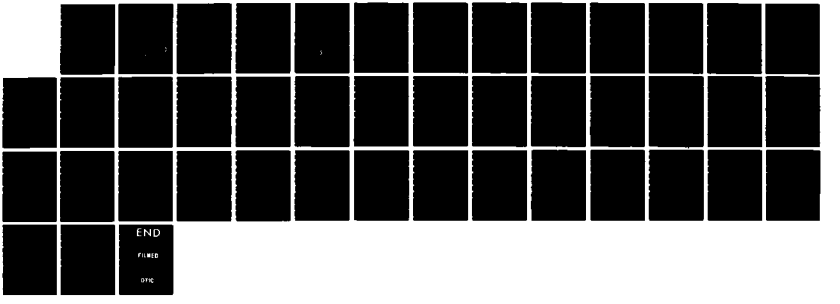
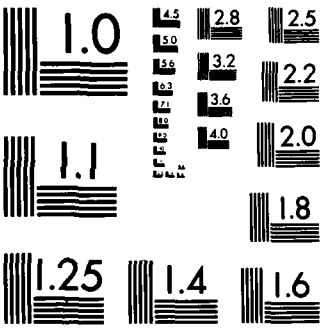
AD-A158 455 SPECIFICATIONS FOR THREE MEMBERS OF THE MILITARY MESSAGE SYSTEM (MMS) FAMILY(U) NAVAL RESEARCH LAB WASHINGTON DC C L HEITMEYER ET AL. 09 SEP 85 1/1

UNCLASSIFIED NRL-MR-5645 F/G 17/2 NL





(2)

NRL Memorandum Report 5645

# Specifications for Three Members of the Military Message System (MMS) Family

C. L. HEITMEYER AND M. R. CORNWELL

*Computer Science and Systems Branch*
*Information Technology Division*



September 9, 1985






NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE

AD-A158 455

## REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b RESTRICTIVE MARKINGS |
| 2a SECURITY CLASSIFICATION AUTHORITY | 3 DISTRIBUTION / AVAILABILITY OF REPORT<br>Approved for public release; distribution unlimited. |
| 2b DECLASSIFICATION / DOWNGRADING SCHEDULE | |
| 4 PERFORMING ORGANIZATION REPORT NUMBER(S)<br>NRL Memorandum Report 5645 | 5 MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a NAME OF PERFORMING ORGANIZATION<br>Naval Research Laboratory<br>6b OFFICE SYMBOL (If applicable)<br>Code 7596 | 7a NAME OF MONITORING ORGANIZATION |
| 6c ADDRESS (City, State, and ZIP Code)<br>Washington, DC 20375-5000 | 7b ADDRESS (City, State, and ZIP Code) |
| 8a NAME OF FUNDING / SPONSORING ORGANIZATION<br>NSWSC<br>8b OFFICE SYMBOL (If applicable)<br>Code 8144 | 9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c ADDRESS (City, State, and ZIP Code)<br>Washington, DC 20363-5001 | 10 SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.<br>DN880-204 |

11 TITLE (Include Security Classification)

Specifications for Three Members of the Military Message System (MMS) Family

12 PERSONAL AUTHOR(S)
Heitmeyer, C.L. and Cornwell, M.R.

| 13a TYPE OF REPORT | 13b TIME COVERED | 14 DATE OF REPORT (Year, Month, Day) | 15 PAGE COUNT |
|---|---|---|---|
| Interim | FROM 11/81 TO 5/85 | 1985 September 9 | 42 |

16 SUPPLEMENTARY NOTATION

| 17 COSATI CODES | | | 18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | Computer security, Requirements documentation, Message systems, Software engineering, (Continues) |
| | | | |
| | | | |

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This report presents informal specifications for the user commands supported by three message systems, all members of the Military Message System (MMS) family. The message systems constitute a family because it is worthwhile to study their similarities before studying their differences. The members of the MMS family share a common definition of security. This report is intended to be an example of both good requirements documentation and an appropriate design for a secure message system.

| 20 DISTRIBUTION / AVAILABILITY OF ABSTRACT | 21 ABSTRACT SECURITY CLASSIFICATION | |
|---|---|---|
| ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS | UNCLASSIFIED | |
| 22a NAME OF RESPONSIBLE INDIVIDUAL<br>Mark R. Cornwell | 22b TELEPHONE (Include Area Code)<br>(202) 767-3365 | 22c OFFICE SYMBOL<br>Code 7596 |

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

18. SUBJECT TERMS (Continued)

Specifications
Office automation
Program families

CONTENTS

# SPECIFICATIONS FOR THREE MEMBERS OF THE MILITARY MESSAGE SYSTEMS (MMS) FAMILY

## Introduction

Contained herein are informal specifications for the user commands supported by three message systems, namely, **M0, M1,** and **M2,** all members of the Military Message System (MMS) family. We call the specifications **informal** because the semantics of each user command, i.e., the command's user-visible effects, are described in English prose. A future report will provide more formal specifications.

This document is divided into four sections: Section 1 provides an overview of the MMS family and the three family members; Section 2 lists the user commands; Section 3 contains specifications for the user commands; and Section 4 provides a glossary.

## 1. Overview of MMS Family and Three Family Members

After introducing MMS terminology, this section describes the Intermediate Command Language (ICL) and its role in the specifications; the relationship between the specifications and the MMS security model [1]; and specific requirements of the three MMS family members.

### 1.1. MMS Terminology

A MMS user issues commands to log into and out of the system and to perform various operations on MMS data items. There are two major classes of MMS data items: users and entities. A **user** is a person authorized to use the MMS. Associated with each user are three attributes: the user's **clearance,** a set of **authorized roles** (e.g., downgrader, releaser, system security officer), and a set of **current roles.** Examples of **entities** are messages and message files. Associated with each entity are four attributes: the entity **classification** (e.g., SECRET), its **value,** its **data type** (which determines the kinds of operations that can be applied to the entity), and its **access set** (which describes the operations that specified users may perform on the entity). Some entities, called **containers,** may have an additional attribute, called **Container Clearance Required (CCR),** that requires any user who wishes to view information in the container to have a clearance greater than or equal to the classification of the container. In a MMS, a **userID** is a name for a user and a **reference** is a name for an entity.

Each MMS user is assigned a special message file, called an **inbox,** in which the MMS places an **entry** for each message sent to the user. An entry contains the message and message status information (e.g., whether the message is "new", whether the message was received "for-action", "for-info", etc.). When the user displays his inbox or any other message file, only part of each message entry in the file is displayed. The part consists of selected message fields (e.g., the Subject and To fields of the message) and the message status information.

Every message is either a draft message or a sent message. A **draft message** is a message in draft form; users create and edit draft messages. A **sent message** is a message that has been released, i.e., sent to one or more other users. By issuing the command SEND_MSG, the user converts a draft message into a sent message. A user may send messages to local users, i.e., users on the same MMS, or to remote users, i.e., users on different message systems. For each remote user to whom a message is sent, the MMS may transmit a copy of the message over the appropriate network. For each local user, the MMS creates an entry for the message and inserts the entry into the user's inbox.

Every message has a **message type.** Many family members support two message types: **formal** messages and **informal** messages. The message type is determined at the time the message is created. When a user issues the SEND_MSG command, thus converting a message from a draft to a sent message, the message type does not change. Messages of two different message types can differ (1) in the set of fields that they contain and (2) in the set of ICL commands that can be applied to them.

Manuscript approved July 1, 1985.

### 1.2. Intermediate Command Language

An important feature of this document is use of an Intermediate Command Language (ICL) to describe the set of functions that MMS family members perform. The ICL is designed so that the functions supported by any single family member can be described by some ICL subset. The ICL subsets associated with M0, M1, and M2 are listed in Section 2.

