Compartmented Mode Workstation Labeling:

Encodings Format DDS-2600-6216-93

Sun Microsystems Federal, Inc.
A Sun Microsystems, Inc. Business
901 San Antonio Road
Palo Alto, CA 94303
U.S.A.

Part No: 805-8012-10
Revision A, July 1997

THE NETWORK IS THE COMPUTER™

Contents

Figures

Tables

Preface

The proper labeling of classified information is a key feature of Compartmented Mode Workstations (CMWs). Because of the complexity of handling the many components of security labels important to the intelligence community, and the need to specify particular label values in a standard way across all CMWs, the Defense Intelligence Agency is promulgating standard labeling software. This software handles the translation of labels in both directions between human-readable and bit-encoded forms based on a set of encodings that control the translation. In performing this translation, the software enforces various rules concerning the well formedness of labels, and controls the system's label adjudication process. This document describes the format of the standard encoding specification processed by release 2.2 of the labeling software (documented in DIA publication DDS-2600-6215-93) and gives examples of its use.

This document supersedes DDS-2600-6216-91, which documented Release 2.1 of the software.

This document is under the formal control of the DIA / DS configuration management process. Exceptions to these specifications and changes will require the approval of the DS Configuration Control Board.

The DIA OPR for this document is DIA / DS-SIM,, (202) 373-8850 or AV 243-8850.

Martin Hurwitz
Director for Information Services
Defense Intelligence Agency

Release 2.2 CMW Labeling Software

This report documents the encodings format processed by Release 2.2 of the Compartmented Mode Workstation Labeling software. It completely replaces [DDS-2600-6216-91], which documented Release 2.1 of the software. This document is intended to be used by vendors who have incorporated Release 2.2 of the software into their products as the basis for their Trusted Facility Manual documentation on the encodings file. This document describes the purpose of the various components of the encodings under the assumptions that:

1. release 2.2 of the Compartmented Mode Workstation Labeling software is incorporated into a CMW in accordance with DIA user interface guidelines;
2. release 2.2 of the software is used to construct printer banner pages in the same manner as the demonstration program distributed with that software, but placing only the long names of words in the printer banner;
3. the flags feature of the translation software is not used by the system;
4. the system supports a maximum of 256 characters per line in the encodings file;
5. the system supports a maximum of 256 classifications;
6. the system supports a maximum of 128 compartment bits;
7. the system supports a maximum of 128 marking bits; and
8. there are no other limitations other than the amount of memory that can be allocated to process the encodings file.

Vendors should modify this documentation as appropriate before incorporating it into their documentation if they have used the flags feature in their system, or have different limitations than those described above.

Furthermore, this document is written without assuming that a graphical user interface for changing labels is supported. If such an interface is supported and is compliant with the guidelines in [DDS-2600-6215-91], then the vendor should consider extending this documentation to more fully explain the effects of the encodings on the graphical user interface, e.g., the effect of hierarchies and combination constraints on the annotation of words that cannot be selected (see guideline G5 in [DDS-2600-6215-91]).

Acknowledgements

This document was prepared by John P. L. Woodward of The MITRE Corporation under the direction of DIA / DS-SIM.

Gary Huber, of The MITRE Corporation (formerly of the Defense Intelligence Agency), is thanked for his overall guidance during the production of the initial version of this paper, as well as for his thorough review of the initial version of this document, and guidance about the requirements for intelligence labeling.

Jeff Berger, Brad O'Neill, Jeff Picciotto, and Mary Lou Urban, of The MITRE Corporation, are thanked for their careful review of the initial version of this document.

Guy King, formerly of the Defense Intelligence Agency, is thanked for his thorough review of the initial version of this document, and guidance about the requirements for intelligence labeling.

Len Busic and Lois Winneberger, of the Defense Intelligence Agency, are thanked for their guidance on labeling requirements.

Bev Nunan is thanked for her careful proofreading and production of this paper.

Note – Because the transfer of electronic copy from one desktop publisher to another changed the positioning of figures and tables, Sun Microsystems Federal has added figure and table captions to help identify figures and tables cited.

List of References

The table below lists documents referenced in this document.

Document	Description
DIAM 50-4	Defense Intelligence Agency,“Security of Compartmented Computer Operations (U),” DIA Manual 50-4, 24 June 1980, CONFIDENTIAL.
DDS-2600-6215-91	Compartmented Mode Workstation Labeling: Source Code and User Interface Guidelines, DDS-2600-6215-91, Defense Intelligence Agency, November 1991.
DDS-2600-6216-91	Compartmented Mode Workstation Labeling: Encodings Format, DDS-2600-6216-91, Defencs Intelligence Agency, June 1991
DDS-2600-6243-9	Compartmented Mode Workstation Evaluation Criteria, Version 1, DDS-2600-6243-91, Defense Intelligence Agency, November 1991.
DOD 5200.28-STD	Trusted Computer System Evaluation Criteria , DOD 5200.28-STD, December 1985.

How This Book is Organized

This book is organized as follows.

Chapter 1, “Introduction” provides general information on what a label encodings file is.

Chapter 2, “Structure and Syntax of Encodings File” explains what is in a label encodings file, how the file is set up, and proper syntax.

Chapter 3, “Classification Encodings” explains how classifications are defined and used in the label encodings file.

Chapter 4, “Information Label Encodings” explains how information labels are defined and used in the label encodings file.

Chapter 5, “Sensitivity Label, Clearance, Channels, and Printer Banner Encodings” explains how sensitivity labels, clearances, channels, and printer banner information are defined and used in the label encodings file.

Chapter 6, “Accreditation Range and Name Information Label Encodings” explains how accreditation ranges and name information labels are defined and used in the label encodings file.

Chapter 7, “General Considerations for Specifying Encodings” explains how to define relationships in the label encodings file.

Chapter 8, “Enforcing Proper Label Adjudications” describes how to use compartments and/or markings to effect proper label adjudication by the system.

Appendix A, “Encodings Specifications Error Messages” documents the error messages generated by incorrect label encodings file specifications.

Appendix B, “Annotated Sample Encodings” provides an example label encodings file with explanations on the various specifications in the file.

Appendix C, “CMW Labeling Software C1.0 Release Notes, 6/8/93” is an addendum to the original document.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals using this program.

For a list of documents and how to order them, see the catalog section of the SunExpress™ Internet site at `http://www.sun.com/sunexpress`.

Introduction 1

The Compartmented Mode Workstation (CMW) Evaluation Criteria, Version 1 [DDS-2600-6243-91] defines minimum security requirements for workstations to be accredited in the Compartmented Mode under the policy set forth in Defense Intelligence Agency Manual 50-4 [DIAM 50-4]. Because of the number of CMWs needed throughout the intelligence community and the need for interoperability among the CMWs, standard encodings of security labels are necessary.

Defining encodings for security labels is a three-step process. First, the set of human-readable labels to be represented must be identified and understood. The definition of this set includes the list of classifications and other words used in the human-readable labels, relations between and among the words, classification restrictions associated with use of each word, and intended use of the words in mandatory access control and labeling system output. Next, this definition is associated with an internal format of integers, bit patterns, and logical relationship statements. Finally, a CMW system file is created to store the encodings. This document emphasizes the second and third steps, and assumes that the first has already been performed.

The encodings are used by a CMW to control the conversion of human-readable labels into the internal format used by the CMW, the conversion from the internal format to a human-readable canonical form, and the construction of banner pages for printed output. Furthermore, though not used directly by the CMW in combining information labels, the encodings values are critical in adjudicating the combinations of different information label components. Encodings must be provided for 1) classifications; 2) other words in information labels, sensitivity labels, clearances, handling channels, and printer banners; and 3) the system and user accreditation ranges and related values.

A companion document, *Compartmented Mode Workstation (CMW) Labeling: Source Code and User Interface Guidelines* [DDS-2600-6215-91], describes the standard software that operates on the encodings described in this document.

Background

As mentioned above, the encodings control the translation between the human-readable and internal formats of information labels, sensitivity labels, and clearance labels (hereafter called clearance). Human-readable labels consist of a classification followed by a set of words. The words can represent compartments[1] (in information labels, sensitivity label, and clearances), and markings (in information labels only). The internal form of labels consists of an integer classification value and a set of bits (all labels have compartment bits, whereas only information labels have marking bits), as summarized below.

Table 1-1 Label Summary

Type of Label	**Classification**	**Compartments**	**Markings**
Information	Integer	Set of Bits	Set of Bits
Sensitivity	Integer	Set of Bits	(NONE)
Clearance	Integer	Set of Bits	(NONE)

Thus, information labels have three components: classification, compartments, and markings, whereas sensitivity labels and clearance have only the first two components.

Given any two labels (information, sensitivity, or clearance), there is a relationship called *dominance* between them, defined as follows:

- Given any two information labels L1 and L2, L2 is said to *dominate* L1 if and only if the classification in L2 is greater than the classification in L1, and all compartment and marking bits that are 1 in L1 are also 1 in L2.

1. The word “compartments” is the intelligence community word most analogous to the word “categories” as used in the National Computer Security Center's *Trusted Computer System Evaluation Criteria* [DOD 5200.28-STD]. The word “compartments” will be used throughout this paper for consistency with other intelligence community documentation, but conceptually means the same as “categories.”

- Given any two labels without markings (sensitivity or clearance) L1 and L2, L2 is said to *dominate* L1 if and only if the classification in L2 is greater than the classification in L1, and all compartment bits that are 1 in L1 are also 1 in L2.
- Given an information label L1 and a label without markings (sensitivity or clearance) L2, L2 is said to *dominate* L1 if and only if the classification in L2 is greater than the classification in L1, and all compartment bits that are 1 in L1 are also 1 in L2.

In addition to the dominance relationship, there are two other relationships between labels with the same components. Two labels are *equal* if their classifications are equal and their sets of bits are equal. Two labels are *incomparable* if each label contains a 1 bit that is 0 in the other label. Stated in terms of dominance, two labels with the same components are equal if *each* label dominates the other, and two labels are incomparable if *neither* label dominates the other.

Figure 1-1 shows three labels and their associated compartment or marking bits. As indicated above, L2 dominates L1. L3 is incomparable to either L1 or L2. Finally, all three labels (in fact all possible labels) both dominate and equal themselves.

