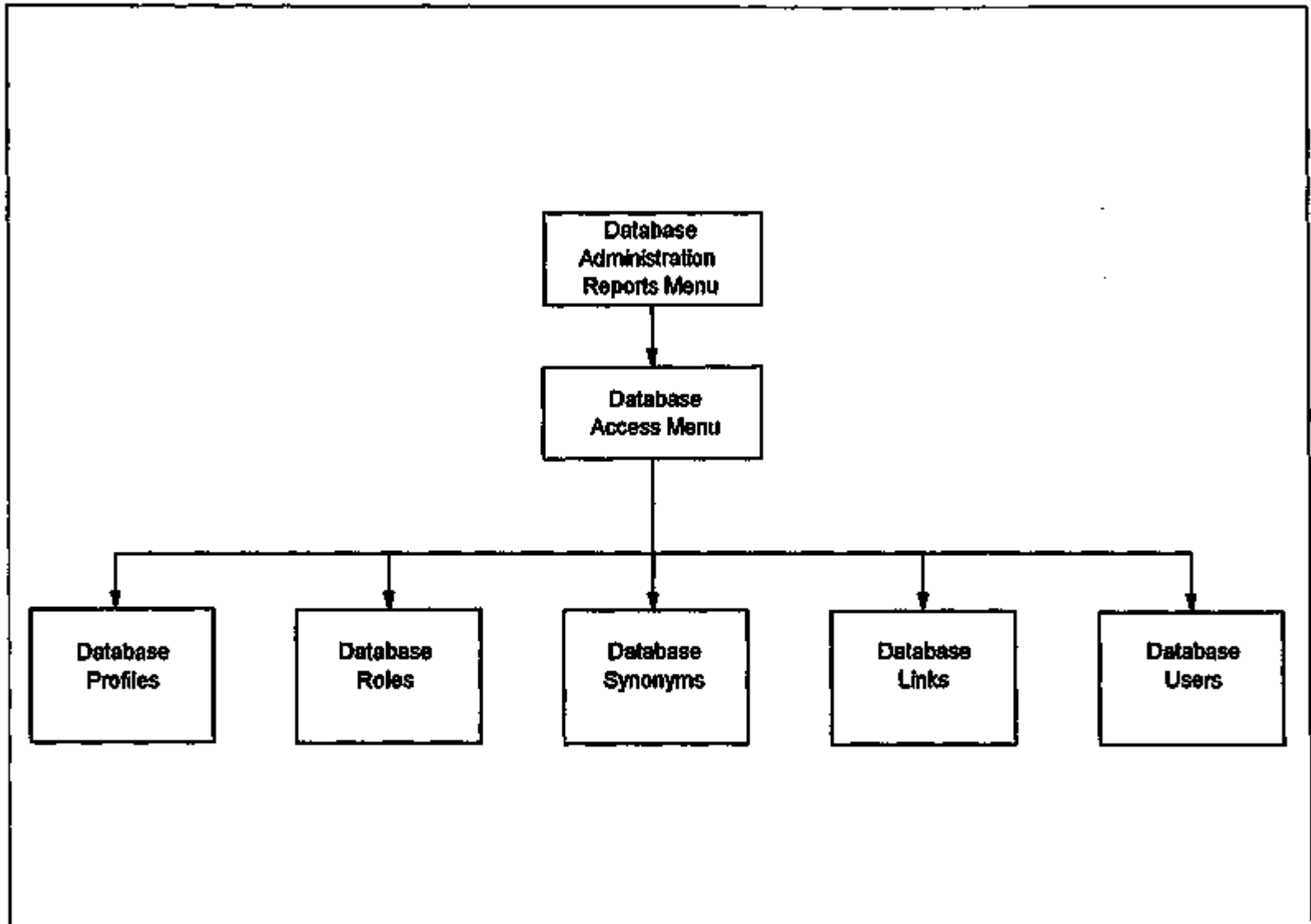


Figure 4.41. The Database Access subfunctions.