The specific form a user command takes can vary from family member to another. For example, a user command to display a message may take any one of the following forms:

- the user types the string "display message"
- the user types the string "show message"
- the user selects the menu item "DISPLAY MESSAGE"
- the user depresses a function key labeled "DISPLAY MESSAGE"

Such syntactic differences are not reflected in the ICL. Each ICL command is an **abstract** description of a user command in that it specifies the command's user-visible effects without imposing unnecessary restrictions on either the command syntax or the physical characteristics of the user's terminal. Thus, if the effect of each of the above user commands is identical, i.e., each causes the user terminal to display the specified message, each user command is associated with the same ICL command, namely, DISPLAY_MSG.

The ICL treats the editing of some MMS entities and all access sets in the following special way. To initiate editing, the user issues the command EDIT_XX, where XX is the data type of the item to be edited. EDIT_XX returns a copy of the item, and the user then applies various non-ICL commands (i.e., editing commands) to modify the item as he desires. Once the user is satisfied with the edited version, he issues an UPDATE_XX, which changes the value of the item to that of the edited version. If the user is not satisfied with the edited version, he can omit issuing the UPDATE_XX command, thus retaining the item's original value.

### 1.3. MMS Security Model

A MMS is required to enforce the rules of the MMS security model [1]. Thus a MMS must begin operation in a **secure state,** where secure state is as defined in the model. Each time a user issues an ICL command, one or more changes in system state can occur. However, these state changes can occur, i.e., the ICL command can be completed, only if the constraints of the security model are satisfied. One immediate consequence of enforcing the MMS security model is that each ICL command that displays the value of a MMS entity must also display the entity's classification. Thus, in the specifications below, all ICL commands that display an entity value have at least two output parameters: a value and a classification.

The MMS security model requires that all entities of a given type be treated as either objects or containers. An **object** is the smallest unit in the MMS that has a classification; it contains no other objects and cannot be multilevel. In contrast, a **container** has a classification but may contain objects and/or other containers each with its own classification. In a MMS, the Subject and To fields of a message are usually objects, while messages, message files, and text files are usually containers.

### 1.4. Specific Requirements

The purpose of this report is to provide specifications for three MMS family members. The first, M0, supports the display and storage into message files of incoming messages. The second system, M1, includes all user commands of M0 and additional commands for the composition and transmission of outgoing messages. The third system, M2, is an extension of M1, providing user commands for the system security officer as well as commands for specifying discretionary access controls on messages and other MMS data items.

Indicated in Table 1 are specific requirements of M0, M1, and M2, namely, which entities are containers, which entities are objects, which containers have the CCR attribute, and which message types are implemented. We note that the first three items in the table are part of the

"interpretation" of the security model, since they define the mapping from abstract concepts of the security model to the concrete data items of M0, M1, and M2.

Table 1. Specific Requirements of M0, M1, and M2

| | *M0* | *M1* | *M2* |
|---|---|---|---|
| Which entities are containers? | terminals<br>messages<br>message files<br>message file directory | terminals<br>messages<br>message files<br>message file directory | terminals<br>messages<br>message files<br>message file directory<br>text files<br>text file directory<br>Text field of message<br>message entries |
| Which entities are objects? | all others | all others | all others |
| Are there any non-CCR containers? | no | no | some message files may be non-CCR; all directories and all output devices are non-CCR; all other containers are CCR |
| What message types are supported? | informal | informal | informal and formal |

## 2. ICL Subsets for M0, M1, and M2

This section contains two lists of the ICL commands associated with M0, M1, and M2, one organized by data type, the other by generic command name. Both lists are based on the following seven data types: message, message file message file directory, text file, text file directory, terminal, and user. (Because there are relatively few ICL commands, such as DELETEME and COPYME, that operate on message entries and because each such command can cause a modification in a message file, we include these commands under the data type "message file".) The Appendix contains a third list of the ICL commands organized alphabetically.

### 2.1. Organization by Data Type

The list below indicates the ICL commands associated with each of the three family members and provides the location of each command's specification. In the list, all ICL commands defined on the same data type are grouped together.

*Commands on Messages*

| *M0* | *M1* | *M2* | *Where Specified* |
|---|---|---|---|
| | CREATE_MSG | CREATE_MSG | 3.8.1 |
| | EDIT_MSG | EDIT_MSG | 3.2.4 |
| | UPDATE_MSG | UPDATE_MSG | 3.2.5 |
| DISPLAY_MSG | DISPLAY_MSG | DISPLAY_MSG | 3.2.2 |
| | SEND_MSG | SEND_MSG | 3.8.2 |
| | REPLY_MSG | REPLY_MSG | 3.9.1 |
| | FORINFO_MSG | FORINFO_MSG | 3.9.2 |
| | | PRINT_MSG | 3.2.3 |
| | | FORCOORD_MSG | 3.10.1 |
| | | FORACTION_MSG | 3.11.1 |
| | | FORRELEASE_MSG | 3.10.2 |
| | | READDRESS_MSG | 3.11.2 |
| | | DUP_MSG | 3.7.1 |
| | | DISPLAYAS_MSG | 3.2.7 |
| | | PRINTAS_MSG | 3.2.8 |
| | | EDITAS_MSG | 3.2.9 |
| | | UPDATEAS_MSG | 3.2.10 |

*Commands on Message Files*

| | | | |
|---|---|---|---|
| CREATE_MF | CREATE_MF | CREATE_MF | 3.3.1 |
| DESTROY_MF | DESTROY_MF | DESTROY_MF | 3.2.1 |
| DISPLAY_MF | DISPLAY_MF | DISPLAY_MF | 3.3.2 |
| DELETEME_MF | DELETEME_MF | DELETEME_MF | 3.3.4 |
| UNDELETEME_MF | UNDELETEME_MF | UNDELETEME_MF | 3.3.5 |
| EXPUNGE_MF | EXPUNGE_MF | EXPUNGE_MF | 3.3.6 |
| COPYME_MF | COPYME_MF | COPYME_MF | 3.3.7 |
| MOVEME_MF | MOVEME_MF | MOVEME_MF | 3.3.8 |
| | RECLASSIFY_MF | RECLASSIFY_MF | 3.2.6 |
| | | DUP_MF | 3.3.9 |
| | | PRINT_MF | 3.3.3 |
| | | DISPLAYAS_MF | 3.2.7 |
| | | PRINTAS_MF | 3.2.8 |
| | | EDITAS_MF | 3.2.9 |
| | | UPDATEAS_MF | 3.2.10 |

*Commands on Message File Directories*

| | | | |
|---|---|---|---|
| DISPLAY_MFD | DISPLAY_MFD | DISPLAY_MFD | 3.5.1 |
| | | PRINT_MFD | 3.5.2 |
| | | DISPLAYAS_MFD | 3.2.7 |
| | | PRINTAS_MFD | 3.2.8 |
| | | EDITAS_MFD | 3.2.9 |
| | | UPDATEAS_MFD | 3.2.10 |

*Commands on Text Files*

| *M0* | *M1* | *M2* | *Where Specified* |
|---|---|---|---|
| | | CREATE_TF | 3.4.1 |
| | | DESTROY_TF | 3.2.1 |
| | | EDIT_TF | 3.2.4 |
| | | UPDATE_TF | 3.2.5 |
| | | DISPLAY_TF | 3.2.2 |
| | | PRINT_TF | 3.2.3 |
| | | COPYFRENT_TF | 3.4.2 |
| | | COPYTOENT_TF | 3.4.3 |
| | | RECLASSIFY_TF | 3.2.6 |
| | | DUP_TF | 3.4.4 |
| | | DISPLAYAS_TF | 3.2.7 |
| | | PRINTAS_TF | 3.2.8 |
| | | EDITAS_TF | 3.2.9 |
| | | UPDATEAS_TF | 3.2.10 |

*Commands on Text File Directories*

| | | | |
|---|---|---|---|
| | | DISPLAY_TFD | 3.5.1 |
| | | PRINT_TFD | 3.5.2 |
| | | DISPLAYAS_TFD | 3.2.7 |
| | | PRINTAS_TFD | 3.2.8 |
| | | EDITAS_TFD | 3.2.9 |
| | | UPDATEAS_TFD | 3.2.10 |