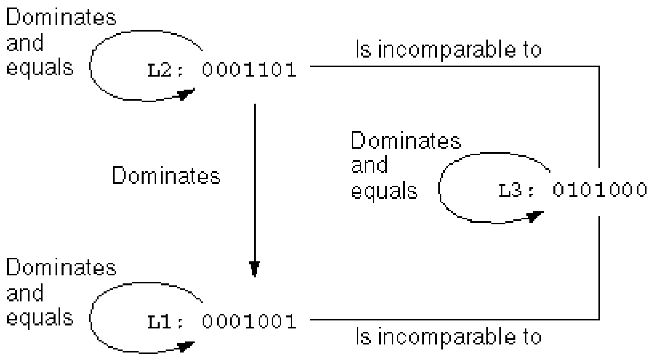


Figure 1-1 Label Relationships

The words that follow the classification in the human-readable representation of labels are said to be either normal or *inverse*[1]. Adding a normal word to the human-readable representation of a label increases the sensitivity of the label (i.e., increases the number of compartment or marking bits that are 1). Adding an inverse word to a human-readable label does not increase the sensitivity of the label, but instead either decreases or otherwise changes the sensitivity of the label, (i.e., changes at least one bit in the internal form of the label from 1 to 0). Stated in terms of dominance, adding a normal word to a label changes the label into a new label that dominates (is hierarchically above) the original one. Finally, adding an inverse word to a label changes the label into one that is either 1) dominated by the original label, or 2) is incomparable to the original label.

Constructing an Encodings File

The first step in constructing an encodings file is to define a set of labels to be implemented in the CMW. Defining the labels involves:

1. determining the long and short names of the classifications and words that comprise information labels, sensitivity labels, and clearances;
2. determining a set of rules for combinations of classifications and words;
3. determining the minimum classification, handling channels, and any other information associated with compartments and markings that must appear on printer banner pages;
4. determining the minimum sensitivity level and clearance specifiable on the system; and
5. determining the user accreditation range: the set of sensitivity levels that can be used by normal system users.

Then, constructing the encodings for this set of labels involves:

1. determining the internal integer representation of classifications;
2. determining the internal compartment bit string representations of sensitivity label and clearance words; and

1. A third type of word, special inverse, is not covered here. See “Defining Prefixes And Suffixes” in Chapter 4, “Information Label Encodings, and “Using Prefixes to Specify Special Inverse Compartment and Marking Bits” in Chapter 7, “General Considerations for Specifying Encodings.”

3. determining the internal compartment and/or marking bit string representations of information label words.

The rules for combinations of classifications and words are used to:

- determine what combinations of classification and words constitute well formed labels and
- determine how the system should adjudicate the combination of two information labels.

These determinations influence the selection of the integer and bit string internal representations.

The most important and complicated aspects of constructing the encodings are the rules for well formedness and adjudication, each of which is discussed in more detail below.

Well-Formed Labels

A label is said to be well formed if it follows a specified set of rules regarding the relationships among classifications and words in the same label. The concept of well formedness applies to information labels, sensitivity labels, and clearances. The encodings and their associated software that translates human-readable labels into their internal format enforces the following types of well formedness rules.

1. A set of “default words” can be associated with 1) the least sensitive clearance and sensitivity label, and/or 2) the least sensitive information label, containing a particular classification. Such words are defined by including the compartment and/or marking bits associated with them in the initial compartments and/or initial markings associated with the classification. See the initial compartments= and initial markings= keywords in Chapter 3, “Classification Encodings.” For example, if all classified data on a particular system was to be considered NOFORN (meaning No Foreign Dissemination), and NOFORN was an information label word (i.e., has marking bits associated), then NOFORN could be encoded as a default word for all classifications above UNCLASSIFIED, and would therefore automatically appear in all information labels.

2. A minimum classification can be associated with each word, thereby preventing the word from appearing in the human-readable form of a label with a classification below the minimum. See the minclass= keyword in

Chapter 4, “Information Label Encodings.” For example, the minimum classification that should be associated with some compartments is TOP SECRET.

3. An “output minimum” classification can be associated with each word, thereby preventing the word from appearing in the human-readable form of a label with a classification below the minimum, even though it can appear in the internal form of the label. See the ominclass= keyword in Chapter 4, “Information Label Encodings.” For example, release markings do not appear in the human-readable form on the label UNCLASSIFIED, and therefore have an output minimum classification of CONFIDENTIAL.

4. An “output maximum” classification can be associated with each word, thereby preventing the word from appearing in the human-readable form of a label with a classification above the maximum, even though it can appear in the internal form of the label. See the omaxclass= keyword in Chapter 4, “Information Label Encodings.”

5. A maximum classification can be associated with each word, thereby preventing the word from appearing in a label with a classification above the maximum. See the maxclass= keyword in Chapter 4, “Information Label Encodings.” For example, see the codeword bravo4 in Appendix B, “Annotated Sample Encodings.”

6. Any specific set of words can be defined to be in a hierarchy, such that only one word in the hierarchy can appear in a label at a time. The hierarchies among words are defined by the compartment and/or marking bits chosen to represent the words internally. Simply stated, if the compartment and marking bits associated with word W2 dominate but do not equal those associated with word W1, then W2 is in a hierarchy above W1, in which case W1 and W2 can never appear in a label together. See the compartments= and markings= keywords in Chapter 4, “Information Label Encodings” and “Hierarchies of Words” in Chapter 8, “Enforcing Proper Label Adjudications.” For example, see the codewords alpha1, alpha2, and alpha3 in Appendix B, “Annotated Sample Encodings.”

7. The presence of any word in a label can require the presence of another word in the same label. See the REQUIRED COMBINATIONS: keyword in Chapter 4, “Information Label Encodings.”. For example, certain subcompartments may require the presence of their main compartment in a sensitivity label.

8. Some words can be prevented from appearing with other words in the same label, even if the words are not hierarchically related. See the COMBINATION CONSTRAINTS: keyword in Chapter 4, “Information Label Encodings.” For example, the codeword bravo4 in Appendix B, “Annotated Sample Encodings” must stand-alone in a label.

Information Label Adjudication

When two pieces of data with separate information labels (e.g., objects, files, part of a window's contents) are merged or combined, the system automatically adjudicates the combination of the two information labels, determining the single information label that properly represents the merged data. This process of adjudicating two information labels is also called *combining* the labels or *floating* one label with the second one. The values assigned to classifications and the internal compartment and marking bit representations assigned to information label words determine how the system will adjudicate information labels.

When the system adjudicates the classifications from two information labels, the resulting classification is always the classification with the greater internal integer value. Since all classifications by definition form a strict hierarchy, specifying integer values for classifications that represent the hierarchy, with the most sensitive classifications having the highest values and the least sensitive classifications having the lowest values, will assure the proper adjudication of classifications.

Considerations for the proper adjudication of words is much more complicated. The system adjudicates information label compartment and marking bits by performing a bitwise logical “or” of the bit strings, as shown in Figure 1-2.

	Bit Strings	
	Compartments	**Markings**
Information Label 1 (IL1)	10100000	00001111
Information Label 2 (IL2)	11010001	11000000
Adjudication (IL1 + IL2)	11110001	11001111

Figure 1-2 Information label bit string combination example

Proper adjudication is assured by defining the bit representation of each information label word such that the desired properties are enforced when the words are combined via logical “or.” Figure 1-2 shows a number of different possibilities for the adjudication of the combination of words. In this and following figures, (NULL) is used to indicate the absence of any word.

As mentioned above, there are two basic types of words: normal and inverse. Additionally, words can optionally appear in a hierarchy with other words. To support these different types of words, the encodings allow for a great deal of flexibility in the association of human-readable word names with internal bit patterns. Rather than simply assigning names to bits, the encodings allow word names to be associated with specific bit patterns. These bit patterns can include compartment bits, marking bits, or both. The examples shown in Figure 1-3 are expanded below, showing how the internal encodings of the words implement the desired adjudication of normal words, inverse words, words in hierarchies, composite words, and a more complex example.

In each example, the relevant bit values associated with words are shown as 1s and 0s. Irrelevant bit positions are denoted with –s. Each example below shows two labels and their combination, in both human-readable and internal forms. (NULL) is used to indicate a label containing no words. The bits shown in the examples below could be compartment bits, marking bits, or a combination of both. From the standpoint of label adjudication, there is no difference between compartment bits and marking bits.

Comment	IL1	IL2	IL1+IL2
Normal word	Word1	(NULL)	Word1
Inverse word	Word2	(NULL)	(NULL)
Both words are normal	Word1	Word3	Word1 Word3
Both words are inverse	Word2	Word6	(NULL)
Both words are inverse	Word2	Word2 Word6	Word2
Hierarchy with Word5 above Word4	Word4	Word5	Word5
Word9 is a composite of words 7 and 8	Word7	Word8	Word9
Word12 is a non-hierarchical compositeof words 10 and 11	Word10	Word11	Word10 Word11 Word12
Word13 is inverse and in a hierarchy below Word14	Word13	(anything other than Word13)	Word14

Figure 1-3 Label adjudication examples

Normal Words

Normal words are associated with internal bit patterns consisting only of 1s. Normal words can have one or more 1 bits associated with them. The example below is for the simplest and most common case, where a single bit is associated with a word. When such a word is combined with a label containing no words, the resulting label contains just the word.

Word1	1———
(NULL)	————
Word1	1———

In the following example, two normal words each associated with different 1 bits are combined. The resulting label contains both words.

Word1	1———
Word3	—1——
Word1 Word3	1–1——

Inverse Words

Inverse words are associated with internal bit patterns containing at least one inverse bit. An inverse bit is a bit whose 0 value is associated with the presence of a word *and* whose value is 1 unless the word is present in the label. Inverse words can have one or more bits associated with them. The example below is for the simplest and most common case, where a single 0 bit is associated with a word. When a bit is used inversely, its value in a NULL label must be 1. When such a word is combined with a label containing no words, the resulting label does not contain the word.

Word2	–0———
(NULL)	–1———
(NULL)	–1———

In the following example, two inverse words each associated with different inverse (0) bits are combined. The resulting label contains neither of the words.

Word2	–0—–1—
Word6	–1—–0—
(NULL)	–1—–1—

In the example below, two labels containing the above inverse words are combined. Only the inverse word that appears in both labels appears in the resulting combination.