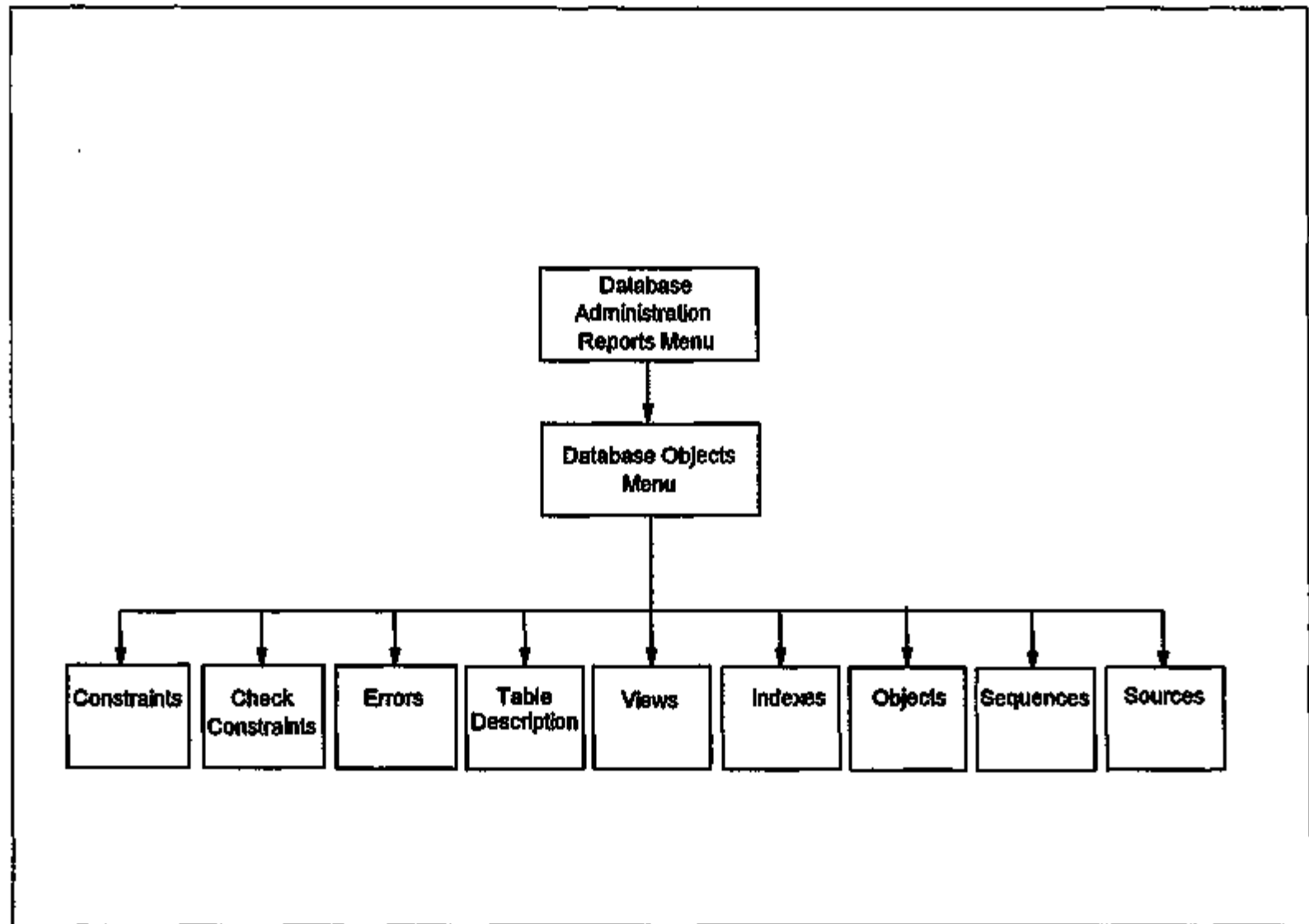


Figure 4.42. The Database Objects subfunctions.

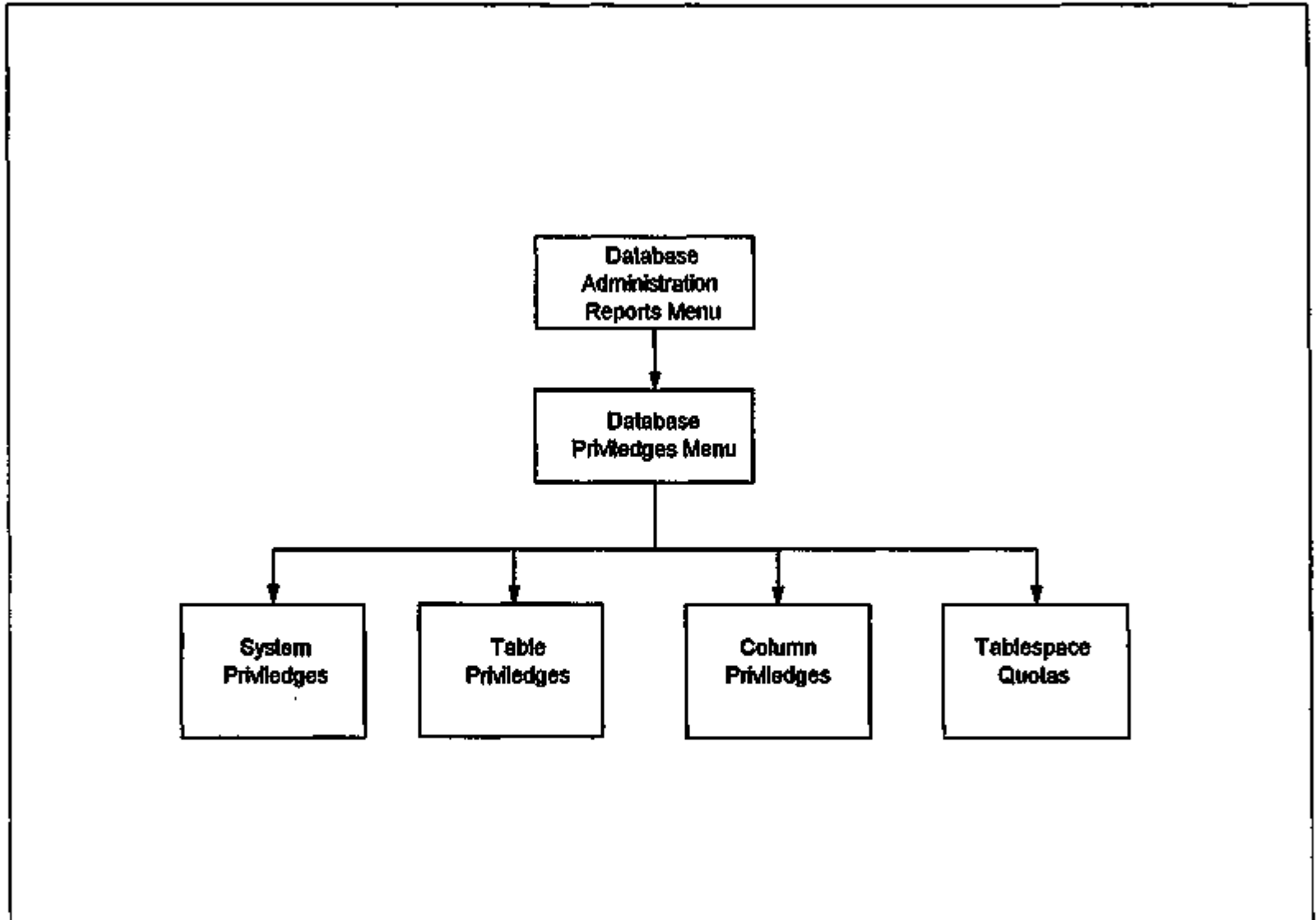


Figure 4.43. The Database Privilege subfunction.

Software Unit Name: dba_constraints
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script prints out all constraints owned by a given owner.

Software Unit Name: dba_datafiles
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays datafile names, status, total storage space, and free space.

Software Unit Name: dba_desc
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays table descriptions of individual tables.

Software Unit Name: dba_errors
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all errors on stored objects owned by a given user.

Software Unit Name: dba_indexes
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script prints out all indexes owned by a given owner.

Software Unit Name: dba_links
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays database links.

Software Unit Name: dba_objects
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all objects owned by a given user.

Software Unit Name: dba_profiles
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all user profiles.

Software Unit Name: dba_readme
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays \$DBA_PATH/./sys_admin.readme

Software Unit Name: dba_roles
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all Oracle roles.

Software Unit Name: dba_rollback_segs
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all rollback segments.

Software Unit Name: dba_segments
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays extents and bytes for all segments owned by a given user.

Software Unit Name: dba_sequences
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all Oracle sequences.