*Commands on Terminals*

| | | | |
|---|---|---|---|
| | | CREATE_TERM | 3.6.1 |
| | | DESTROY_TERM | 3.2.1 |
| | | DISPLAY_TERM | 3.6.2 |
| | | PRINT_TERM | 3.6.3 |
| | | RECLASSIFY_TERM | 3.2.6 |
| | | MAXCLAS_TERM | 3.6.4 |

*Commands on Users*

| | | | |
|---|---|---|---|
| | | CREATE_USER | 3.12.1 |
| | | DESTROY_USER | 3.12.2 |
| | | DISPLAY_USER | 3.12.3 |
| | | PRINT_USER | 3.12.4 |
| | | CHGCLEAR_USER | 3.12.5 |
| | | CHGPW_USER | 3.12.6 |
| | | ADDAROLE_USER | 3.12.7 |
| | | RMVAROLE_USER | 3.12.8 |
| | | ADDCROLE_USER | 3.12.9 |
| | | RMVCROLE_USER | 3.12.10 |
| LOGIN_USER | LOGIN_USER | LOGIN_USER | 3.12.11 |
| LOGOUT_USER | LOGOUT_USER | LOGOUT_USER | 3.12.12 |

## 2.2. Organization by Generic Command Name

ICL commands with the same or similar meanings share the same generic command name. In some cases, two commands with the same generic name have identical semantics (e.g., RECLASSIFY_MF and RECLASSIFY_TERM). In other cases, two commands with the same generic name have somewhat different semantics (e.g., DISPLAY_MF and DISPLAY_MFD). Table 2 lists the generic command names, indicates the data types for which each generic command is defined, and provides the location of the specification of the ICL command with the given generic command name and data type. A blank table entry indicates that the generic command is not defined on the given data type.

In Table 2, related generic commands are listed together. Commands may be related because they are inverses (e.g., CREATE and DESTROY), because their semantics are similar (e.g., DISPLAY and PRINT), or because they are usually invoked sequentially (e.g., EDIT and UPDATE).

Table 2: ICL Commands Organized by Generic Command Name

| *COMMAND* | *DATA TYPE* | | | | | | |
|---|---|---|---|---|---|---|---|
| | *MSG* | *MF* | *TF* | *MFD* | *TFD* | *TERM* | *USER* |
| CREATE_ | 3.8.1 | 3.3.1 | 3.4.1 | | | 3.6.1 | 3.12.1 |
| DESTROY_ | | 3.2.1 | 3.2.1 | | | 3.2.1 | 3.12.2 |
| DISPLAY_ | 3.2.2 | 3.3.2 | 3.2.2 | 3.5.1 | 3.5.1 | 3.6.2 | 3.12.3 |
| PRINT_ | 3.2.3 | 3.3.3 | 3.2.3 | 3.5.2 | 3.5.2 | 3.6.3 | 3.12.4 |
| EDIT_ | 3.2.4 | | 3.2.4 | | | | |
| UPDATE_ | 3.2.5 | | 3.2.5 | | | | |
| DELETEME_ | | 3.3.4 | | | | | |
| UNDELETEME_ | | 3.3.5 | | | | | |
| EXPUNGE_ | | 3.3.6 | | | | | |
| COPYME_ | | 3.3.7 | | | | | |
| MOVEME_ | | 3.3.8 | | | | | |
| COPYFRENT_ | | | 3.4.2 | | | | |
| COPYTOENT_ | | | 3.4.3 | | | | |
| DUP_ | 3.7.1 | 3.3.9 | 3.4.4 | | | | |
| RECLASSIFY_ | | 3.2.6 | 3.2.6 | | | 3.2.6 | |
| MAXCLAS_ | | | | | | 3.6.4 | |
| SEND_ | 3.8.2 | | | | | | |
| READDRESS_ | 3.11.2 | | | | | | |
| REPLY_ | 3.9.1 | | | | | | |
| FORINFO_ | 3.9.2 | | | | | | |
| FORACTION_ | 3.11.1 | | | | | | |
| FORCOORD_ | 3.10.1 | | | | | | |
| FORRELEASE_ | 3.10.2 | | | | | | |
| DISPLAYAS_ | 3.2.7 | 3.2.7 | 3.2.7 | 3.2.7 | 3.2.7 | | |
| PRINTAS_ | 3.2.8 | 3.2.8 | 3.2.8 | 3.2.8 | 3.2.8 | | |
| EDITAS_ | 3.2.9 | 3.2.9 | 3.2.9 | 3.2.9 | 3.2.9 | | |
| UPDATEAS_ | 3.2.10 | 3.2.10 | 3.2.10 | 3.2.10 | 3.2.10 | | |
| CHGCLEAR_ | | | | | | | 3.12.5 |
| CHGPW_ | | | | | | | 3.12.6 |
| ADDAROLE_ | | | | | | | 3.12.7 |
| RMVAROLE_ | | | | | | | 3.12.8 |
| ADDCROLE_ | | | | | | | 3.12.9 |
| RMVCROLE_ | | | | | | | 3.12.10 |
| LOGIN_ | | | | | | | 3.12.11 |
| LOGOUT_ | | | | | | | 3.12.12 |

## 3. Command Specifications

### 3.1. Introduction

This section contains specifications for each ICL command listed in Section 2. The specifications are modifications of earlier work [2].

#### 3.1.1. Data Type Hierarchy

The set of ICL commands can be partitioned into two smaller sets: the first contains commands associated with entities (e.g., UPDATE_MSG), the second commands associated with users (e.g., CREATE_USER). The specifications of many commands in the first set differ only in the data types of their parameters. For example, the ICL commands, RECLASSIFY_MF and RECLASSIFY_TERM, have the same syntax and semantics, except the former operates on a message file, whereas the latter operates on a terminal. To simplify and shorten the specifications, we use a single form to define such commands.

To accomplish this, we have constructed the hierarchy of data types shown in Figure 1. In the figure, an arrow pointing from data type "a" to data type "b" indicates that "a" inherits the properties of "b"; we refer to "b" as a **donor** of "a". The most **abstract** data type in Figure 1, i.e., the data type with the fewest properties, is the lowest in the hierarchy, namely, "entity". Every other data type shown possesses both its own associated properties and properties inherited from its donors. Thus the type "draft message" has the properties of "draft message", of "message", and of "entity". This means, for example, that a draft message in M2 is a container (because every message in M2 is a container) and that a draft message has a classification (because every entity has a classification).

Among the properties associated with each data type in Figure 1 is a set of ICL commands. In addition, each data type *potentially* inherits each ICL command associated with its donors. We say *potentially* because each data type above "entity" may only inherit *some* of the ICL commands associated with "entity". Consider, for example, the ICL commands, UPDATE and RECLASSIFY, both of which are associated with the data type "entity". In M2, "draft message" inherits UPDATE but does not inherit RECLASSIFY, since draft messages in M2 may be modified but not reclassified. We note, however, that a data type always inherits *all* ICL commands associated with donors *other* than "entity".

Table 3 lists the data type "user" as well as each data type shown in Figure 1, the ICL commands with which the data type is associated, and the number of the subsection in which the ICL commands are specified. Within each subsection, the ICL commands are specified in the order presented in Table 2.

A few of the entries in Table 3, e.g., "informal draft message", have no associated ICL commands. Such data types inherit all of their ICL commands from donor data types. Thus, for example, "informal draft message" inherits the ICL commands associated with "draft message" and "message" along with the subset of ICL commands associated with "entity" that are inherited by "message" and "draft message".