Word2	–0—–1—
Word2 Word6	–0—–0—
Word2	–0—–1—

Hierarchies of Words

Two words form a hierarchy if their associated relevant bits form a hierarchy (i.e., if one set of bits includes the other). Words in hierarchies can be either normal or inverse words. The following example is the simplest case of a hierarchy of two normal words. In this example, as should be evident from the bits, Word5 is hierarchically above Word4. Therefore, when the two words are combined, the result is the higher of the two words, Word5. Two words in the same hierarchy can never appear together in a label.

Word5	––11––
Word4	––1–––
Word5	––11––

Composite Words

This example is very similar to the above example involving Word1 and Word3, with the difference being that this example contains a third word that is the composite of the other two. Word9 is a composite word whose meaning is "the combination of Word7 and Word8."Such a composite word might be used rather than having the individual words combined to appear in the combination label. In this example, the composite word and the words it combines are a special case of word hierarchies. Therefore, the composite word cannot appear in the same label with either of the words of which it is composite.

Word7	––––1–
Word8	–––––1
Word9	––––11

Non-Hierarchical Composite Words

It is possible to form a composite word without a hierarchy involved. Non-hierarchical composite words are possible for words that have more than one bit associated. In the following example, Word12 is a composite of Word10 and Word11, but has no hierarchical relationship with either word. Therefore, Word12 can appear in the same label with Word10 and Word11. When Word10 and Word11 are combined the resulting label contains all three words.

Word10	1———1–
Word11	–1———1
Word12	———11

A Complex Example

Both normal and inverse words can appear in hierarchies. The example below shows a complex combination of an inverse word and hierarchies. Word13 is a word whose internal representation consists of one normal (1) bit and one inverse (0) bit. Because one of the bits is inverse, its value in any label not containing Word13 will be 1, as shown on the second line of the example. Word14 is a normal word in a hierarchy above Word13. The interesting result of this particular combination of hierarchies and inverse bits is that if Word13 is combined with *any* label that does not contain Word13, the resulting label contains Word14 instead of Word13.

Word13	10———
(any label without Word13)	–1———
Word14	11———

Plan of Paper

The remainder of this document specifies how to construct a standard encodings file for CMWs. Chapter 2, “Structure and Syntax of Encodings File” describes the general structure of an encodings file. Chapter 3, “Classification Encodings” describes how classifications are specified. Chapter 4, “Information Label Encodings” describes how the words that make up information labels are specified. Chapter 5, “Sensitivity Label, Clearance, Channels, and Printer Banner Encodings “describes how the words that make up sensitivity labels, clearances, printer banner and other non-handling caveats for printer banner pages are specified. Chapter 6, “Accreditation Range and Name Information Label Encodings”describes how the system and user accreditation ranges the information labels of classified names are specified. Chapter 7, “General Considerations for Specifying Encodings” discusses general considerations for specifying encodings. Chapter 8, “Enforcing Proper Label Adjudications” describes how to use the encodings to enforce proper label adjudication. Appendix A, “Encodings Specifications Error Messages” describes the error messages that can occur from improper encodings specification. Appendix B, “Annotated Sample Encodings” contains a sample encodings file with annotations to describe what the entries in the file are designed to accomplish. The “Glossary” is provided to define important terms as they are used in this document. An Index of important terms is provided to facilitate reference.

Structure and Syntax of Encodings File

2

An encodings file is a standard operating system file containing free text, and can be created and maintained by any desired editor or word processor, as long as no formatting codes are placed in the encodings file. Case is insignificant throughout the file, and blank lines are always ignored. There is a maximum length for lines in the file of 256 characters. The encodings are comprised mainly of a number of keywords, some of which require a value. Some keywords are required to be present, whereas some are optional. Keywords that require a value always end with an equal sign without a blank before it. The value is separated from the keyword by one or more blanks or tabs, and runs until the first semicolon or the end of the line, whichever comes first. Keywords or keyword / value pairs are separated by semicolons or the end of the line. Keywords themselves can contain blanks, but cannot be split over two lines. There is no maximum length of a keyword value *per se*, but the keyword and its value must fit on the same line. Figure 2-1 shows examples of keywords and keyword / value pairs.

keyword1; keyword2	the second keyword is ended by the end of the line
keyword3= three; keyword4;	the value of keyword3 is “three”; keyword4 is ended by the semicolon before the end of the line
keyword5 = five	this line does not contain a single keyword / value pair, because the = has a blank before it; this line represents a common error in attempting to specify the keyword “keyword5= “

Figure 2-1 Keyword and Value Pairs

Note that if the keyword / value pair sits alone on a line, no terminating semicolon is needed (but is always allowed).

Comments can be placed on any line in the encodings file. Comments can appear anywhere a keyword can start. Comments begin with a * and continue through the end of the line. It is not possible to begin a comment in the middle of a keyword or a value, nor is it possible to have any keywords following a comment on the same line. On lines that do not contain keywords (i.e., required combination, combination constraint, and user accreditation range sensitivity label lines), comments can appear only at the beginning of the line, or following white space on the line.

The encodings are comprised of a version specification and seven mandatory sections: classifications, information labels, sensitivity labels, clearances, channels, printer banners, and accreditation range which must appear in this order. Following these sections can be an optional name information labels section. A number of these sections in turn have subsections. The version specification is the single keyword VERSION=, followed by a character string that identifies this particular version of encodings. An example is:

VERSION= DISTRIBUTED DEMO VERSION

The sections mentioned above each begin with a keyword that ends with a colon, as follows:

CLASSIFICATIONS:
INFORMATION LABELS:
SENSITIVITY LABELS:
CLEARANCES:
CHANNELS:
PRINTER BANNERS:
ACCREDITATION RANGE:
NAME INFORMATION LABELS:

Each section ends when a keyword that starts the next section is found, or at the end of the file.

The INFORMATION LABELS:, SENSITIVITY LABELS:, CLEARANCES:, CHANNELS:, and PRINTER BANNERS: sections all contain a mandatory subsection started with the keyword WORDS:. Furthermore, the

INFORMATION LABELS:, SENSITIVITY LABELS:, and CLEARANCES: sections also have two other mandatory subsections started with the keywords REQUIRED COMBINATIONS: and COMBINATION CONSTRAINTS:

Thus, a complete template for an encodings file is as follows.

```
VERSION=

CLASSIFICATIONS:

INFORMATION LABELS:
  WORDS:
  REQUIRED COMBINATIONS:
  COMBINATION CONSTRAINTS:

SENSITIVITY LABELS:

  WORDS:
  REQUIRED COMBINATIONS:
  COMBINATION CONSTRAINTS:

CLEARANCES:
  WORDS:
  REQUIRED COMBINATIONS:
  COMBINATION CONSTRAINTS:

CHANNELS:
  WORDS:

PRINTER BANNERS:
  WORDS:

ACCREDITATION RANGE:

NAME INFORMATION LABELS:
```

Classification Encodings 3

The CLASSIFICATIONS: section specifies the hierarchical classifications to be used by the system. Each classification is defined in terms of its full name, a short name, an internal integer value, and the initial compartments and markings to be associated with a classification, to be described below. This section is used by the system to convert a human-readable representation of a classification into the internal integer form, and to translate the internal form to a human-readable representation.

The CLASSIFICATIONS: section must contain one or more classification specifications. The specification of a classification starts with a name= keyword, and ends with the next name= keyword or the start of the INFORMATION LABELS: section. Therefore, a name= keyword must be the first keyword to follow the CLASSIFICATIONS: keyword. Other keywords used to define a classification are the sname=, aname=, value=, initial compartments=, and initial markings= keywords. These keywords can appear in any order following a name= keyword. A classification cannot contain the slash (/) or comma (,) or whitespace (space, tab, carriage return, linefeed, formfeed) characters.

The Name= Keyword

The name= keyword is used to define the full name of the classification. The name is taken to begin with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The name can contain blanks. A particular name value should not be specified more than once in the encodings file.

The full name specified is used when producing all human-readable classifications in information labels and on printer banner pages. The full name specified can also be entered by users any time a classification is needed (in sensitivity labels, information labels, and clearances).

The Sname= Keyword

The sname= keyword must be present to define the short or abbreviated name of the classification. The name is taken to begin with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The name can contain blanks. A particular sname value should not be specified more than once in the encodings file. If sname= is specified more than once for the same classification, all specifications except the last are ignored. Hence, sname= should be specified only once.

The short name specified is used when producing human-readable classifications in sensitivity labels and clearances. The short name specified can also be entered by users any time a classification is needed (in sensitivity labels, information labels, and clearances).

The Aname= Keyword

The optional aname= keyword can be used to define an alternate name of the classification. The name is taken to begin with the first non-blank character following the blank after the keyword and continues up to the next semicolon or the end of the line. The name can contain blanks.

The alternate name specified can be entered by users any time a classification is needed (in sensitivity labels, information labels, and clearances).

The Value= Keyword

The value= keyword must be present to specify the internal integer value to be associated with the classification. The smallest integer that can be specified is 0. The maximum size integer that can be specified is 255. The decimal integer specified for the value starts with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. A null integer specification, as in the specification:

value= ;

is taken to be 0. If value= is specified more than once for the same classification name, all specifications except the last are ignored. Hence, value= should be specified only once. A particular value should be specified only once in the encodings file.

The hierarchy among different classifications must be evident from the values specified. A *more sensitive* classification must have a larger value than a *less sensitive* classification.

The Initial Compartments= Keyword

The initial compartments= keyword is optional, and is used to specify the initial, or default, values of compartment bits in an information label, sensitivity label, or clearance containing the associated classification. The initial compartments specification consists of a list of those compartment bits that should be 1 in such a label, with unspecified bits being 0. If no initial compartments= keyword is specified for a particular classification, all initial compartment bits for that classification are taken to be 0.

There are two reasons why the initial compartments associated with a classification would need to be non-zero. The first is the case where it is desired that *all* labels on the system containing that classification must have some normal word present, in which case the 1 compartment bits associated with the word[1] would be specified as 1 in the initial compartment bits. Such a word is called a *default word*, and the compartment bits associated with the word are called *default bits.*

1. The association between words and their compartment bits is made by the **compartments=** keyword in the **INFORMATION LABELS:**, **SENSITIVITY LABELS:**, and **CLEARANCES:** sections of the encodings.