Software Unit Name: dba_source
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays source code for a procedure or function, after displaying all source objects for a given owner.

Software Unit Name: dba_synonyms
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays Oracle synonyms.

Software Unit Name: dba_sys_privs
Software Unit Type: Unix Script/SQL*Plus Script

Software Unit Description: This script displays all system privileges granted.

Software Unit Name: dba_table_privs

Software Unit Type: Unix Script/SQL*Plus Script

Software Unit Description: This script displays all grants on objects owned by a given user.

Software Unit Name: dba_table_usage

Software Unit Type: Unix Script/SQL*Plus Script

Software Unit Description: This script displays extents and percent block usage of all tables owned by a given user.

Software Unit Name: dba_tablespaces

Software Unit Type: Unix Script/SQL*Plus Script

Software Unit Description: This script displays Oracle tablespaces.

Software Unit Name: dba_ts_frag

Software Unit Type: Unix Script/SQL*Plus Script

Software Unit Description: This script displays tablespace fragmentation.

Software Unit Name: dba_tables

Software Unit Type: Unix Script/SQL*Plus Script

Software Unit Description: This script prints out selected tables owned by a given user.

Software Unit Name: dba_ts_quotas

Software Unit Type: Unix Script/SQL*Plus Script

Software Unit Description: This script displays all tablespace quotas.

Software Unit Name: dba_users

Software Unit Type: Unix Script/SQL*Plus Script

Software Unit Description: This script displays all users in the database.

Software Unit Name: dba_version
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays Oracle version numbers.

Software Unit Name: dba_views
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays the text of all views owned by a given user.

4.4.3.3 Maintenance Subsystem. The Maintenance subsystem helps to configure the database. Figure 4.44 shows the logical flow for the Maintenance subsystem. The following software units are components of this subsystem.

Software Unit Name: maint_compare_tables
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script compares tables owned by different owners to determine any differences in table structure. This comparison is helpful when testing tables need to be inserted into a production environment. This script does not check constraints, data, indexes or grants.

Software Unit Name: maint_cre_public_syn
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script creates public synonyms of selected objects owned by a given user.

Software Unit Name: maint_cre_public_syn
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script creates public synonyms of selected objects owned by a given user.

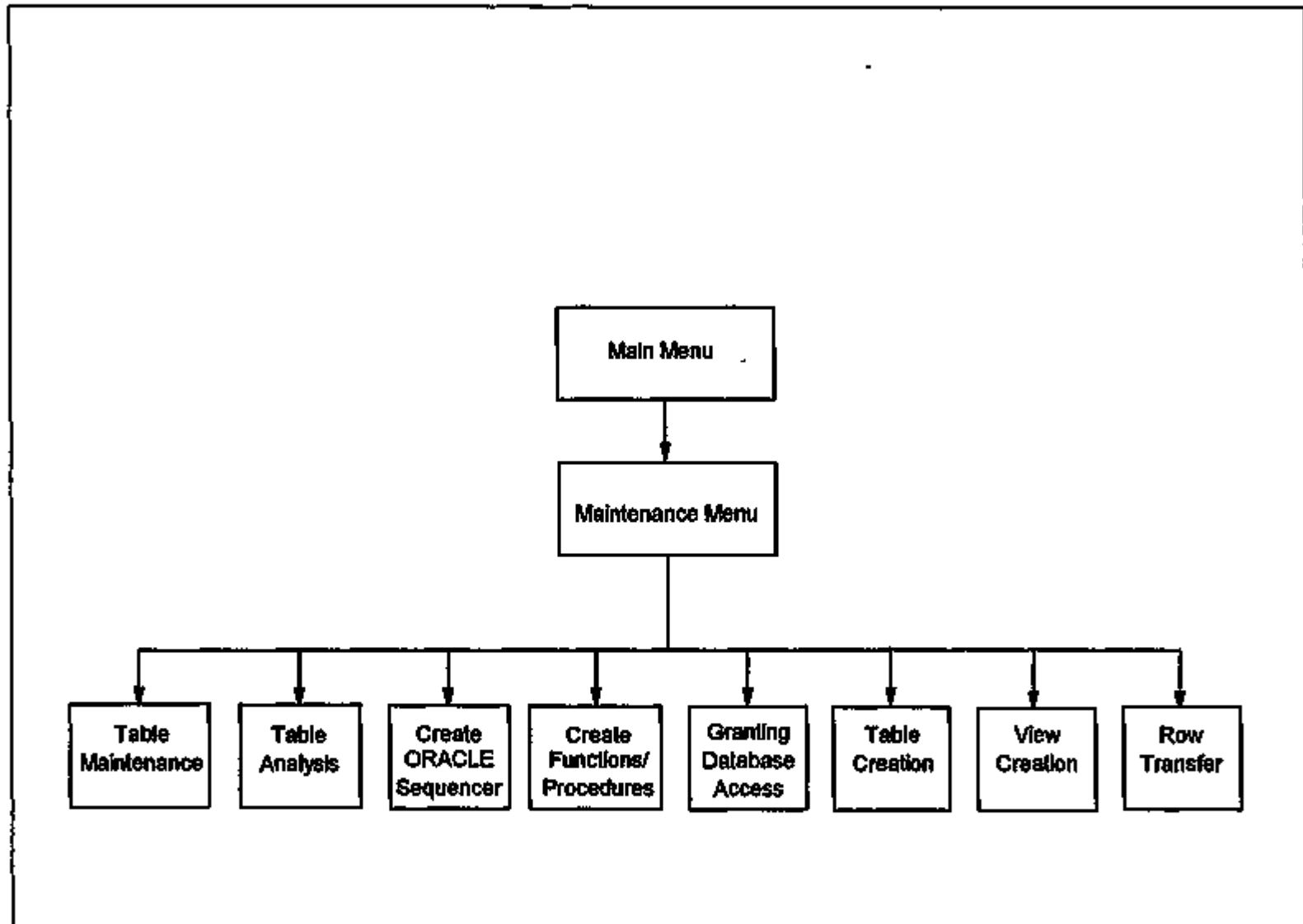


Figure 4.44. The Maintenance subsystem.

Software Unit Name: maint_drop_public_syn
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script drops public synonyms of selected objects owned by a given user.

Software Unit Name: maint_gen_sequences
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script generates a SQL create script for all sequences owned by a given user.

Software Unit Name: maint_gen_source
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script generates a script file for functions/procedures owned by a given user. These scripts will be placed in the current directory, so it is advised that this script be run in an empty directory.