### 3.5. Commands on directory

**3.5.1.**

| | | | |
|---|---|---|---|
| *Generic Name* | DISPLAY | *Specific ICL Cmds* | DISPLAY_MFD<br>DISPLAY_TFD |
| *Input Pars* | dname:directory ref | | |
| *Output Pars* | c:classification<br>sequence of<br>(r:ref,cl:classification[,ccr:boolean]) | | |
| *Constraints* | None. | | |
| *Description* | Displays the name, classification, and, if the directory is a message file directory, the CCR mark of each entity in the directory named **dname.** Also displays the classification **c** of the directory. | | |

**3.5.2.**

| | | | |
|---|---|---|---|
| *Generic Name* | PRINT | *Specific ICL Cmds* | PRINT_MFD<br>PRINT_TFD |
| *Input Pars* | dname:directory ref | | |
| *Output Pars* | c:classification<br>sequence of<br>(r:ref,cl:classification[,ccr:boolean]) | | |
| *Constraints* | None. | | |
| *Description* | Prints the name, classification, and, if the directory is a message file directory, the CCR mark of each entity in the directory named **dname.** Also prints the classification **c** of the directory. | | |

**3.4.3.**

| | | | |
|---|---|---|---|
| *Generic Name* | COPYTOENT | *Specific ICL Cmds* | COPYTOENT_TF |
| *Input Pars* | tname:text file ref<br>ename:entity ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Appends a copy of each object in **tname** to the entity **ename.** Each new object retains the value and classification of the corresponding object in **tname.** The data type of each new object may have a different data type than the corresponding object in **tname,** since the object type must be consistent with the entity into which the new object is being inserted. | | |

**3.4.4.**

| | | | |
|---|---|---|---|
| *Generic Name* | DUP | *Specific ICL Cmds* | DUP_TF |
| *Input Pars* | tname1:text file ref<br>tname2:char string | | |
| *Output Pars* | - | | |
| *Constraints* | The new text file cannot be created if a text file with reference **tname2** already exists in the user's text file directory. | | |
| *Description* | Creates a new text file that is a duplicate of the text file **tname1.** The new text file contains a copy of each paragraph in **tname1.** The new text file has the reference **tname2** and the same classification and access set as **tname1.** Inserts text file **tname** into the user's text file directory. | | |

### 3.4. Commands on text file

#### 3.4.1.

| | | | |
|---|---|---|---|
| *Generic Name* | CREATE | *Specific ICL Cmds* | CREATE_TF |
| *Input Pars* | tname:char string<br>c:classification<br>as:access set | | |
| *Output Pars* | tf:text file val<br>c:classification | | |
| *Constraints* | The new text file cannot be created if a text file with reference **tname** already exists in the user's text file directory. | | |
| *Description* | Creates a text file with reference **tname**, classification **c**, and access set **as.** (The MMS provides a default access set; in M2, the user can later modify the access set using EDITAS_TF and UPDATEAS_TF.) Inserts text file **tname** in the user's text file directory. Displays the value **tf** of the new text file and its classification **c.** The user is permitted to edit the displayed text file. | | |

#### 3.4.2.

| | | | |
|---|---|---|---|
| *Generic Name* | COPYFRENT | *Specific ICL Cmds* | COPYFRENT_TF |
| *Input Pars* | ename:entity ref<br>tname:text file ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Appends a copy of the entity **ename** to the text file **tname.** If the entity **ename** is an object, the new entity has the same value and classification as **ename** but has 'paragraph' as its data type. If the entity **ename** is a sequence of objects, each new object has the same value and classification as its counterpart object but has 'paragraph' as its data type. | | |

**3.3.9.**

| | | | |
|---|---|---|---|
| *Generic Name* | DUP | *Specific ICL Cmds* | DUP_MF |
| *Input Pars* | mfname1:message file ref<br>mfname2:char string | | |
| *Output Pars* | - | | |
| *Constraints* | The new message file cannot be created if a message file with reference **mfname2** already exists in the user's message file directory. | | |
| *Description* | Creates a new message file that is a duplicate of the message file **mfname1.** The new message file has the reference **mfname2** and the same classification, CCR-mark, and access set as **mfname1.** Creates a copy of each entry in message file **mfname1** and inserts the entry in message file **mfname2.** New copies of the messages associated with the entries are NOT created. The new entries point to the same message copies as the existing entries. Inserts message file **mfname** into the user's message file directory. | | |

**3.3.6.**

| | | | |
|---|---|---|---|
| *Generic Name* | EXPUNGE | *Specific ICL Cmds* | EXPUNGE_MF |
| *Input Pars* | mfname:message file ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Removes all message entries marked 'deleted' from the message file **mfname.** Maintains the order of the remaining message entries. | | |

**3.3.7.**

| | | | |
|---|---|---|---|
| *Generic Name* | COPYME | *Specific ICL Cmds* | COPYME_MF |
| *Input Pars* | mename:message entry ref<br>mfname:message file ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Appends a copy of the message entry **mename** to the message file **mfname.** The new entry in **mfname** is marked 'new'. | | |

**3.3.8.**

| | | | |
|---|---|---|---|
| *Generic Name* | MOVEME | *Specific ICL Cmds* | MOVEME_MF |
| *Input Pars* | mename:message entry ref<br>mfname:message file ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Appends a copy of the message entry **mename** to the message file **mfname.** Marks 'deleted' the message entry **mename.** The new entry in **mfname** is marked 'new'. | | |

**3.3.3.**

| | | | |
|---|---|---|---|
| *Generic Name* | PRINT | *Specific ICL Cmds* | PRINT_MF |
| *Input Pars* | mfname:message file ref<br>f:filter | | |
| *Output Pars* | mf:message file val<br>or subset<br>c:classification | | |
| *Constraints* | None. | | |
| *Description* | Prints the value of all message entries in the message file **mfname** that satisfy the filter **f.** Also prints the classification **c** of the message file. | | |

**3.3.4.**

| | | | |
|---|---|---|---|
| *Generic Name* | DELETEME | *Specific ICL Cmds* | DELETEME_MF |
| *Input Pars* | mename:message entry ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Marks 'deleted' the message entry **mename.** | | |

**3.3.5.**

| | | | |
|---|---|---|---|
| *Generic Name* | UNDELETEME | *Specific ICL Cmds* | UNDELETEME_MF |
| *Input Pars* | mename:message entry ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Removes the 'deleted' mark from the message entry **mename.** | | |

### 3.3. Commands on message file

#### 3.3.1.

| | | | |
|---|---|---|---|
| *Generic Name* | CREATE | *Specific ICL Cmds* | CREATE_MF |
| *Input Pars* | mfname:char string<br>c:classification<br>ccr:boolean<br>as:access set | | |
| *Output Pars* | - | | |
| *Constraints* | The new message file cannot be created if a message file with reference **mfname** already exists in the user's message file directory. | | |
| *Description* | Creates a message file with reference **mfname**, classification **c**, CCR mark **ccr**, and access set **as.** (In all three family members, the MMS provides a default access set; in M2, the user can later modify the access set using EDITAS_MF and UPDATEAS_MF.) The new message file contains no entries. Inserts message file **mfname** in the user's message file directory. | | |

#### 3.3.2.

| | | | |
|---|---|---|---|
| *Generic Name* | DISPLAY | *Specific ICL Cmds* | DISPLAY_MF |
| *Input Pars* | mfname:message file ref<br>f:filter | | |
| *Output Pars* | mf:message file val<br>or subset<br>c:classification | | |
| *Constraints* | None. | | |
| *Description* | Displays the value of all message entries in the message file **mfname** that satisfy the filter **f.** Also displays the classification **c** of the message file. | | |

**3.2.9.**

| | | | |
|---|---|---|---|
| *Generic Name* | EDITAS | *Specific ICL Cmds* | EDITAS_MF<br>EDITAS_TF<br>EDITAS_MSG<br>EDITAS_MFD<br>EDITAS_TFD |
| *Input Pars* | ename:entity ref | | |
| *Output Pars* | accset:access set | | |
| *Constraints* | The type of the entity **ename** is message file, text file, message, message file directory, or text file directory. | | |
| *Description* | Displays the access set **accset** of the entity **ename.** The user is permitted to edit the displayed access set. | | |

**3.2.10.**

| | | | |
|---|---|---|---|
| *Generic Name* | UPDATEAS | *Specific ICL Cmds* | UPDATEAS_MF<br>UPDATEAS_TF<br>UPDATEAS_MSG<br>UPDATEAS_MFD<br>UPDATEAS_TFD |
| *Input Pars* | ename:entity ref<br>accset:access set | | |
| *Output Pars* | - | | |
| *Constraints* | The type of the entity **ename** is message file, text file, message, message file directory, or text file directory. | | |
| *Description* | Modifies the access set of the entity **ename** to have the value **accset.** | | |