The second is the case where the initial compartments specify which compartment bits are used *inversely*[1]. An inverse compartment bit is a bit whose 0 value is associated with the presence of a word *and* whose value is 1 unless the word is present in the label. Inverse compartment bits are associated with inverse words, whose addition to a label does not increase the sensitivity of the label, but instead either decreases or otherwise changes the sensitivity of the label (i.e., changes at least one bit in the internal form of the label from 1 to 0). Normal (non-inverse) compartment bits are associated with words whose addition to a label increases the sensitivity of the label by changing one or more bits from 0 to 1.

All initial compartment bits that are not inverse bits are considered default bits. All default bits must have default words associated. All words that have associated default bits can have *only* default bits associated.

The specification of compartment bits starts with the first non-blank character following the blank after the initial compartments= keyword, and continues up to the next semicolon or the end of the line. The specification consists of one or more blank-separated subspecifications that consist of either 1) a decimal integer specification of a bit position, numbering bits from the left starting at 0, or 2) a range of such bit positions specified as two decimal integers with a "-" in between. The start of a range must be lower than the end of a range. The maximum bit position allowed is 127, for a total of 128 bits. Table 3-1 contains examples of initial compartments specifications.

Table 3-1 Initial Compartments Specifications

Specification	**Meaning**
initial compartments= 1;	Compartment bit 1 must be on (**1**)
initial compartments= 2-3;	Compartment bits 2 and 3 must be on (**1**)
initial compartments= 1 3;	Compartment bits 1 and 3 must be on (**1**)
initial compartments= 2 4-6;	Compartment bits 2, 4, 5, and 6 must be on (**1**)

1. A prefix can also be used to specify compartment bits to be used inversely. Inverse bits specified by prefix words are called special inverse bits. See "Defining Prefixes And Suffixes" in Chapter 4, "Information Label Encodings, and "Using Prefixes to Specify Special Inverse Compartment and Marking Bits" in Chapter 7, "General Considerations for Specifying Encodings."

If initial compartments= is specified more than once for the same classification, the compartment bits indicated in each specification are taken together to form one composite initial compartments specification. Thus, the two specifications:

initial compartments= 4; initial compartments= 5;

are equivalent to the single specification:

initial compartments= 4-5;

However, for the sake of clarity, at most one initial compartments= keyword should be specified per classification.

The Initial Markings= Keyword

The initial markings= keyword is optional, and is used to specify the initial, or default values of marking bits in an information label containing the associated classification. The initial markings specification consists of a list of those marking bits that should be 1 in such a label, with unspecified bits being 0. If no initial markings= keyword is specified for a particular classification, all initial marking bits for that classification are taken to be 0.

There are two reasons why the initial markings associated with a classification would need to be non-zero. The first is the case where it is desired that *all* information labels on the system containing that classification must have some word (e.g., NOFORN) present, in which case the 1 marking bits associated with the word[1] would be specified as 1 in the initial marking bits. Such a word is called a *default word* and the marking bits associated with the word are called *default bits.*

The second is the case where the initial markings specify which marking bits are used *inversely*[2]. An inverse marking bit is a bit whose 0 value is associated with the presence of a word *and* whose value is 1 unless the word is present in the label. Inverse marking bits are associated with inverse words, whose addition to a label does not increase the sensitivity of the label, but instead either decreases or otherwise changes the sensitivity of the label, (i.e., changes

1. The association between words and their marking bits is made by the **markings=** keyword in the **INFORMATION LABELS:** section of the encodings.

2. A prefix can also be used to specify compartment bits to be used inversely. Inverse bits specified by prefix words are called special inverse bits. See “Defining Prefixes And Suffixes” in Chapter 4, “Information Label Encodings, and “Using Prefixes to Specify Special Inverse Compartment and Marking Bits” in Chapter 7, “General Considerations for Specifying Encodings.”

at least one bit in the internal form of the label from 1 to 0). Normal (non-inverse) marking bits are associated with words whose addition to a label increases the sensitivity of the label by changing one or more bits from 0 to 1.

All initial marking bits that are not inverse bits are considered default bits. All default bits must have default words associated. All words that have associated default bits can have *only* default bits associated.

The specification of marking bits starts with the first non-blank character following the blank after the initial markings= keyword, and continues up to the next semicolon or the end of the line. The specification consists of one or more blank-separated subspecifications which consist of either 1) a decimal integer specification of a bit position, numbering bits from the left starting at 0, or 2) a range of such bit positions specified as two decimal integers with a "-" in between. The start of a range must be lower than the end of a range. The maximum bit position allowed is 127, for a total of 128 bits. Table 3-2 shows examples of initial markings specifications.

Table 3-2 Initial Markings Specifications

Specification	**Meaning**
initial markings= 1;	Marking bit 1 must be on (**1**)
initial markings= 2-3;	Marking bits 2 and 3 must be on (**1**)
initial markings= 1 3;	Marking bits 1 and 3 must be on (**1**)
initial markings= 2 4-6;	Marking bits 2, 4, 5, and 6 must be on (**1**)

If initial markings= is specified more than once for the same classification, the marking bits indicated in each specification are taken together to form one composite initial markings specification. Thus, the two specifications:

initial markings= 11; initial markings= 17;

are equivalent to the single specification:

initial markings= 11 17;

However, for the sake of clarity, at most one initial markings= keyword should be specified per classification.

Information Label Encodings 4

The INFORMATION LABELS: section specifies the words that make up a human-readable representation of an information label, as well as the required combinations and combination constraints on these words. This section is used by the system to convert a human-readable representation of non-classification information label words into the internal bit-string form, and to translate the internal form to a human-readable representation.

The WORDS: subsection, which must appear immediately after INFORMATION LABELS:, specifies each word as well as the prefixes and suffixes needed by some words. The REQUIRED COMBINATIONS: and COMBINATION CONSTRAINTS: subsections follow the WORDS: subsection, in that order.

The Words: Subsection

The WORDS: subsection consists of zero or more word specifications. Each word specification starts with a name= keyword, and ends with the next name= keyword or the start of the REQUIRED COMBINATIONS: subsection. Therefore, a name= keyword must be the first keyword to follow the INFORMATION LABELS: keyword. Any other WORDS: keywords appearing before the first name= keyword are ignored. Other keywords used to define information label words are the sname=, iname=, prefix, suffix, minclass=, maxclass=, ominclass=, omaxclass=, compartments=, markings=, prefix=, suffix=, access related, and flags= keywords. These keywords can appear in any order after a name= keyword. A word cannot have the slash (/), comma (,) or whitespace (space, tab, carriage return, linefeed, formfeed) characters.

Defining Prefixes And Suffixes

Words can have prefixes and suffixes. An example of words that might have a prefix are country names. REL CNTRY1 is an example of the word CNTRY1 that requires the prefix REL. REL CNTRY1/CNTRY2 is an example of two words (CNTRY1 and CNTRY2), both of which require the same prefix. An example of words that might have a suffix are project names, whose suffix might be the word LIMDIS to indicate that distribution of the information is limited to people working on the project. Thus PROJECT X LIMDIS is an example of the word PROJECT X that requires the suffix LIMDIS, and PROJECT X/PROJECT Y LIMDIS is an example of two words that require the same suffix LIMDIS.

It is also possible for a word to require both a prefix and a suffix. However, in such a case, the prefix and suffix must *always* be used together. In other words, if word W requires prefix P and suffix S, word X cannot require prefix P without *also* requiring suffix S.

It should be noted that words themselves can contain blanks (such as PROJECT X above), so simply having a blank in a word is no reason to define it as a word with a prefix. Rather, prefixes and suffixes should be used only when there are multiple words that require the same prefix or suffix.

The Name= Keyword

The name= keyword is used to define the full name of the prefixes, suffixes, and other words. The name is taken to begin with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The name can contain underscores and dashes (-). A particular name value should not be specified more than once per section of the encodings file.

The full name specified is used when producing all human-readable labels. The full name specified can also be entered by users as part of a label.

The Sname= Keyword

The optional sname= keyword assigns an alternative (presumably shorter or abbreviated) name to prefixes, suffixes, and other words. The name is taken to begin with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The name can contain underscores and dashes (-). A particular sname value should not be specified more than once per section of the encodings file. If sname= is specified more than once for the same word, all specifications except the last are ignored. Hence, sname= should be specified at most once.

Any short names specified can be entered by users as part of a label. Short names are never used in producing human-readable labels.

The Iname= Keyword

The optional iname= keyword assigns input names to prefixes, suffixes, and other words. The name is taken to begin with the first non-blank character following the blank after the keyword and continues up to the next semicolon or the end of the line. The name can contain underscores and dashes (-). A particular iname value should not be specified more than once per section of the encodings file. The iname= keyword can be specified any number of times for the same word to specify multiple input names.

A specified iname or input name can be entered by users as part of a label, but are never used in producing human-readable labels. The intended purpose of the iname= keyword is to specify common word misspellings.

note that extra names for the same word can also be specified with an exact alias (a word with markings and compartments identical to a preceding word), as described in Chapter 7, “General Considerations for Specifying Encodings.” Using an iname is preferable to using an exact alias.

Defining Prefix and Suffix Words

All prefixes and suffixes themselves must be defined as words at the beginning of the WORDS: subsection. As with any other word, a prefix or suffix itself can have both long and short names. A prefix is defined by specifying its name with the name= keyword, its optional short name with the sname= keyword, optional input names with iname=, and the keyword prefix. A suffix is defined by specifying its name with the name= keyword, its optional short name with the sname= keyword, optional input names with iname=, and the keyword suffix.

Optionally, prefixes can be specified with compartments= and markings= keywords. These specify special inverse bits. Words that specify a prefix with special inverse bits are called special inverse words. See “Using Prefixes to Specify Special Inverse Compartment and Marking Bits” in Chapter 7, “General Considerations for Specifying Encodings.”

Defining Non-Prefix/Non-Suffix Words

The order in which non-prefix/non-suffix words are specified in the encodings file is extremely important. When translating an internal format of a label to human-readable form, the applicable words will be placed in the human-readable string *in the order in which they appear in the encodings file*. Therefore, the order of the words in the encodings file determines the canonical form of the human-readable representation of the label. By convention, the most important words appear first. Usually, the most important words are those that designate sensitive data, and therefore typically represent compartments, subcompartments, or codewords.