Software Unit Name: maint_gen_tables
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script generates a SQL create script for all tables owned by a given user.

Software Unit Name: maint_gen_views
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script generates a SQL create script for all views owned by a given user.

Software Unit Name: maint_grant_access
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script grants access to objects in a given schema to a single user.

Software Unit Name: `maint_revoke_access`
Software Unit Type: Unix Script/SQL*Plus Script objects in a given schema.
Software Unit Description: This script revokes access to objects in a given schema from a single user.

Software Unit Name: `maint_schema_report`
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script stores in system files the schema reports on tables which are selected based on group codes found in the `tab_group` table (selects from CASE dictionary tables).

Software Unit Name: `maint_transfer_rows`
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script truncates all tables selected for a particular user and copies rows from tables with the same name owned by a different user.

Software Unit Name: `maint_validate_tables`
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script runs the analyze table command for all tables owned by a given user.

4.4.3.4 Unix Reports Subsystem. The Unix Report subsystem generates Unix file system reports. Figure 4.45 shows the logical flow for the Unix Reports subsystem. The following software units are components of this subsystem.

Software Unit Name: `sys_admin`
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script provides menu access using an ansi terminal driver.

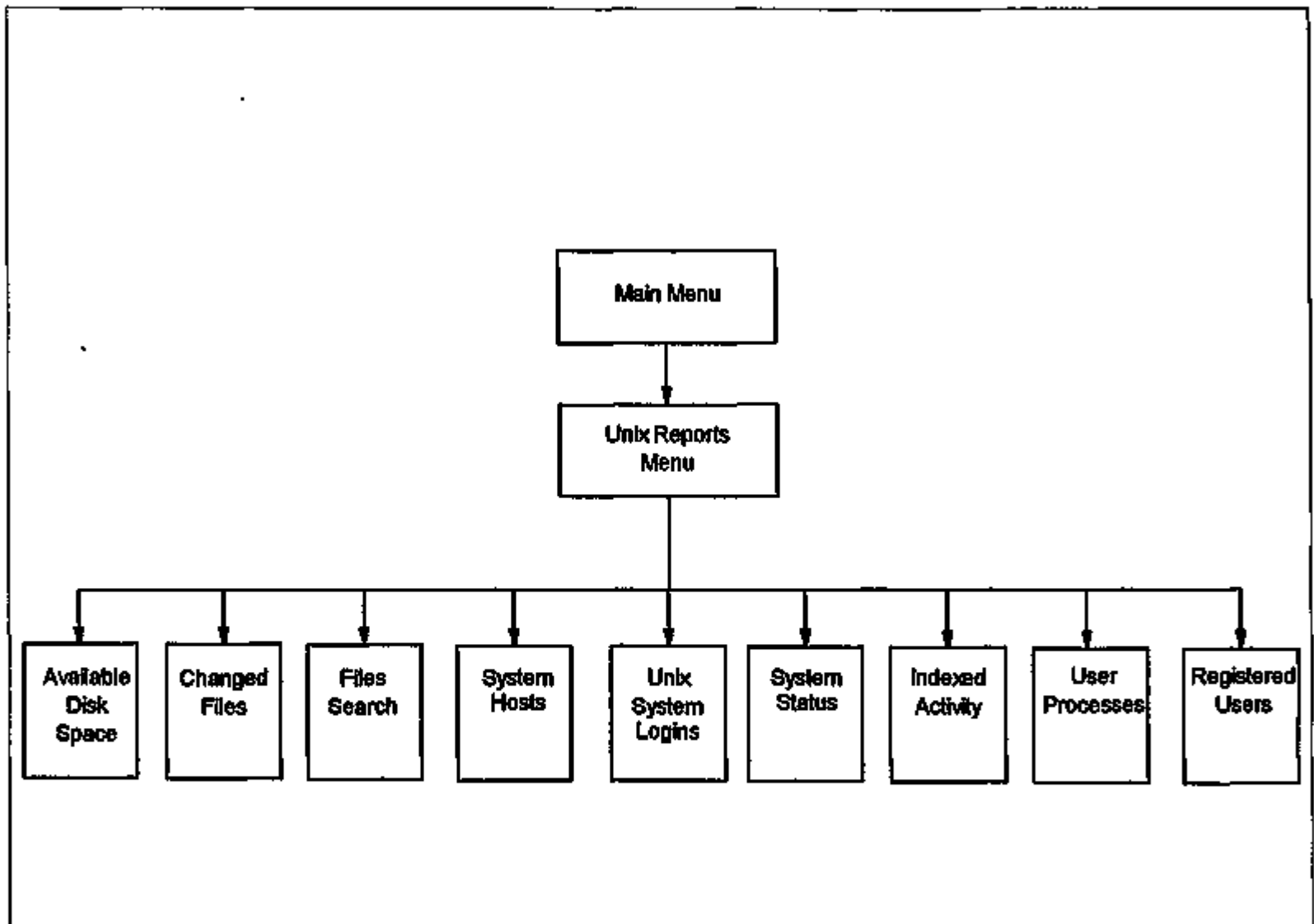


Figure 4.45. The Unix Reports subsystem.

Software Unit Name: sys_bdf
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays disk-free space for the Unix file system.

Software Unit Name: sys_change_case
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script changes the case of file names from upper case to lower case.

Software Unit Name: sys_files_changed
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all files changed in the past <n> days.

Software Unit Name: sys_find_files
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script searches for files matching <expr>.

Software Unit Name: sys_hosts
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays host machines registered.

Software Unit Name: sys_logins
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all users logged on to the system.

Software Unit Name: sys_morning_report
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays extensive information concerning the state of the network, Unix filesystem, ORACLE databases and data. This should be monitored each morning.

Software Unit Name: sys_netstat
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all active Internet connections.

Software Unit Name: sys_processes
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays processes for a given user.

Software Unit Name: sys_users
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script displays all users registered.

4.4.3.5 Registration Subsystem. Figure 4.46 shows the logical flow for the Registration subsystem. This subsystem allows DBAs and system administrators(SAs) to register system users, sites, modems, and data extract recipients. The following software units are components of this subsystem.

Software Unit Name: site_registration.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows DBAs and SAs to register port sites and hubs that are part of the regional system. They may also change a site status from disabled to enabled (or vice versa).