**3.2.7.**

| | | | |
|---|---|---|---|
| *Generic Name* | DISPLAYAS | *Specific ICL Cmds* | DISPLAYAS_MF<br>DISPLAYAS_TF<br>DISPLAYAS_MSG<br>DISPLAYAS_MFD<br>DISPLAYAS_TFD |
| *Input Pars* | ename:entity ref | | |
| *Output Pars* | accset:access set | | |
| *Constraints* | The type of the entity **ename** is message file, text file, message, message file directory, or text file directory. | | |
| *Description* | Displays the access set **accset** of the entity **ename.** The user is not allowed to modify the access set. | | |

**3.2.8.**

| | | | |
|---|---|---|---|
| *Generic Name* | PRINTAS | *Specific ICL Cmds* | PRINTAS_MF<br>PRINTAS_TF<br>PRINTAS_MSG<br>PRINTAS_MFD<br>PRINTAS_TFD |
| *Input Pars* | ename:entity ref | | |
| *Output Pars* | accset:access set | | |
| *Constraints* | The type of the entity **ename** is message file, text file, message, message file directory, or text file directory. | | |
| *Description* | Prints the access set **accset** of the entity **ename.** | | |

**3.2.4.**

| | | | |
|---|---|---|---|
| *Generic Name* | EDIT | *Specific ICL Cmds* | EDIT_MSG<br>EDIT_TF |
| *Input Pars* | ename:entity ref | | |
| *Output Pars* | ent:entity val<br>c:classification | | |
| *Constraints* | The type of the entity **ename** is draft message or text file. | | |
| *Description* | Displays the value **ent** and the classification **c** of the entity **ename.** The user is permitted to edit the displayed entity. | | |

**3.2.5.**

| | | | |
|---|---|---|---|
| *Generic Name* | UPDATE | *Specific ICL Cmds* | UPDATE_MSG<br>UPDATE_TF |
| *Input Pars* | ename:entity ref<br>ent:entity val | | |
| *Output Pars* | - | | |
| *Constraints* | The type of the entity **ename** is draft message or text file. | | |
| *Description* | Sets the value of the entity **ename** to **ent.** | | |

**3.2.6.**

| | | | |
|---|---|---|---|
| *Generic Name* | RECLASSIFY | *Specific ICL Cmds* | RECLASSIFY_MF<br>RECLASSIFY_TF<br>RECLASSIFY_TERM |
| *Input Pars* | ename:entity ref<br>c:classification | | |
| *Output Pars* | - | | |
| *Constraints* | The type of the entity **ename** is message file, text file, or terminal. | | |
| *Description* | Assigns the classification **c** to the entity **ename.** | | |

### 3.2. Commands on entity

#### 3.2.1.

| | | | |
|---|---|---|---|
| *Generic Name* | DESTROY | *Specific ICL Cmds* | DESTROY_MF<br>DESTROY_TF<br>DESTROY_TERM |
| *Input Pars* | ename:entity ref | | |
| *Output Pars* | - | | |
| *Constraints* | The type of the entity **ename** is message file, text file, or terminal. | | |
| *Description* | Destroys the entity **ename.** Removes the entity from any containers that contain it. | | |

#### 3.2.2.

| | | | |
|---|---|---|---|
| *Generic Name* | DISPLAY | *Specific ICL Cmds* | DISPLAY_MSG<br>DISPLAY_TF |
| *Input Pars* | ename:entity ref | | |
| *Output Pars* | ent:entity val<br>c:classification | | |
| *Constraints* | The type of the entity **ename** is message or text file. | | |
| *Description* | Displays the value **ent** and the classification **c** of the entity **ename.** The user is not permitted to edit the displayed entity. | | |

#### 3.2.3.

| | | | |
|---|---|---|---|
| *Generic Name* | PRINT | *Specific ICL Cmds* | PRINT_MSG<br>PRINT_TF |
| *Input Pars* | ename:entity ref | | |
| *Output Pars* | ent:entity val<br>c:classification | | |
| *Constraints* | The type of the entity **ename** is message or text file. | | |
| *Description* | Prints the value **ent** and the classification **c** of the entity **ename.** | | |

### 3.1.2. Form Used to Specify the ICL Commands

Each form lists the one or more ICL commands to which it applies and provides a generic name for the commands. The sections of the form named "Input Pars" and "Output Pars" indicate the command parameters. The user provides the input parameters along with the command name (the user interface may provide defaults for some of the input parameters). The output parameters identify the items that are displayed at the user terminal or output by a printer. Each parameter is expressed as "x:y", where x is the parameter and y is the attribute type (e.g., classification, access set) or the data type of the parameter. The "Description" section gives a prose description of the command semantics. The "Constraints" section defines any constraints on the command's input parameters. In particular, this section identifies the specific data types that inherit the ICL commands associated with the data type "entity".

The successful completion of each ICL command requires that the preconditions described in the MMS security model and various message system preconditions are satisfied. This document does not include the security model preconditions nor a complete statement of the message system preconditions. A complete statement of these preconditions will be included in a future report. However, we do list below three preconditions that *all* ICL commands must enforce:

(a) Each input parameter must have the type defined in the "Input Pars" section and must satisfy the conditions listed in the "Constraints" section.

(b) Each reference/userID supplied as an input parameter must refer to an existing entity/user.

(c) To invoke any command other than the LOGIN command, the user must be logged in to some terminal.

Any ICL command that does not satisfy the required preconditions cannot be completed.

The following abbreviations are used in the specifications:

| Abbreviation | Meaning |
|---|---|
| ref | reference |
| char | character |
| setof | set of |
| val | value |

Table 3: MMS Data Types and Their Associated ICL Commands

| *Data Type* | *Section* | *ICL Commands* |
|---|---|---|
| entity | 3.2 | DESTROY,DISPLAY,PRINT, EDIT,UPDATE,RECLASSIFY, DISPLAYAS,PRINTAS,EDITAS, UPDATEAS |
| message file | 3.3 | CREATE,DISPLAY,PRINT, DELETEME,UNDELETEME,EXPUNGE, COPYME,MOVEME,DUP |
| text file | 3.4 | CREATE,COPYFRENT,COPYTOENT, DUP |
| directory | 3.5 | DISPLAY,PRINT |
| message file directory | - | None |
| text file directory | - | None |
| terminal | 3.6 | CREATE,DISPLAY,PRINT,MAXCLAS |
| message | 3.7 | DUP |
| draft message | 3.8 | CREATE,SEND |
| sent message | 3.9 | REPLY,FORINFO |
| formal draft message | 3.10 | FORCOORD,FORRELEASE |
| formal sent message | 3.11 | FORACTION,READDRESS |
| informal draft message | - | None |
| informal sent message | - | None |
| user | 3.12 | CREATE,DESTROY,DISPLAY, PRINT,CHGCLEAR,CHGPW, ADDAROLE,RMVAROLE, ADDCROLE,RMVCROLE, LOGIN,LOGOUT |