All words must have a name, and can optionally have a short name and multiple input names. If the word requires a prefix, the prefix it requires must be specified with the prefix= keyword. The prefix specified starts with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. Either the short or long name of the prefix can be specified, as long as the prefix is defined at the beginning of the WORDS subsection. If the word requires a suffix, it must have a suffix= keyword. The suffix specified starts with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. Either the short or long name of the suffix can be specified, as long as the suffix is defined at the beginning of the WORDS subsection.

The remaining keywords associated with words define the semantics of the word, rather than the syntax of its human-readable representation. The meaning and specification of each of these remaining keywords are described below.

The Minclass= Keyword

The optional minclass= keyword specifies the minimum classification with which the word should appear in a human-readable label. The classification specified starts with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The classification can be either the short, long, or alternate name of a classification defined in the CLASSIFICATIONS: section. If the minimum classification with which a word can appear is the classification with the lowest value (as defined by the value= keyword), then there is no need to make a minclass= keyword specification.

If a word with an associated minclass is added to a label with a classification below that minclass, the classification in the label is automatically raised to the minclass, assuming the well formedness rules otherwise allow adding the word to the label.

The Ominclass= Keyword

The optional ominclass= keyword specifies the output minimum classification for the word. The output minimum classification is the minimum classification with which the word can be output (i.e., appear in a human-readable representation of a label converted from internal format). The classification specified starts with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The classification can be the short, long, or alternate name of a classification defined in the CLASSIFICATIONS: section. If the output minimum classification with which a word can be associated is the classification with the lowest value (as defined by the value= keyword), then there is no need to make an ominclass= keyword specification.

The distinction between minclass= and ominclass= is subtle but very important. Specifying ominclass for a word prevents that word from appearing in human-readable labels with classifications below the ominclass, *even if the internal representation of the label specifies the word.* A word with an associated ominclass *cannot* be added to a label with a classification below that ominclass, unless the word also has a minclass that is greater than or equal to the ominclass.[1] The following examples shed more light on the differences between ominclass and minclass.

Typically, ominclass= would be specified only for those inverse words associated *only* with inverse bits,[2] when the word—by convention—is not shown in labels below a certain classification. The best example of such a word is a release marking, e.g., REL CNTRY1. The word REL CNTRY1 indicates that the information is releasable to CNTRY1. Therefore, CONFIDENTIAL information that was releasable to CNTRY1 would have a label of CONFIDENTIAL REL CNTRY1. However, note that UNCLASSIFIED information is—by virtue of its not being classified—releasable to CNTRY1. Therefore, the semantics of REL CNTRY1 is such that its internal representation must be present in UNCLASSIFIED labels, yet—by convention—it is not shown in the human-readable representation of the label UNCLASSIFIED. Therefore, specifying an ominclass= CONFIDENTIAL for the word REL CNTRY1 prevents REL CNTRY1 from appearing with UNCLASSIFIED in human-readable labels. In conjunction with specifying the CONFIDENTIAL output minimum classification for REL CNTRY1, the bit patterns that represent the presence of REL CNTRY1 in a label should be specified in the initial compartments and/or markings of all classifications below CONFIDENTIAL.

An ominclass can be specified in conjunction with a minclass, for a variety of reasons. As mentioned above, specifying a minclass equal to the ominclass allows adding the word to a label with a classification below the ominclass. Specifying an ominclass greater than the minclass is a common case, as indicated in the above REL CNTRY1 example, and automatically occurs when an ominclass greater than the lowest classification is specified, but no minclass is specified, in which case the minclass becomes the lowest classification.

1. In this case, the only reason the word can be added is that the **minclass**, being greater than or equal to the **ominclass**, causes the label's classification to be raised when the word is added, such that the classification of the label is greater than or equal to the **ominclass**, so that the word can appear in the label.

2. The most typical case of an inverse word is one associated with only inverse bits. This is the case for all the words of the form **REL XX** in Appendix B, “Annotated Sample Encodings.” However, more complex inverse words are possible. An example is the codeword **bravo4** in Appendix B, “Annotated Sample Encodings.” This codeword is associated with an inverse bit and several non-inverse bits. There is no need to specify an ominclass for **bravo4**, primarily because of the presence of the non-inverse bits in its internal form.

It is meaningful, in some cases, to specify an ominclass below the minclass of the word. The word charlie in Appendix B, “Annotated Sample Encodings.” B illustrates such a case. The word charlie is an inverse word with a minclass of SECRET and an ominclass of CONFIDENTIAL. The internal representation of charlie is specified by UNCLASSIFIED labels. Ignoring the minclass specification, charlie looks very similar to the REL CNTRY1 word described above. However, with the minclass specified as SECRET, charlie can appear only in labels with classifications of SECRET or higher. Thus, UNCLASSIFIED labels have an internal representation that specifies charlie, but the word charlie does not appear in UNCLASSIFIED labels. CONFIDENTIAL labels have an internal representation that does not specify charlie, and charlie cannot appear in such a label. Adding charlie to such a label changes the classification in the label to SECRET. SECRET labels have an internal representation that does not specify charlie but charlie can be added to such a label without changing its classification, assuming the well formedness rules allow adding charlie to the label. If the ominclass for charlie was equal to the minclass instead of being below it, charlie could *not* be added to a confidential label (forcing the label to SECRET, as described above). With the word charlie, the choice of an ominclass of CONFIDENTIAL versus SECRET depends entirely on the desired behavior of the system when a user tries to add charlie to a CONFIDENTIAL label.

The Maxclass= Keyword

The optional maxclass= keyword specifies the maximum classification with which the word should be associated. The classification specified starts with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The classification can be either the short, long, or alternate name of a classification defined in the CLASSIFICATIONS: section. If the maximum classification with which a word can be associated is the classification with the highest value (as defined by the value= keyword), then there is no need to make a maxclass= keyword specification.

The maxclass= keyword must be used with extreme caution. Care must be taken if maxclass= is specified for a word to insure that the classification in a label with the word cannot be raised through combination with a label containing a higher classification. Such a combination must automatically remove the word with the maxclass. Note that both words in Appendix B, "Annotated Sample Encodings" with a maxclass= specification, bravo4 and charlie, are inverse words that are removed upon combination with a label with a higher classification:

- The word bravo4 has a minclass and a maxclass of SECRET, and has a combination constraint that requires bravo4 to stand alone with the classification in a label. Therefore, the only valid label that can contain bravo4 is SECRET BRAVO4. Since bravo4 is associated with inverse marking bit position 12, and is the *only* word to use marking bit 12, the combination of any other label with SECRET BRAVO4 results in the inverse bit being forced to 1, with the net result that the bravo4 word becomes bravo2. These words are explained more fully in Appendix B, "Annotated Sample Encodings".
- The word charlie also has a minclass and a maxclass of SECRET, and has a combination constraint and a required combination that requires charlie to appear with alpha2 and *only* alpha2 in a label. Therefore, the only valid label that can contain charlie is SECRET ALPHA2 CHARLIE. Since charlie is associated with inverse marking bit position 17, and is the *only* word to use marking bit 17, the combination of any other label with SECRET ALPHA2 CHARLIE results in the inverse bit being forced to 1, with the net result that the charlie disappears from the label. These words are explained more fully in Appendix B, "Annotated Sample Encodings".

The Omaxclass= Keyword

The optional omaxclass= keyword specifies the output maximum classification for the word. The output maximum classification is the maximum classification with which the word can be output (i.e., appear in a human-readable representation of a label converted from internal format). The classification specified starts with the first non-blank character following the blank after the keyword and continues up to the next semicolon or the end of the line. The classification can be the long, short, or alternate name of a classification defined in the CLASSIFICATIONS: section. If the output maximum classification with which a word can be associated is the classification with the highest value (as defined by the value= keyword), then there is no need to make an omaxclass= keyword specification.

This keyword supports a marking like EFTO (Encrypt For Transmission Only), which should appear in only UNCLASSIFIED human-readable labels, but is semantically present in all labels with classifications above UNCLASSIFIED. To support EFTO, with markings bit N, the encodings should specify markings bit N as a default bit for classification above UNCLASSIFIED:

```
CLASSIFICATIONS:

name= UNCLASSIFIED;     value= 1;
name= CONFIDENTIAL;     value= 4;     initial markings= N;
name= SECRET:           value= 5;     initial markings= N;
name= TOP SECRET;       value= 6;     initial markings= N;
```

and then specify EFTO as a word with an omaxclass of UNCLASSIFIED:

```
name= EFTO;   omaxclass=UNCLASSIFIED;   MARKINGS= N;
```

With these specifications, EFTO does not appear in human-readable representations of CONFIDENTIAL, SECRET, and TOP SECRET labels, but its internal (bit) representation is present in these labels. With these specifications, if an information label of UNCLASSIFIED EFTO is combined with one of SECRET, the result is SECRET.

The Compartments= Keyword

The optional compartments= keyword is used to specify which compartment bits (if any) must be 1 or 0 if the word is present in a label. For example, if the word is a codeword of a particular compartment, the compartment bit associated with that compartment would also be associated with the codeword.

The specification of compartment bits starts with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The specification consists of zero or more blank-separated subspecifications which consist of either 1) a decimal integer specification of a bit position, numbering bits from the left starting at 0, or 2) a range of such bit positions specified as two decimal integers with a “-” in between. The start of a range must be lower than the end of a range. The maximum bit position allowed is 127, for a total of 128 bits. Each of these subspecifications can be immediately preceded by a ~ (with no blank in between) to indicate that the specified bits must be 0.

The following table shows compartments specification examples.

Table 4-1 Compartments Specifications

Specification	**Meaning**
compartments= 1;	Compartment bit 1 must be on (**1**)
compartments= 2-3;	Compartment bits 2 and 3 must be on (**1**)
compartments= ~4;	Compartment bit 4 must be off (**0**); this would be an inverse compartment bit
compartments= ~5-7;	Compartment bits 5, 6, and 7 must be off (**0**)
compartments= 1 3;	Compartment bits 1 and 3 must be on (**1**)
compartments= ~4 6;	Compartment bit 4 must be off (**0**), and bit 6 must be on (**1**)
compartments= ~4 ~6;	Compartment bits 4 and 6 must be off (**0**)
compartments= 2 4-6;	Compartment bits 2, 4, 5, and 6 must be on (**1**)
compartments= ;	Ignored

The compartments= keyword is of critical importance in implementing the desired label adjudications for the words in a system, because this keyword, along with the markings= keyword, specifies the association between human-readable words and the internal bit representations that are logically "or-ed" together when labels are adjudicated. Chapter 8, "Enforcing Proper Label Adjudications" discusses how the compartments= and markings= (see below) keywords can be used to effect the various types of adjudications described in Chapter 1, "Introduction."