Software Unit Name: dist_sites.sh
Software Unit Type: Unix script
Software Unit Description: This script is used to distribute added sites and site information to the Hubs.

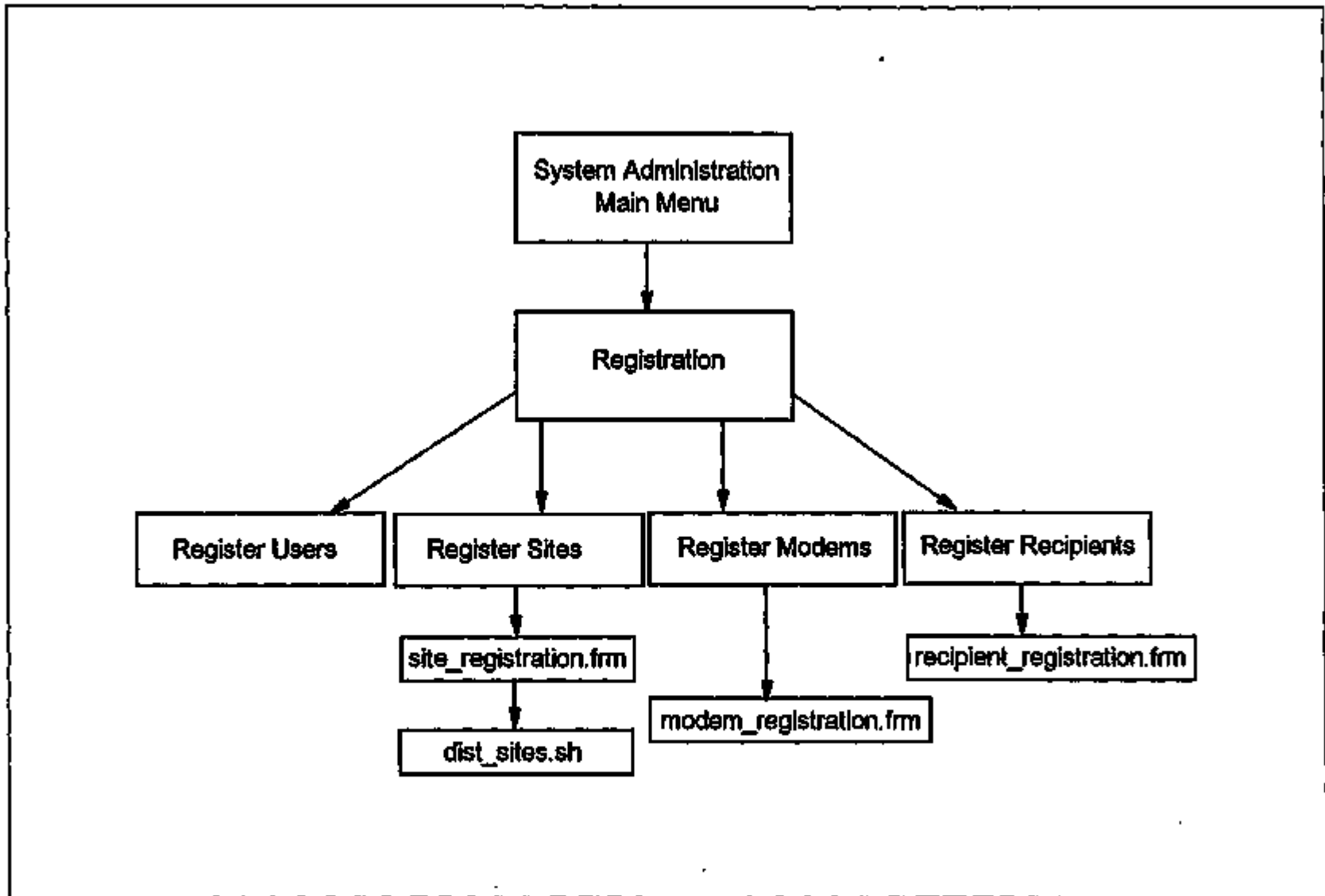


Figure 4.46. The Registration subsystem.

Software Unit Name: modem_registration.frm
Software Unit Type: SQL*Form
Software Unit Description: This form is used to distribute modem information for added modems to all the Hubs.

Software Unit Name: recipient_registration.frm
Software Unit Type: SQL*Form
Software Unit Description: This form is used to distribute recipient information used for data extracts for added recipients to all the Hubs.

The Register Users Subfunction

Figure 4.47 shows the logical flow for software units that are components of the Register User Subfunction. These units allow DBAs and SAs to add/edit users, delete users, change a user's password, or change a user's role. The following software units are components of this subfunction.

Software Unit Name: user_registration.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows DBAs and SAs to add new users to the region or to edit certain information about new users. The user's password and the user's role are not editable fields on this screen.

Software Unit Name: user_add_user
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script is called from the user registration form to add new regional users to the Server and the Hubs.