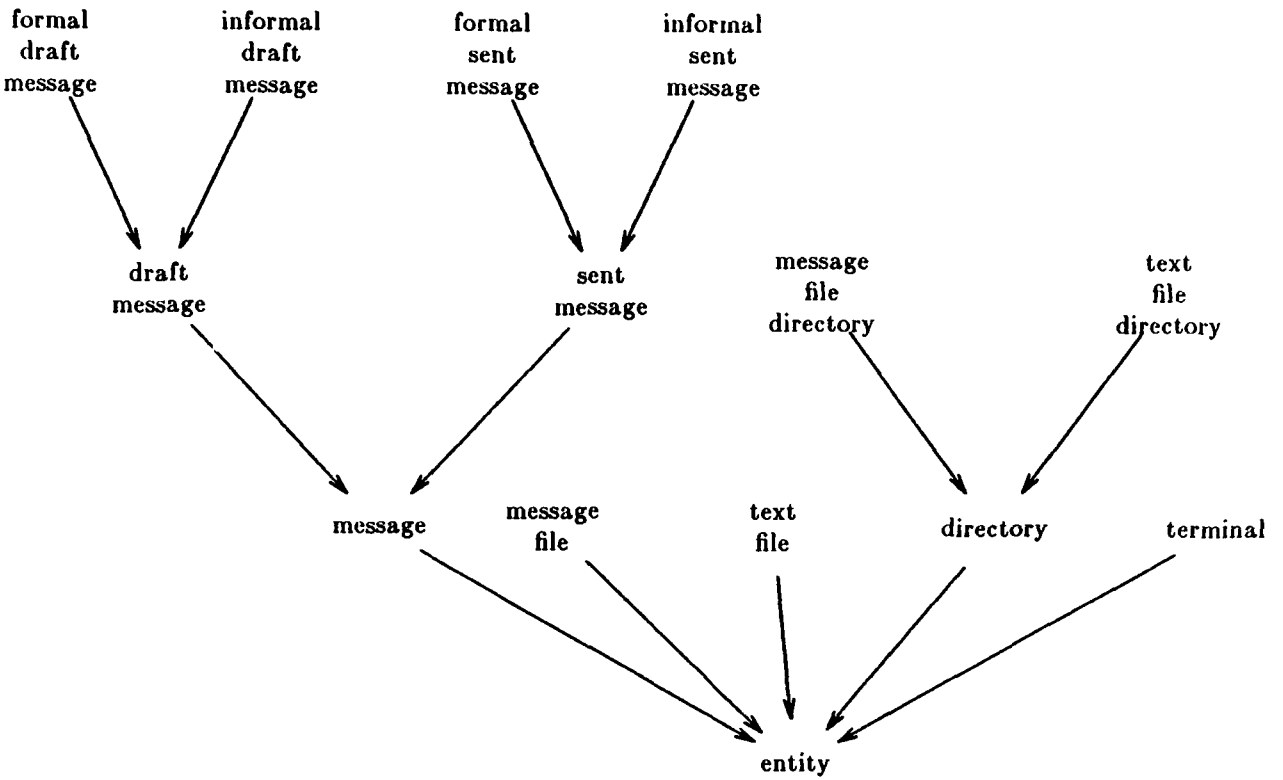



Figure 1: Hierarchy of MMS Entities*

*Because the hierarchy shown in Figure 1 includes only those MMS data types that are needed to specify the ICL commands, several of the MMS data types, e.g., paragraph and TO-field, do not appear in the figure.

### 3.6. Commands on terminal

**3.6.1.**

| | | | |
|---|---|---|---|
| *Generic Name* | CREATE | *Specific ICL Cmds* | CREATE_TERM |
| *Input Pars* | tname:char string<br>c:classification<br>as:access set | | |
| *Output Pars* | - | | |
| *Constraints* | The new terminal cannot be created if a terminal with reference **tname** already exists. | | |
| *Description* | Creates a terminal with reference **tname**, maximum classification **c**, and access set **as.** (The MMS provides a default access set that is not user-visible.) | | |

**3.6.2.**

| | | | |
|---|---|---|---|
| *Generic Name* | DISPLAY | *Specific ICL Cmds* | DISPLAY_TERM |
| *Input Pars* | tname:terminal ref | | |
| *Output Pars* | c1:classification<br>c2:classification | | |
| *Constraints* | None. | | |
| *Description* | Displays the maximum classification **c1** and the current classification **c2** of the terminal **tname.** | | |

**3.6.3.**

| | | | |
|---|---|---|---|
| *Generic Name* | PRINT | *Specific ICL Cmds* | PRINT_TERM |
| *Input Pars* | tname:terminal ref | | |
| *Output Pars* | c1:classification<br>c2:classification | | |
| *Constraints* | None. | | |
| *Description* | Prints the maximum classification **c1** and the current classification **c2** of the terminal **tname.** | | |

**3.6.4.**

| | | | |
|---|---|---|---|
| *Generic Name* | MAXCLAS | *Specific ICL Cmds* | MAXCLAS_TERM |
| *Input Pars* | tname:terminal ref<br>c:classification | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Sets the maximum classification of the terminal **tname** to **c.** | | |

### 3.7. Commands on message

**3.7.1.**

| | | | |
|---|---|---|---|
| *Generic Name* | DUP | *Specific ICL Cmds* | DUP_MSG |
| *Input Pars* | msgid:message ref<br>mfname:message file ref | | |
| *Output Pars* | dmsg:draft message val<br>c:classification | | |
| *Constraints* | None. | | |
| *Description* | Creates a copy of the message **msgid.** The new message has a new reference in its ID field and the same classification, the same message type (e.g., formal or informal), and, if the original message is a draft message, the same access set as the original message; the userID or role of the user who created the message is in the From field; and the creating user's site is in the Originator field. Creates a message entry for the new message and appends it to the message file **mfname.** The new entry is marked 'new'. Displays both the value **dmsg** and the classification **c** of the new message. The user is permitted to modify the displayed message. | | |

### 3.8. Commands on draft message

**3.8.1.**

| | | | |
|---|---|---|---|
| *Generic Name* | CREATE | *Specific ICL Cmds* | CREATE_MSG |
| *Input Pars* | t:message type<br>c:classification<br>as:access set<br>mfname:message file ref | | |
| *Output Pars* | dmsg:draft message val<br>c:classification | | |
| *Constraints* | None. | | |
| *Description* | Creates a draft message of message type **t**, classification **c**, and access set **as.** (The MMS provides a default access set; in M2, the user can later modify the access set using EDITAS_MSG and UPDATEAS_MSG.) Creates a message entry for the new message and appends it to the message file **mfname.** The new entry is marked 'new'. Displays the value **dmsg** and the classification **c** of the new message. The new message has a reference assigned by the MMS in its ID field, the userID or role of the user who created the message in its From field, and the user's site in its Originator field. The user is permitted to edit the displayed message. | | |

**3.8.2.**

| | | | |
|---|---|---|---|
| *Generic Name* | SEND | *Specific ICL Cmds* | SEND_MSG |
| *Input Pars* | msgid:draft message ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Sends the draft message **msgid** to all addressees listed in the address fields of the message. The message type is changed from draft message to sent message. The sent message is assigned a new ID. For each local addressee, the MMS creates a message entry for the message and appends the entry to the user's inbox, marking it 'new'. For each remote addressee, the MMS may transmit a copy of the message over the appropriate network. | | |

### 3.9. Commands on sent message

**3.9.1.**

| | | | |
|---|---|---|---|
| *Generic Name* | REPLY | *Specific ICL Cmds* | REPLY_MSG |
| *Input Pars* | msgid:sent message ref<br>t:message type<br>c:classification<br>as:access set<br>mfname:message file ref | | |
| *Output Pars* | dmsg:draft message val<br>c:classification | | |
| *Constraints* | None. | | |
| *Description* | Creates a message of message type **t**, classification **c**, and access set **as**. Creates a message entry for the message and appends it to the message file **mfname**. The new entry is marked 'new'. Displays the value **dmsg** of the new message. The new message has a reference assigned by the MMS in its ID field, the userID or role of the user who created the message in its From field, the contents of the From field of the message **msgid** in the To field, and the same Subject field as **msgid**. Also displays the message classification **c**. The user is allowed to edit the new message. | | |

**3.9.2.**

| | | | |
|---|---|---|---|
| *Generic Name* | FORINFO | *Specific ICL Cmds* | FORINFO_MSG |
| *Input Pars* | msgid:sent message ref<br>A:setof(addressee) | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | For formal messages, forwards for "info" the message **msgid** to all addressees in **A**. For informal messages, forwards the message **msgid** to all addressees in **A**. For each user in **A**, creates a message entry for the message and appends the entry to the user's inbox. The entry has the 'for info' mark and a 'new' mark. | | |