The Markings= Keyword

The optional markings= keyword is used to specify which marking bits (if any) must be 1 or 0 if the word is present in a label. For example, if the word is a codeword, the marking bit(s) associated with that codeword would be specified.

The specification of marking bits starts with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The specification consists of zero or more blank-separated subspecifications which consist of either 1) a decimal integer specification of a bit position, numbering bits from the left starting at 0, or 2) a range of such bit positions specified as two decimal integers with a “**-**” in between. The start of a range must be lower than the end of a range. The maximum bit position allowed is 127, for a total of 128 bits. Each of these subspecifications can be immediately preceded by a ~ (with no blank in between) to indicate that the specified bits must be 0.

The following table shows markings specification examples.

Table 4-2 Markings Specifications

Specification	**Meaning**
markings= 1;	Marking bit 1 must be on (**1**)
markings= 2-3;	Marking bits 2 and 3 must be on (1)
markings= ~4;	Marking bit 4 must be off (**0**); this would be an inverse marking
markings= ~5-7;	Marking bits 5, 6, and 7 must be off (**0**)
markings= 1 3;	Marking bits 1 and 3 must be on (**1**)
markings= ~4 6;	Marking bit 4 must be off (**0**), and bit 6 must be on (**1**)
markings= ~4 ~6;	Marking bits 4 and 6 must be off (**0**)
markings= 2 4-6;	Marking bits 2, 4, 5, and 6 must be on (**1**)
markings= ;	Ignored

The markings= keyword is of critical importance in implementing the desired label adjudications for the words in a system, because this keyword, along with the compartments= keyword, specifies the association between human-readable words and the internal bit representations that are logically “or-ed” together when labels are adjudicated.

Chapter 8, “Enforcing Proper Label Adjudications” discusses how the compartments= and markings= (see above) keywords can be used to effect the various types of adjudications described in Chapter 1, “Introduction.”

The Access Related Keyword

The optional access related keyword, when present, specifies that the word is considered access related and must therefore appear in the warning statement on printed output banner pages. More precisely, the access related keyword should be specified for information label words 1) whose addition to a label *increases* the sensitivity of the label and 2) that do not also appear in sensitivity labels. Banner pages contain warning statements specifying how information is to be protected unless it is manually reviewed and downgraded. Therefore, if an information label word is access related (e.g. NOFORN), it will appear in the banner page warning statement if it is defined with the access related keyword. Figure 4-1 shows the format of an printer banner page example with an access related word denoted.

The Flags= Keyword

The optional flags= keyword, when present, specifies which of 15 flags should be associated with this word. The flags are specified as the numbers 0 through 14, in a manner identical to the specification of compartment or marking bits, although the ~ has no meaning in this context. Flags are not used by the system itself, but could be used by applications specifically written to use them.

Flags might be used to define certain words that appear only in printer banner labels, not in normal labels. Flags could also be used to define certain words that appear only in labels embedded in formal message traffic. See ""Specifying Aliases" in Chapter 7, "General Considerations for Specifying Encodings" for more information on the potential usage of flags.

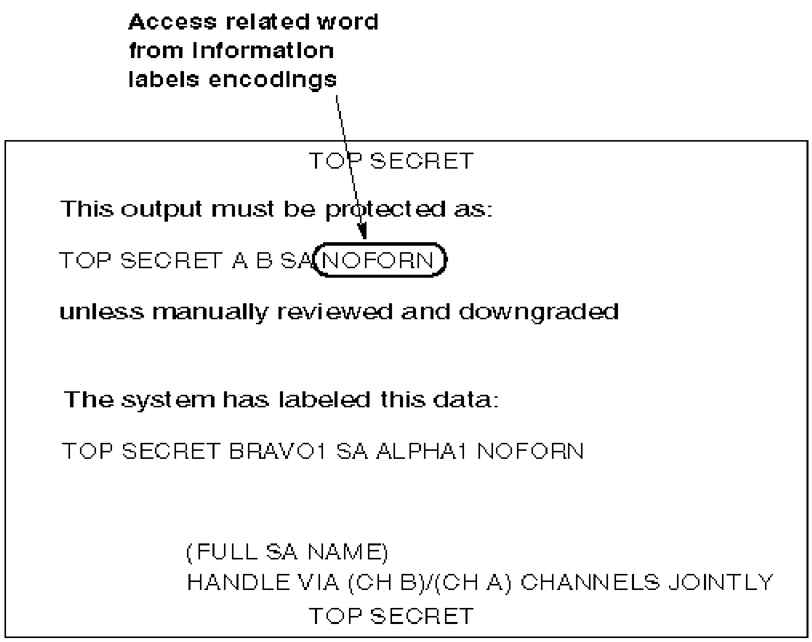


Figure 4-1 Printer banner example denoting access-related word

The Required Combinations: Subsection

The REQUIRED COMBINATIONS: subsection is used to specify any combinations of two words that must always appear together in a human-readable label. This subsection states criteria for well formed labels. This section can contain zero or more required combination specifications. Each required combination is specified on a separate line following the REQUIRED COMBINATIONS: keyword. Each such line should consist of exactly two label words, and is taken to mean that the first word, if present, must be combined with the second word. Any required prefixes or suffixes must also be specified. Any number of lines containing required combinations can be specified. The end of the list is taken to be a line starting with the COMBINATION CONSTRAINTS: keyword.

Note that required combinations are *not* bi-directional. That is, the required combination:

WORD1 WORD2

means that WORD1 cannot appear without WORD2, but does *not* mean that WORD2 cannot appear without WORD1.

Required combinations are automatically enforced during translation from human-readable to internal formats. Thus, given the above example, if the label TS WORD1 were entered, the translation would automatically add WORD2 to the label.

Chapter 7, "General Considerations for Specifying Encodings" discusses some very important considerations concerning the specification of required combinations.

The Combination Constraints: Subsection

The COMBINATION CONSTRAINTS: subsection is used to specify certain kinds of constraints on the combination of words in a human-readable label. This subsection states criteria for well formed labels. This section can contain zero or more required combination specifications. Whereas the REQUIRED COMBINATIONS: subsection described above specified which label words *must* be combined, the COMBINATION CONSTRAINTS: subsection specifies what label words *cannot* be combined. Each combination constraint is specified on a separate line following the COMBINATION CONSTRAINTS: keyword. Each such line should consist of exactly one constraint specification (see below for a discussion of continuation lines). Any required prefixes or suffixes must also be specified. Any number of lines containing combination constraints can be specified. The end of the list is taken to be a line starting with the SENSITIVITY LABELS: keyword.

Combination constraint lines can take one of the following three forms:

1. WORDS1 ! WORDS2
2. WORDS1 & WORDS2
3. WORDS1 &

In all of the cases above, WORDS1 and WORDS2 represent either a single word or multiple words separated by " | " (think of | as meaning "or"). Note that blanks or tabs must appear on both sides of the !, &, and | characters in combination constraints. Form 1 specifies that none of the words WORDS1 can be combined with any of the words WORDS2. Form 2 specifies that any of the words WORDS1 can be combined *only* with any of the words WORDS2 (it specifies nothing about what the words WORDS2 can or cannot be combined with). Form three specifies that any of the words WORDS1 cannot be combined with any other words.

Because combination constraint lines can get very long, and because each constraint must be on its own line, combination constraint lines can be continued onto the next real line of the file. A combination constraint line is continued by using a \ as the last character of the line. A blank must precede the \. The next non-blank line after a line ending with a \ is taken to be a logical continuation of that line. A single word (including its prefix and/or suffix) cannot be separated with a \; each word must appear together on the same actual line. Thus, using words from Appendix B, "Annotated Sample Encodings."

REL CNTRY3 ! REL CNTRY1 | \

REL CNTRY2

is a valid continuation of a combination constraint, whereas

REL CNTRY3 ! REL \

CNTRY1 | REL CNTRY2

is invalid because the prefix REL and the base word CNTRY1 were split across two lines.

Combination constraints need not be specified among words in a hierarchy. In other words, if WORD2 is in a hierarchy over WORD1, then these two words can never appear together in a label. There is no need to additionally specify a combination constraint to prevent their combination.

Similarly, any non-hierarchical mutually exclusive words need not be specified in combination constraints. For example, consider the specification of the following two words:

name= WORD7; markings= 6 ~7;

name= WORD8; markings= ~6 7;

The bit encodings of these two words make them mutually exclusive, yet they are not hierarchically related. Such words are automatically prevented from appearing together in a label, even if a combination constraint is not specified.

Conversely, combination constraints cannot prevent the combination of words if the bit encodings of those words force their combination, such with a non-hierarchical composite word (see Chapter 8, “Enforcing Proper Label Adjudications”). For example, consider the three words:

name= WORD12; markings= 6 7;

name= WORD10; markings= 0 6;

name= WORD11; markings= 1 7;

Note that there are no hierarchies among these words. If both WORD10 and WORD11 are in a label, WORD12 is also automatically present. A combination constraint cannot be used to keep WORD12 from appearing in a label with either WORD10 or WORD11, as long as WORD10 and WORD11 can be combined.

Chapter 7, “General Considerations for Specifying Encodings” discusses some very important considerations concerning the specification of combination constraints.

Sensitivity Label, Clearance, Channels, and Printer Banner Encodings 5

The encodings described in this chapter have similar structures.

Sensitivity Labels Encodings

The SENSITIVITY LABELS: section specifies the words that make up a human-readable representation of a sensitivity label, as well as the required combinations and combination constraints on these words. This section is used by the system to convert a human-readable representation of non-classification sensitivity label words into the internal bit-string form, and to translate the internal form to a human-readable representation.