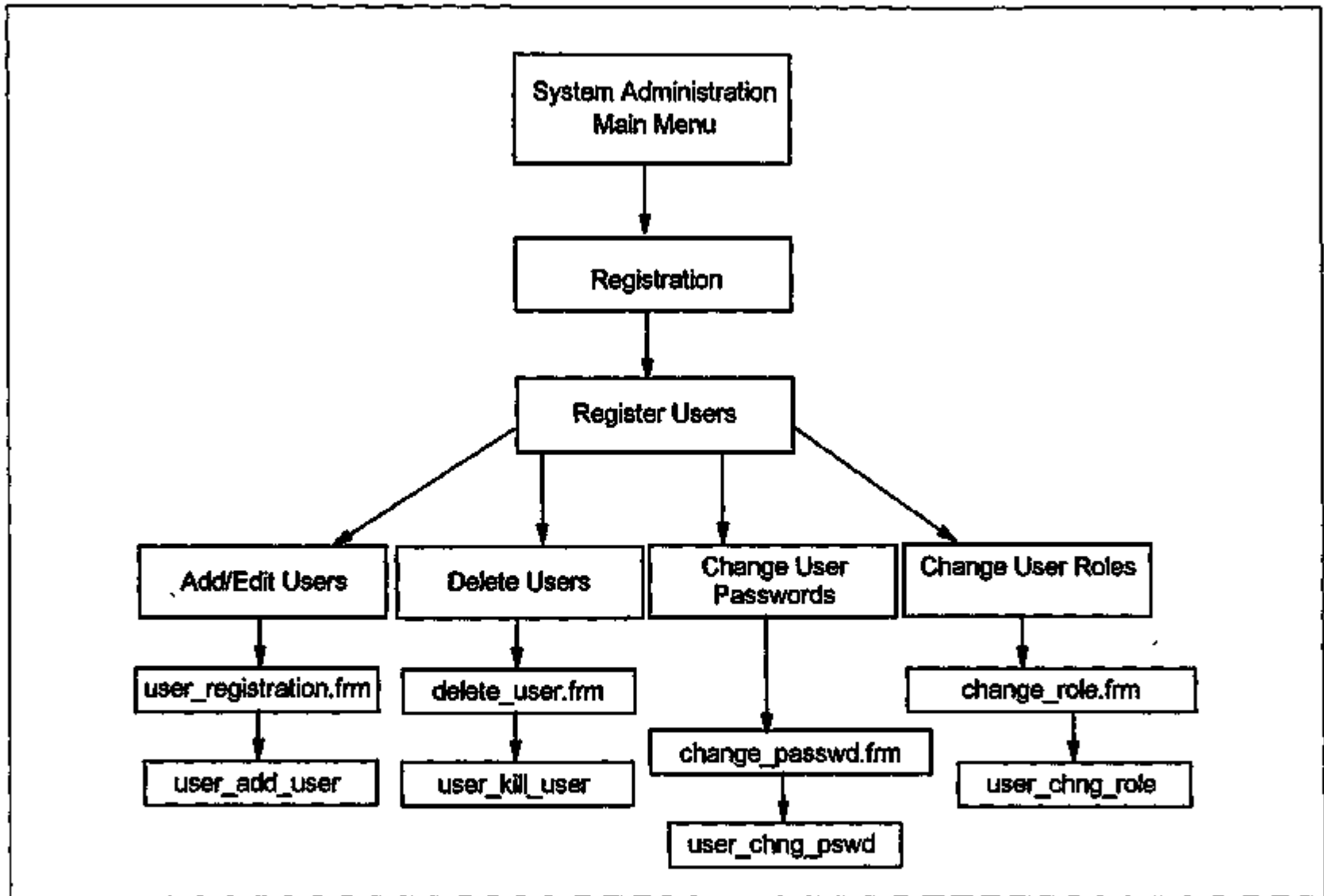


Figure 4.47. The User Registration subsystem.

Software Unit Name: delete_user.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows DBAs and SAs to identify users to delete on the Server and the Hubs.

Software Unit Name: user_kill_user
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script is called from the delete user form to delete users from the regional Server and the Hubs.

Software Unit Name: change_passwd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows DBAs and SAs to identify a changed password for existing users.

Software Unit Name: user_chg_pswd
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script is called from the change password form to change users passwords on the Server and the Hubs.

Software Unit Name: change_role.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows DBAs and SAs to identify a changed role for existing users.

Software Unit Name: user_chg_role
Software Unit Type: Unix Script/SQL*Plus Script
Software Unit Description: This script is called from the change role form to change users roles on the Server and the Hubs.

Export Manifest Communications Subsystem

Figure 4.48 shows the logical flow for the software units that are components of the Export Manifest Communications Subsystem. This subsystem is currently being developed for the regional system. More information about these individual software units must be supplied by developers of the WPS terminal-level system.

The Code Table Distribution Subsystem

The Code Table Distribution Subsystem was added late in the development process for the regional system. It works like the ICDB user interface code table distribution processes. However, two code tables - termac and cargo record status code - have been identified as tables that only a DBA or SA should be able to distribute. Figure 4.49 shows the logical flow for software units that are components of this subsystem.

4.4.3.6 Menu Interface Subsystem. A menu interface is provided for use with the ICDB System and Database Administration scripts. The following files contain the complete menu application.

Software Unit Name:	sysadmin.dmm
Software Unit Type:	SQL*Menu50 Executable
Software Unit Description:	This is an Oracle Menu executable which provides access to sys_admin scripts via a menu interface.

Software Unit Name:	sysadmin.sql
Software Unit Type:	SQL*Plus Script
Software Unit Description:	This is an unloaded script file which contains the menu application sysadmin.