### 3.10. Commands on formal draft message

**3.10.1.**

| | | | |
|---|---|---|---|
| *Generic Name* | FORCOORD | *Specific ICL Cmds* | FORCOORD_MSG |
| *Input Pars* | msgid:formal draft message ref<br>A:setof(addressee) | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Forwards the formal draft message **msgid** to all addressees in **A.** The data type of **msgid** does not change. For each user in **A,** creates an entry for the message and appends it to the user's inbox. The entry has the 'for coord' mark and a 'new' mark. Each addressee may either send comments on the message via another message or, if the addressee has "update" permission, he may modify **msgid.** | | |

**3.10.2.**

| | | | |
|---|---|---|---|
| *Generic Name* | FORRELEASE | *Specific ICL Cmds* | FORRELEASE_MSG |
| *Input Pars* | msgid:formal draft message ref<br>a:addressee | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | The formal draft message **msgid** is forwarded to the addressee **a** for release. The user who receives the message "for release" has a new entry for the message in his inbox; the entry has the 'for release' mark and a 'new' mark. | | |

### 3.11. Commands on formal sent message

**3.11.1.**

| | | | |
|---|---|---|---|
| *Generic Name* | FORACTION | *Specific ICL Cmds* | FORACTION_MSG |
| *Input Pars* | msgid:formal sent message ref<br>A:setof(addressee) | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Forwards for "action" the formal sent message **msgid** to each addressee in **A.** For each user in **A,** creates an entry for the message and appends it to the user's inbox. The entry has the 'for action' mark and a 'new' mark. | | |

**3.11.2.**

| | | | |
|---|---|---|---|
| *Generic Name* | READDRESS | *Specific ICL Cmds* | READDRESS_MSG |
| *Input Pars* | msgid:formal sent message ref<br>mfname:message file ref | | |
| *Output Pars* | dmsg:draft message val<br>c:classification | | |
| *Constraints* | None. | | |
| *Description* | Creates a copy of the sent formal message **msgid.** The new formal draft message has the same value as **msgid,** namely, **dmsg;** the same classification as **msgid,** namely, **c;** a reference assigned by the MMS in its ID field; and the same message type as **msgid.** Creates a message entry for the new message and appends it to the message file **mfname.** The user is only permitted to fill in the address fields of the new message. | | |

### 3.12. Commands on user

#### 3.12.1.

| | | | |
|---|---|---|---|
| *Generic Name* | CREATE | *Specific ICL Cmds* | CREATE_USER |
| *Input Pars* | uid:char string<br>pw:password<br>cl:clearance<br>A:setof(role) | | |
| *Output Pars* | - | | |
| *Constraints* | The new user cannot be created if there already exists a user with userID **uid.** | | |
| *Description* | Creates a user with userID **uid,** password **pw,** clearance **cl,** and the authorized roles in **A.** Also creates a message file directory, a text file directory, an inbox for the new user, and initial access sets for the directories and the inbox. | | |

#### 3.12.2.

| | | | |
|---|---|---|---|
| *Generic Name* | DESTROY | *Specific ICL Cmds* | DESTROY_USER |
| *Input Pars* | uid:userID | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Destroys the user **uid.** Also destroys the user's message file directory, text file directory, and inbox. Any message files or text files that are solely in this user's directories are also destroyed. | | |

**3.12.3.**

| | | | |
|---|---|---|---|
| *Generic Name* | DISPLAY | *Specific ICL Cmds* | DISPLAY_USER |
| *Input Pars* | uid:userID | | |
| *Output Pars* | cl:clearance<br>A:setof(role)<br>B:setof(role)<br>tname:terminal ref | | |
| *Constraints* | None. | | |
| *Description* | Displays the clearance **cl** of user **uid,** his set of authorized roles **A,** and, if **uid** is logged on, his set of current roles **B** and the terminal **tname** that he is logged onto. Only the system security officer is permitted to execute this command. | | |

**3.12.4.**

| | | | |
|---|---|---|---|
| *Generic Name* | PRINT | *Specific ICL Cmds* | PRINT_USER |
| *Input Pars* | uid:userID | | |
| *Output Pars* | cl:clearance<br>A:setof(role)<br>B:setof(role)<br>tname:terminal ref | | |
| *Constraints* | None. | | |
| *Description* | Prints the clearance **cl** of user **uid,** his set of authorized roles **A,** and, if **uid** is logged on, his set of current roles **B** and the terminal **tname** that he is logged onto. Only the system security officer is permitted to execute this command. | | |

**3.12.5.**

| | | | |
|---|---|---|---|
| *Generic Name* | CHGCLR | *Specific ICL Cmds* | CHGCLR_USER |
| *Input Pars* | uid:userID<br>cl:clearance | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Changes the clearance of the user **uid** to **cl.** | | |

**3.12.6.**

| | | | |
|---|---|---|---|
| *Generic Name* | CHGPW | *Specific ICL Cmds* | CHGPW_USER |
| *Input Pars* | uid:userID<br>oldpw:password<br>newpw:password | | |
| *Output Pars* | - | | |
| *Constraints* | A precondition for this command is that the old value of the password is **oldpw.** (The system security officer need not provide the old password.) | | |
| *Description* | Changes the password of the user **uid** to **newpw.** | | |

**3.12.7.**

| | | | |
|---|---|---|---|
| *Generic Name* | ADDAROLE | *Specific ICL Cmds* | ADDAROLE_USER |
| *Input Pars* | uid:userID<br>A:setof(role) | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Adds the roles in **A** to the set of roles authorized for the user **uid.** | | |

**3.12.8.**

| | | | |
|---|---|---|---|
| *Generic Name* | RMVAROLE | *Specific ICL Cmds* | RMVAROLE_USER |
| *Input Pars* | uid:userID<br>A:setof(role) | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Removes the roles in **A** from the set of roles authorized for the user **uid.** | | |

**3.12.9.**

| | | | |
|---|---|---|---|
| *Generic Name* | ADDCROLE | *Specific ICL Cmds* | ADDCROLE_USER |
| *Input Pars* | uid:userID<br>A:setof(role) | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Adds the roles in **A** to the set of current roles authorized for the user **uid.** | | |

**3.12.10.**

| | | | |
|---|---|---|---|
| *Generic Name* | RMVCROLE | *Specific ICL Cmds* | RMVCROLE_USER |
| *Input Pars* | uid:userID<br>A:setof(role) | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Removes the roles in **A** from the set of current roles authorized for the user **uid.** | | |

**3.12.11.**

| | | | |
|---|---|---|---|
| *Generic Name* | LOGIN | *Specific ICL Cmds* | LOGIN_USER |
| *Input Pars* | tname:terminal ref<br>uid:userID<br>pw:password<br>c:classification<br>A:setof(role) | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Logs the user with userID **uid** and password **pw** onto the terminal **tname.** Sets the classification of the terminal to **c.** Assigns the roles in **A** as the current roles assigned to the user. Fixes the access set of **tname** so that the user **uid** can RECLASSIFY the terminal and LOGOUT of the terminal. | | |

**3.12.12.**

| | | | |
|---|---|---|---|
| *Generic Name* | LOGOUT | *Specific ICL Cmds* | LOGOUT_USER |
| *Input Pars* | uid:userID<br>tname:terminal ref | | |
| *Output Pars* | - | | |
| *Constraints* | None. | | |
| *Description* | Logs the user with userID **uid** off of the terminal **tname.** Fixes the access set of the terminal so that the user **uid** can no longer RECLASSIFY or LOGOUT of the terminal. Removes all message entries marked 'deleted' from each of the user's message files. Removes the 'new' mark from each entry that is marked 'new'. Sets the user's current role set to the empty set. | | |

## 4. Glossary

This section provides definitions for terms used in this report. In particular, terms that appear as parameters to ICL commands are defined. Many of the definitions have been extracted from the MMS security model [1].

**access set**: a set of triples associated with an entity. Each triple consists of a userID or role, an operation, and an operand index. If a given operation requires more than one operand, the operand index specifies the position in which a reference to this entity may appear as an operand. The existence of a particular triple in the access set implies that a user corresponding to the userID or role is authorized to invoke the specified operation on the entity with which the access set is associated.

**address field**: a field of a message that contains addresses. The To field, the CC field, and the From field are examples of address fields.