In all cases, the SENSITIVITY LABELS: section must associate words with exactly the same compartment bits as the INFORMATION LABELS: section. Moreover, for every word in the INFORMATION LABELSi.information label;: section with associated normal (non-inverse) compartment bits, there can be no word in the SENSITIVITY LABELS: section whose associated normal compartment bits dominate but do not equal the compartment bits of the information labeli.information label; word, unless the sensitivity label word is an alias. Also, for each inverse compartment word in the SENSITIVITY LABELS: section, there must be a corresponding inverse compartment word in the INFORMATION LABELSi.information label;: section whose compartment bits are dominated by the sensitivity label word's compartment bits, and whose markings contain no normal bits.

The SENSITIVITY LABELS: section has a structure identical to the INFORMATION LABELS: section with the following exceptions:

1. the markings= keyword cannot be specified because sensitivity labels—by definition—do not contain markings; and
2. the access related keyword cannot be specified because all components of sensitivity labels are—by definition—access related, so this keyword would be redundant.

All other keywords described above for the information labels section work in the sensitivity labels section, with the same purpose, rules, restrictions, and caveats.

Another difference between the INFORMATION LABELS: section and the SENSITIVITY LABELS: section is that the conventional order within the encodings file of sensitivity label words in the intelligence community is with the *least* sensitive words first.

Chapter 7, “General Considerations for Specifying Encodings” discusses some very important considerations concerning the specification of sensitivity label encodings.

Clearance Encodings

The CLEARANCES: section specifies the words that make up a human-readable representation of user's clearances, as well as the required combinations and combination constraints on these words. This section is used by the system to convert a human-readable representation of non-classification clearance words into the internal bit-string form, and to translate the internal form to a human-readable representation.

The CLEARANCES: section has a structure identical to the SENSITIVITY LABELS: section. In fact, in most cases, the CLEARANCES: section will be identical to the SENSITIVITY LABELS: section. A separate CLEARANCES: section is provided to add flexibility to the system in case there are any different required combinations or combination constraints on clearances than on sensitivity labels, or if—by convention—clearance compartment names are slightly different than corresponding sensitivity label compartment names.

In all cases, the CLEARANCES: section must associate words with exactly the same compartment bits as the INFORMATION LABELS: and SENSITIVITY LABELS: sections. Moreover, for every word in the SENSITIVITY LABELSi.information label;: section with associated normal (non-inverse) compartment bits, there can be no word in the CLEARANCES: section whose associated normal compartment bits dominate but do not equal the compartment bits of the sensitivity labeli.information label; word, unless the clearance word is an alias. Also, for each inverse compartment word in the CLEARANCES: section, there must be a corresponding inverse compartment word in the SENSITIVITY LABELSi.information label;: section whose compartment bits are dominated by the clearance word's compartment bits.

The encodings example in Appendix B, "Annotated Sample Encodings" illustrates a case where the CLEARANCES: combination constraints are different than the SENSITIVITY LABELS: combination constraints. This example occurs when release compartments are encoded in sensitivity labels. In the Appendix B, "Annotated Sample Encodings" example, REL CNTRY1 and REL CNTRY2 are treated as release compartments. The meaning of having REL CNTRY1 in a sensitivity label is that the data is releasable to system users whose clearance contains the compartment REL CNTRY1.

In the clearance section, the prefix REL is called NATIONALITY: to indicate that the application of this word in a clearance specifies the nationality of the user. Therefore, the meaning of having REL CNTRY2 in a sensitivity label is that the data is releasable to system users whose clearance contains the compartment NATIONALITY: CNTRY2. The meaning of having REL CNTRY1/CNTRY2 in a sensitivity label is that the data is releasable to system users whose clearance contains either NATIONALITY: CNTRY1 or NATIONALITY: CNTRY2. Therefore, it is perfectly valid for a sensitivity label to contain both REL CNTRY1 and REL CNTRY2.

However, such is not the case for a clearance. The meaning of NATIONALITY: CNTRY1 in a clearance is that the user is a citizen of CNTRY1. Similarly, the meaning of NATIONALITY: CNTRY2 in a clearance is that the user is a citizen of CNTRY2. However, in most systems, it would make no sense for a user to be treated as a citizen of more than one country, so having both NATIONALITY: CNTRY1 and NATIONALITY: CNTRY2 in a clearance would be invalid and is prevented by the clearance combination constraint:

NATIONALITY: CNTRY1 ! NATIONALITY: CNTRY2

Chapter 7, "General Considerations for Specifying Encodings" discusses some very important considerations concerning the specification of clearances encodings.

Channels Encodings

The CHANNELS: section specifies the words that make up a human-readable representation of the HANDLE VIA... handling caveats that must be included in printer banner pages. This section is used by the system only to translate the internal form of sensitivity labels to the appropriate human-readable handling channel caveat. The example in Figure 5-1 shows the format of a printer banner page with the handling channel caveat denoted. Since this section is not used for translating human-readable handling caveats into the internal format, there is no need for the REQUIRED COMBINATIONS: and COMBINATION CONSTRAINTS: subsections. Therefore, the CHANNELS: section contains a single subsection: WORDS:.

The CHANNELS: WORDS: subsection has a structure similar to the SENSITIVITY LABELS: WORDS: subsection. However, the sname=, iname=, minclass=, and maxclass= keywords have no meaning or purpose for handling caveats, and are therefore ignored.

In all cases, the CHANNELS: section cannot associate words with compartment bits not associated with words in the INFORMATION LABELS:, SENSITIVITY LABELS:, and CLEARANCES: sections.

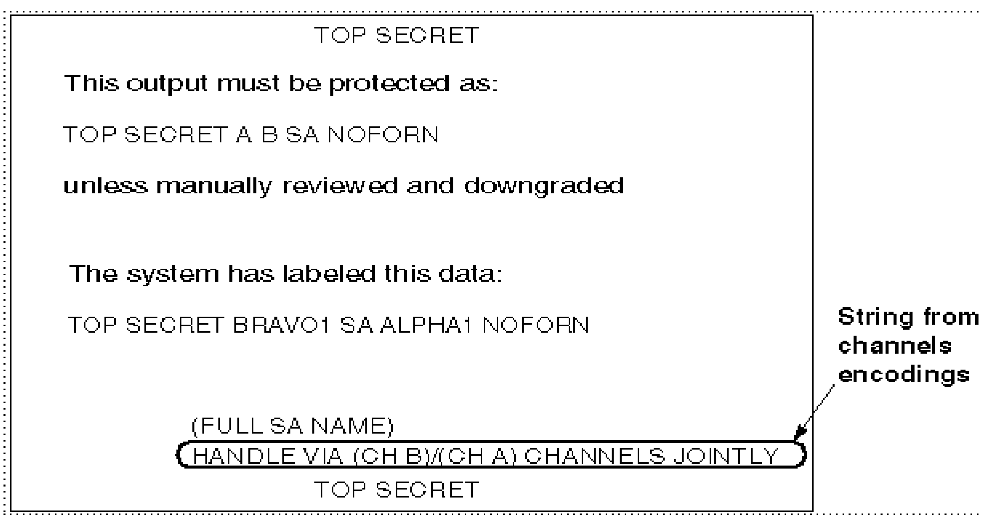


Figure 5-1 Printed banner example denoting channels string

Printer Banner Encodings

The PRINTER BANNERS: section specifies the words that make up a human-readable representation of caveats other than HANDLE VIA... caveats that must be included in printer banner pages. Whereas the CHANNELS: section specifies HANDLE VIA... caveats based typically on the main (non-sub) compartments in the sensitivity label, the PRINTER BANNERS: section specifies caveats based on any of the compartments or markings bits. This section is used by the system only to translate the internal form of a sensitivity label along with the markings from the corresponding information label to the appropriate human-readable non-handling channel caveat. The example in Figure 5-2 shows the format of a printer banner page with the non-handling channel caveat denoted. Since this section is not used for translating human-readable handling caveats into the internal format, there is no need for the REQUIRED COMBINATIONS: and COMBINATION CONSTRAINTS: subsections. Therefore, the PRINTER BANNERS: section contains a single subsection: WORDS:.

The PRINTER BANNERS: WORDS: subsection has a structure identical to the CHANNELS: WORDS: subsection, except that the markings= keyword is also allowed, for reasons described above. Note that the sname=, iname=, minclass=, and maxclass= keywords have no meaning or purpose for printer banner caveats, and are therefore ignored or not allowed.

In all cases, the PRINTER BANNERS: section cannot associate words with compartment bits not associated with words in the INFORMATION LABELS:, SENSITIVITY LABELS:, and CLEARANCES: sections, and cannot associate words with marking bits not associated with words in the INFORMATION LABELS: section.

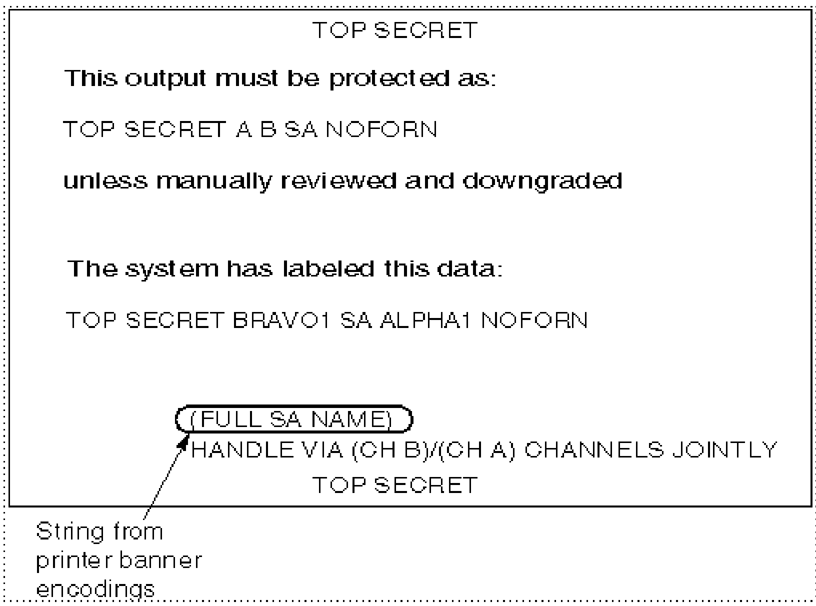


Figure 5-2 Printer banners encodings string

Accreditation Range and Name Information Label Encodings 6

The ACCREDITATION RANGE: section specifies the system and user accreditation ranges as well as a number of system constants related to the system accreditation range. The system accreditation range is the set of sensitivity labels that the system as a whole can process. It is specified by a minimum sensitivity label and a maximum sensitivity label. The minimum sensitivity label is directly specified in the ACCREDITATION RANGE: section, whereas the maximum sensitivity label is deduced from the classifications and word specifications in the encodings file. The user accreditation range is a subset of the system accreditation range, and contains those sensitivity labels that normal (non-authorized) users of the system can set (i.e., those sensitivity labels at which users can create subjects or objects, or to which users can change existing sensitivity labels).