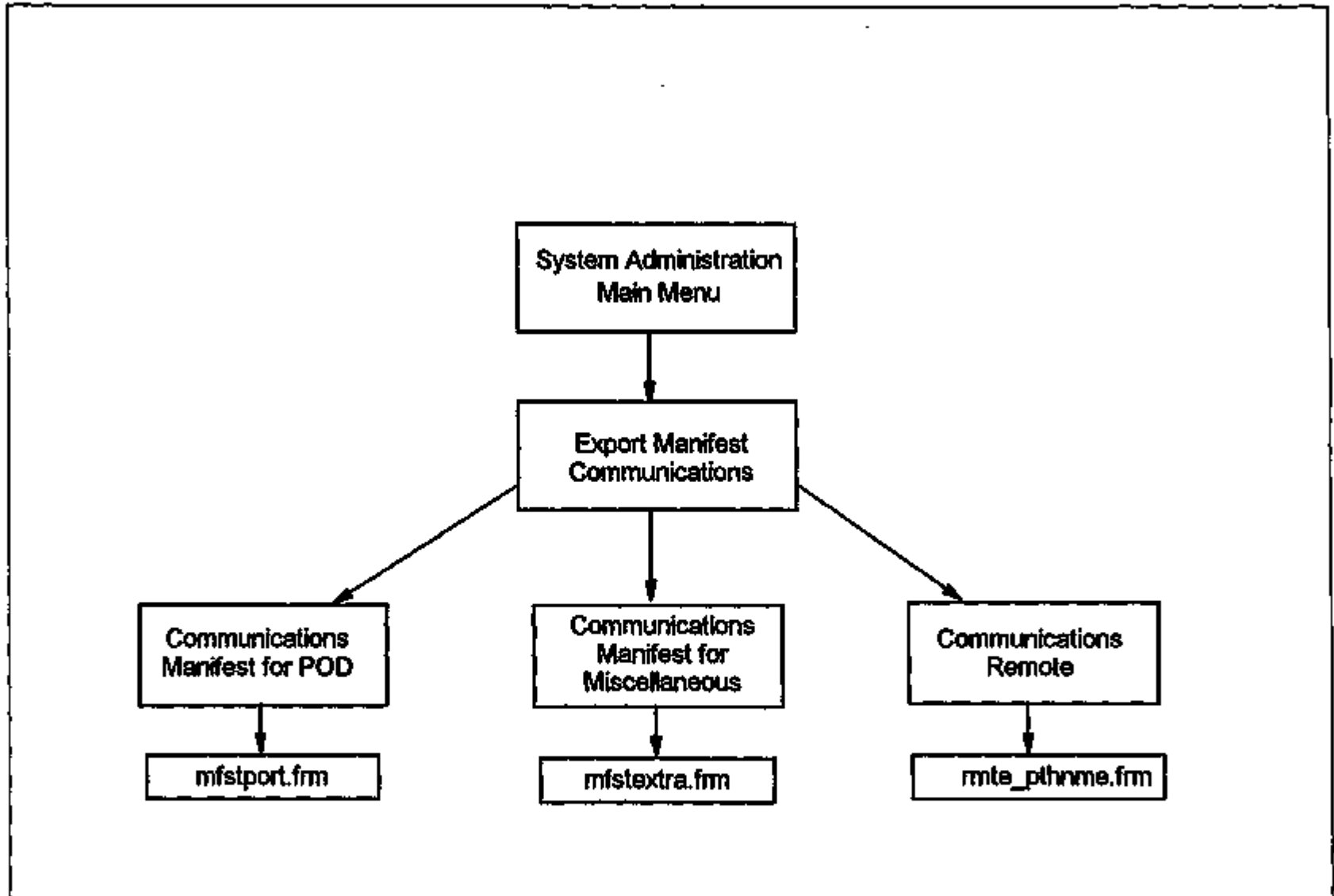


Figure 4.48. The Export Manifest Communications subsystem.

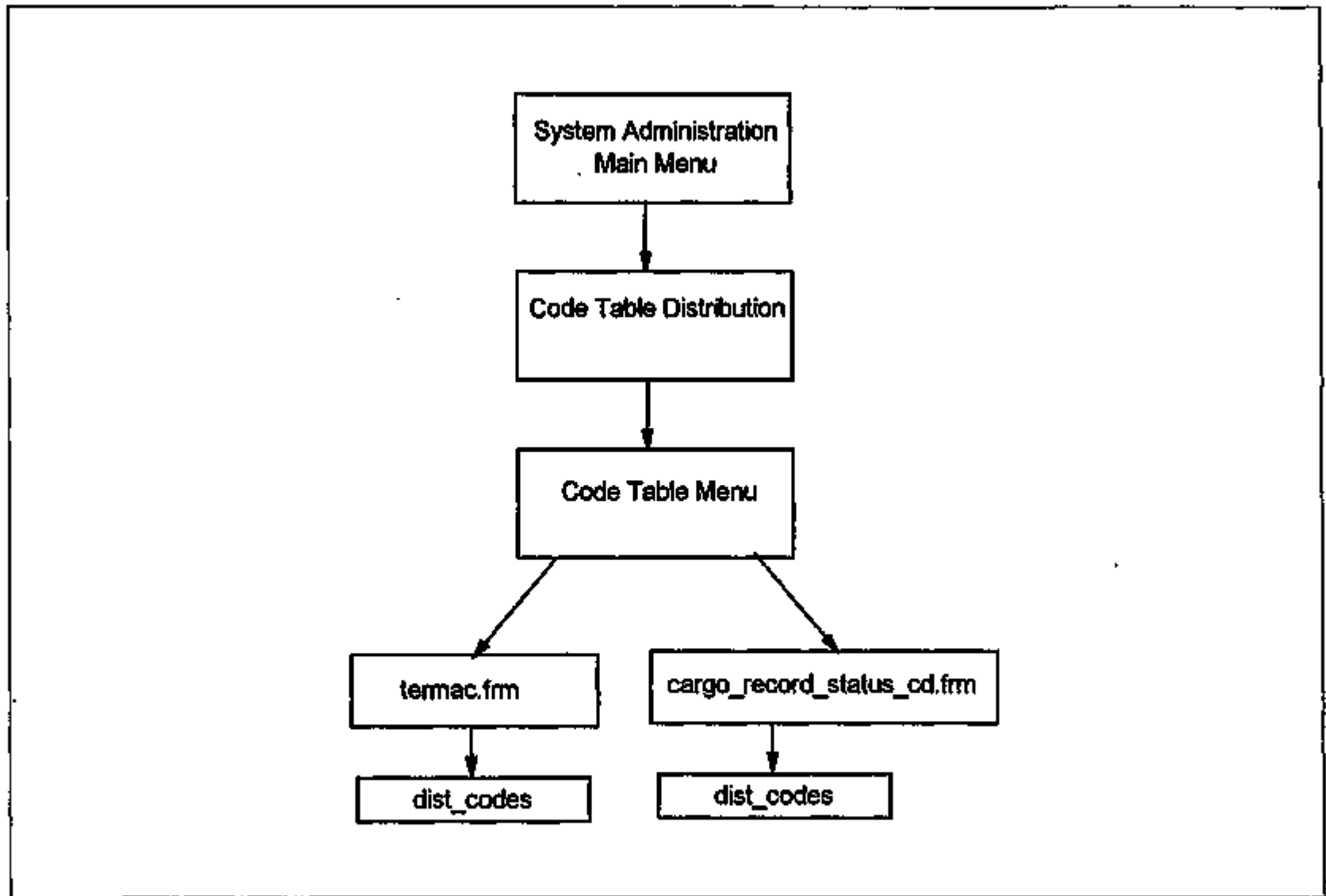
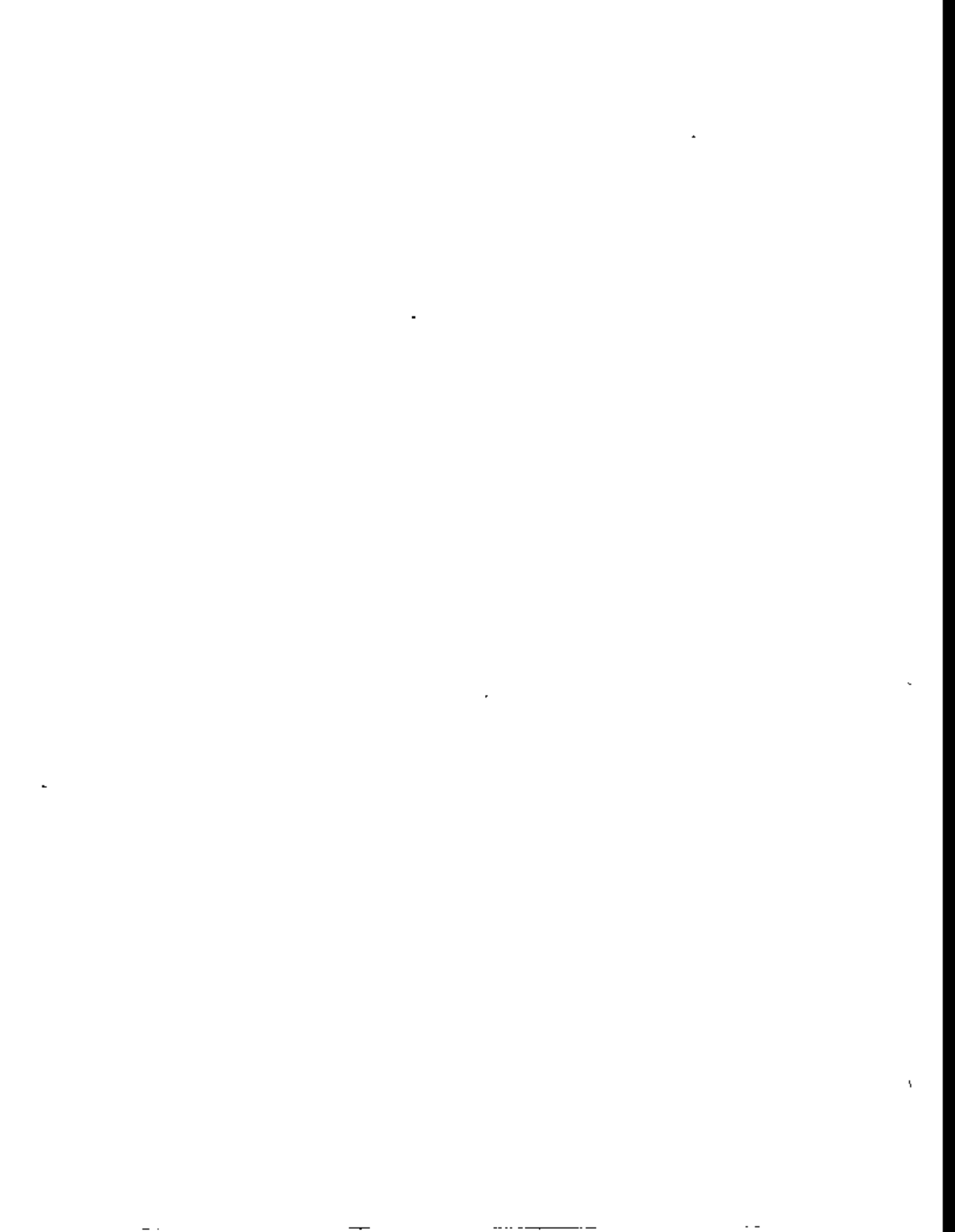