**addressee**: a userID, a role, or an organization to whom a message can be sent.

**authorized role**: a role that the user is authorized for. The system security officer assigns the user's set of authorized roles.

**classification**: a designation reflecting the damage that could be caused by unauthorized disclosure of information. Includes a sensitivity level (UNCLASSIFIED, CONFIDENTIAL, SECRET, TOP SECRET), and a set of zero or more compartments (CRYPTO, NUCLEAR, etc.).

**clearance**: the degree of trust associated with a person, expressed in the same way as a classification. In a secure MMS, each user will have a clearance.

**container**: a unit of information that has a classification and that may contain objects and/or other containers each with their own classification. Unlike an object, a container can be multilevel.

**CCR**: an attribute of a container. CCR is an abbreviation for Container Clearance Required. CCR containers require that a user who wishes to view information in a container have a clearance that exceeds or equals the classification of the container.

**current role**: a role that the user has at a given time. The user's set of current roles is a subset of his set of authorized roles. The user defines his set of current roles.

**draft message**: a message in draft form. Draft messages are messages that users create and edit. The SEND_MSG command converts a draft message into a sent message, distributing the latter to its addressees.

**entity**: an object or a container.

**filter**: a set of criteria. To satisfy a filter, a message entry must satisfy all the criteria of the filter. An example of a filter is ALL; every message entry satisfies the filter ALL.

**formal message**: a class of messages that are exchanged by military organizations rather than by individuals. Such messages are always "on the record" and are stored for lengthy periods after their transmission. They contain special fields, such as Info and Precedence. Special operations, e.g., FORACTION_MSG, FORINFO_MSG, and READDRESS_MSG, may be applied to formal messages. Only users authorized for the role "releaser" can releas · a formal message.

**inbox**: a message file in which the MMS inserts each message that is sent to a given user. Every user has an inbox.

**informal message**: a class of messages that are exchanged by individuals. Such messages are "off the record" and there is no official requirement for retaining copies. They have fewer fields than formal messages and fewer operations can be applied to them.

**message**: a set of fields, including To, From, Subject, and Text. A message is either a draft message or a sent message. Moreover, every message has a message type, e.g., formal.

**message entry**: consists of a message and status information about the message, e.g., whether the message is 'for action', 'for release', etc.

**message file**: a sequence of message entries.

**message file directory**: a set of named message files that a user owns. Each user has a single message file directory.

**message type**: an attribute of a message that determines the fields that the message contains, the set of operations that can be applied to the message, and whether the message is "on the record". In many MMSs, there are two message types: formal and informal.

**object**: the smallest unit of information in the MMS that has a classification. An object contains no other objects and cannot be multilevel.

**password**: a character string that is used to restrict usage of a userID.

**reference**: a name for an entity. The reference may be direct, e.g., a date-time-group coupled with an originator. A reference may also be indirect, e.g., the "current message's Text field's third paragraph."

**role**: the job that the user is performing, such as downgrader, releaser, etc. A user is always associated with at least one role at any instant, and the user can change roles during a session. The system security officer assigns the user's set of authorized roles. Whenever the user is logged in, his set of current roles specifies the roles that are valid at the time.

**sent message**: a message that has been released. The SEND_MSG command converts a draft message into a sent message.

**terminal**: the device onto which the user logs in to use the MMS. Most MMS output is displayed on the user's terminal.

**text file**: a sequence of paragraphs. A user may save text, address lists, and other miscellaneous information in text files.

**text file directory**: a set of named text files. Each user has a single text file directory.

**user**: a person authorized to use the MMS.

**userID**: a character string used to denote a user of the system. To use the MMS, a person presents a userID to the system, and the system authenticates that the person is the user corresponding to that userID. Each user has a unique userID.

## REFERENCES


[1] C. Landwehr, C. Heitmeyer, and J. McLean, "A Security Model for Military Message Systems," *ACM Trans. on Computer Systems*, Vol. 2, No. 3, Aug. 1984.

[2] C. Heitmeyer, "An Intermediate Command Language for Military Message Systems," NRL tech memo 7590-450:CH:ch, Nov. 1981.

## APPENDIX

This section provides an alphabetical index to the ICL command specifications.

| ICL Command | Where Spec. |
|---|---|
| ADDAROLE_USER | 3.12.7 |
| ADDCROLE_USER | 3.12.9 |
| CHGCLEAR_USER | 3.12.5 |
| CHGPWD_USER | 3.12.6 |
| COPYME_MF | 3.3.7 |
| COPYFRENT_MF | 3.4.2 |
| COPYTOENT_MF | 3.4.3 |
| CREATE_MF | 3.3.1 |
| CREATE_MSG | 3.8.1 |
| CREATE_TERM | 3.6.1 |
| CREATE_TF | 3.4.1 |
| CREATE_USER | 3.12.1 |
| DELETEME_MF | 3.3.4 |
| DESTROY_MF | 3.2.1 |
| DESTROY_TERM | 3.2.1 |
| DESTROY_TF | 3.2.1 |
| DESTROY_USER | 3.12.2 |
| DISPLAY_MF | 3.3.2 |
| DISPLAY_MFD | 3.5.1 |
| DISPLAY_MSG | 3.2.2 |
| DISPLAY_TERM | 3.6.2 |
| DISPLAY_TF | 3.2.2 |
| DISPLAY_TFD | 3.5.1 |
| DISPLAY_USER | 3.12.3 |
| DISPLAYAS_MF | 3.2.7 |
| DISPLAYAS_MFD | 3.2.7 |
| DISPLAYAS_MSG | 3.2.7 |
| DISPLAYAS_TF | 3.2.7 |
| DISPLAYAS_TFD | 3.2.7 |
| DUP_MF | 3.3.9 |
| DUP_MSG | 3.7.1 |
| DUP_TF | 3.4.4 |
| EDIT_TF | 3.2.4 |
| EDIT_MSG | 3.2.4 |
| EDITAS_MF | 3.2.9 |
| EDITAS_MFD | 3.2.9 |
| EDITAS_MSG | 3.2.9 |
| EDITAS_TF | 3.2.9 |
| EDITAS_TFD | 3.2.9 |
| EXPUNGE_MF | 3.3.6 |
| FORACTION_MSG | 3.11.1 |
| FORCOORD_MSG | 3.10.1 |
| FORINFO_MSG | 3.9.2 |
| FORRELEASE_MSG | 3.10.2 |
| LOGIN_USER | 3.12.11 |
| LOGOUT_USER | 3.12.12 |
| MAXCLAS_TERM | 3.6.4 |
| MOVEME_MF | 3.3.8 |
| PRINT_MF | 3.3.3 |
| PRINT_MFD | 3.5.2 |
| PRINT_MSG | 3.2.3 |
| PRINT_TERM | 3.6.3 |
| PRINT_TF | 3.2.3 |
| PRINT_TFD | 3.5.2 |
| PRINT_USER | 3.12.4 |
| PRINTAS_MF | 3.2.8 |
| PRINTAS_MFD | 3.2.8 |
| PRINTAS_MSG | 3.2.8 |
| PRINTAS_TF | 3.2.8 |
| PRINTAS_TFD | 3.2.8 |
| READDRESS_MSG | 3.11.2 |
| RECLASSIFY_MF | 3.2.6 |
| RECLASSIFY_TERM | 3.2.6 |
| RECLASSIFY_TF | 3.2.6 |
| REPLY_MSG | 3.9.1 |
| RMVAROLE_USER | 3.12.8 |
| RMVCROLE_USER | 3.12.10 |
| SEND_MSG | 3.8.2 |
| UNDELETEME_MF | 3.3.5 |
| UPDATE_MSG | 3.2.5 |
| UPDATE_TF | 3.2.5 |
| UPDATEAS_MF | 3.2.10 |
| UPDATEAS_MFD | 3.2.10 |
| UPDATEAS_MSG | 3.2.10 |
| UPDATEAS_TF | 3.2.10 |
| UPDATEAS_TFD | 3.2.10 |

END

FILMED

10-85

DTIC