Figure 4.49. The Code Table Distribution subsystem.

Software Unit Name: termac.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the TERMAC table. A UNIX script is then called which distributes the changes to the WPS sites.

Software Unit Name: cargo_record_status_cd.frm
Software Unit Type: SQL*Form
Software Unit Description: This form allows the Data Administrator to query, update, insert, or delete records from the CARGO RECORD STATUS CODE table.

This concludes the descriptions of the System Administration/Database Administration processes.



INTERNAL DISTRIBUTION

- | | | | |
|--------|------------------|--------|----------------------------|
| 1. | D. Alvic | 13. | D. E. Reichle |
| 2. | M. S. Bronzini | 14. | R. B. Shelton |
| 3. | G. E. Courville | 15. | P. C. Shipe |
| 4. | E. Z. Faby | 16-19. | L. F. Truett |
| 5. | R. A. Ferguson | 20. | Central Research Library |
| 6. | J. Fluker | 21. | Document Reference Section |
| 7. | J. W. Grubb | 22-23. | Laboratory Records |
| 8. | S. G. Hildebrand | 24. | Laboratory Records - RC |
| 9. | S. Moore | 25. | ORNL Patent Office |
| 10-12. | J. P. Rollow | | |

EXTERNAL DISTRIBUTION

26. Ms. Betty Childers, ICDB Product Management Office, Military Traffic Management Command, 5611 Columbia Pike, Falls Church, VA 22041-5050.
27. Mr. Herb Kaskoff, WPS Project Management Office, Military Traffic Management Command, 5611 Columbia Pike, Falls Church, VA 22041-5050.
28. Dr. Thomas E. Drabek, Professor, Department of Sociology, University of Denver, Denver, CO 80208-0209.
- 29-31. Mr. Jack Hetman, ICDB Product Management Office, Military Traffic Management Command, 5611 Columbia Pike, Falls Church, VA 22041-5050.
32. Mr. Robert Loy, Ogden Professional Services, 3211 Jermantown Road, Fairfax, VA, 22030.
33. Mr. Calvin D. MacCracken, President, Calmac Manufacturing Corporation, 101 West Sheffield Avenue, Englewood, New Jersey 07631.
34. Mr. Bob Porter, G3 Future Systems, Office of the Assistant Deputy Chief of Staff for Operations, Military Traffic Management Command, 5611 Columbia Pike, Falls Church, VA 22041-5050.
35. Ms. Lisa Seery, MTIM-B, Military Traffic Management Command, 5611 Columbia Pike, Falls Church, VA 22041-5050.
36. Mr. George F. Sowers, Senior Vice President, Law Companies Group, Inc., 114 Townpark Drive, Suite 250, Kennesaw, GA 30144-5599.
37. Mr. Dave Terry, G3 Future Systems, Office of the Assistant Deputy Chief of Staff for Operations, Military Traffic Management Command, 5611 Columbia Pike, Falls Church, VA 22041-5050.
38. Dr. C. Michael Walton, Paul D. and Betty Robertson Mead Centennial, Professor and Chairman, Department of Civil Engineering, College of Engineering, The University of Texas at Austin, Cockrell Hall, Suite 4.2, Austin, TX 78712.
39. Office of Assistant Manager for Energy Research and Development, DOE-ORO, P.O. Box 2001, Oak Ridge, Tennessee 37831-8600.
- 40-41. Office of Scientific and Technical Information, U.S. Department of Energy, P.O. Box 62, Oak Ridge, Tennessee 37831.