The ACCREDITATION RANGE: section consists of one or more user accreditation range specifications, followed by the specification of various system accreditation range-related constants, the minimum clearance, the minimum sensitivity label, and the minimum protect as classification.

This section explains the concept of a user accreditation range in terms of examples, then explains how to specify the user accreditation range and system accreditation range-related constants.

User Accreditation Range Examples

The user accreditation range of a system is the set of sensitivity labels at which normal users are intended to be able to operate. The user accreditation range; is conceptually specified as a list (or set) of sensitivity labels. In this manner, certain sensitivity labels can be left out of the list, as will be the case for many systems.

Note that the restriction the user accreditation range; places on sensitivity labels does *not* apply to either clearances or information labels.

For example, consider a system that processes TS along with compartments A, B, and C. The complete list of possible sensitivity labels (and hence the largest possible user accreditation range;) for such a system is:

TS	TS A	TS B	TS C
TS A B	TS A C	TS B C	TS A B C

However, a more realistic user accreditation range; for such a system might be:

TS A B	TS A C	TS A B C

In this example, compartments B and C can be processed only in combination with A, and A cannot be processed alone.

Specifying the User Accreditation Range

The encodings for classifications and sensitivity label words specify which potential sensitivity labels are well formed. Based on these encodings alone, every potential sensitivity label may not be well formed. Given the compartments A, B, and C from the above example, if compartment C has a REQUIRED COMBINATION of C A, then compartment C can never appear in a well formed label without compartment A. Thus, the well formed sensitivity labels in the example would be:

TS	TS A	TS B
TS A B	TS A C	TS A B C

The user accreditation range specification is stated in terms of the set of well formed sensitivity labels.

Whereas the above examples dealt with the classification TS only, specifying a user accreditation range; in general requires specifying the compartment combinations valid with each classification in the user accreditation range. Furthermore, specifying the valid compartment combinations, in the case where *all* well formed combinations are not valid, can be done by specifying those combinations *are* valid, or by specifying those combinations that *are not* valid.

There must be one or more user accreditation range; specifications. There should be one specification for each classification that appears in a sensitivity label in the user accreditation range;. Each specification consists of a classification= keyword followed by *one* of the keywords all compartment combinations valid, all compartment combinations valid except:;, or only valid compartment combinations:;;, as described below.

The Classification= Keyword

The classification= keyword should be specified for each classification in the user accreditation range;. The keyword is followed by a valid classification (short, long, or alternate name) from the CLASSIFICATIONS: section, and one of the three keywords described below. The classification name is taken to begin with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The name specified must match either the short, long, or alternate name of one of the classifications specified in the classifications section of the encodings file.

The All Compartment Combinations Valid Keyword

The all compartment combinations valid keyword specifies that all well formed compartment combinations are valid along with the classification specified by the preceding classification= keyword. Note that *only* those compartment combinations that are well formed according to the encodings in the CLASSIFICATIONS: and SENSITIVITY LABELS: sections are valid. For example, if the SENSITIVITY LABELS: COMBINATIONS CONSTRAINTS: specifies

A ! B

Then compartment B cannot appear in a sensitivity label along with compartment A, regardless of the classification or the user accreditation range; specification.

An example of a user accreditation range; specification using the all compartment combinations valid keyword is:

classification= TS; all compartment combinations valid;

The All Compartment Combinations Valid Except: Keyword

The all compartment combinations valid except: keyword specifies that all compartment combinations are valid along with the classification specified by the preceding classification= keyword, except those that are listed, one per line, on the lines that follow until the next keyword. Each subsequent line (other than blank lines and comment lines) should contain exactly one sensitivity label, up until a line containing a classification= or minimum clearance= keyword is found. At least one sensitivity label should be specified.

Each sensitivity label specified must be well formed according to the encodings in the CLASSIFICATIONS: and SENSITIVITY LABELS: sections. Furthermore, each sensitivity label must be in canonical form. A sensitivity label is in canonical form if it begins with the sname of a classification followed by the name of zero or more SENSITIVITY LABELS: WORDS:, in the order in which the words appear in the SENSITIVITY LABELS: section.

The sensitivity labels are used to specify compartment combinations only; the classification in the sensitivity label is ignored after validity checking. However, the classification in each sensitivity label must be the same as the classification= keyword that precedes it.

A specification of the realistic user accreditation range; from the example above using the all compartment combinations valid except: keyword is:

classification= TS; all compartment combinations valid except:

TS

TS A

TS B;

The Only Valid Compartment Combinations: Keyword

The only valid compartment combinations: keyword specifies that no compartment combinations are valid along with the classification specified by the preceding classification= keyword, except those that are listed, one per line, on the lines that follow until the next keyword. Each subsequent line (other than blank lines and comment lines) should contain exactly one sensitivity label, up until a line containing a classification= or minimum clearance= keyword is found. At least one sensitivity label should be specified.

Each sensitivity label specified must be well formed according to the encodings in the CLASSIFICATIONS: and SENSITIVITY LABELS: sections. Furthermore, each sensitivity label must be in canonical form. A sensitivity label is in canonical form if it begins with the sname of a classification followed by the name of zero or more SENSITIVITY LABELS: WORDS:, in the order in which the words appear in the SENSITIVITY LABELS: section.

The sensitivity labels are used to specify compartment combinations only; the classification in the sensitivity label is ignored after validity checking. However, the classification in each sensitivity label must be the same as the classification= keyword that precedes it.

A specification of the realistic user accreditation range; from the example above using the only valid compartment combinations: keyword is:

classification= TS; only valid compartment combinations:

TS A B

TS A C

TS A B C

Appendix B, “Annotated Sample Encodings” contains more examples of each of the above types of user accreditation range; specifications.;;

Specifying System Accreditation Range-Related Constants

Following the specification of each classification in the user accreditation range;, a number of system accreditation range-related system constants are specified with the keywords minimum clearance=, minimum sensitivity label=, and minimum protect as classification=, as described below.

The Minimum Clearance= Keyword

Following the user accreditation range; specifications is the minimum clearance= keyword. This keyword is followed by a specification of the minimum clearance of any user on the system. This minimum clearance will be enforced by the system when setting user's clearances. The clearance is taken to begin with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The clearance must be well formed and in canonical form. A clearance is in canonical form if it begins with the sname of a classification followed by the name of zero or more CLEARANCES: WORDS:, in the order in which the words appear in the CLEARANCES: section. This clearance must be valid according to the CLEARANCES: encodings, but does not have to conform to the clearance combination constraints (and is therefore not well formed), and does not have to be in the user accreditation range;.

The Minimum Sensitivity Label= Keyword

Following the minimum clearance= keyword is the minimum sensitivity label= keyword. This keyword is followed by a specification of the minimum sensitivity label to be used on the system. This minimum sensitivity label forms the low end of the system accreditation range, and will be enforced by the system when setting sensitivity labels. The sensitivity label is taken to begin with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The sensitivity label must be well formed and in canonical form. A sensitivity label is in canonical form if it begins with the sname of a classification followed by the name of zero or more SENSITIVITY LABELS: WORDS:, in the order in which the words appear in the SENSITIVITY LABELS: section. The minimum sensitivity label does not have to be in the user accreditation range;. However, the minimum sensitivity label must be dominated by the minimum clearance.

The Minimum Protect As Classification= Keyword

Following the minimum sensitivity label= keyword is the minimum protect as classification= keyword. Following this keyword is the minimum classification at which all system output is to be protected unless it is manually reviewed and downgraded. The classification name is taken to begin with the first non-blank character following the blank after the keyword, and continues up to the next semicolon or the end of the line. The name specified must match either the short, long, or alternate name of one of the classifications specified in the classifications section of the encodings file. The minimum protect as classification cannot be greater than the classification in the minimum clearance.

Figure 6-1 is an example of how the minimum protect as classification will be used by the system when producing printed output. The system puts the maximum of the minimum protect as classification and the classification in the sensitivity label of the data being printed at the top and bottom of the banner page, and in the warning statement about how the output must be protected.

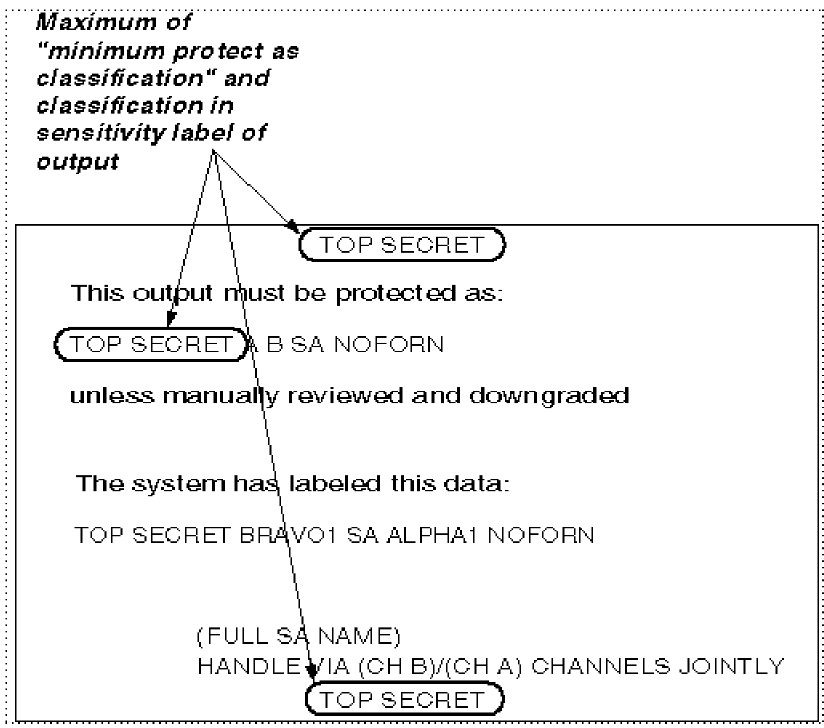


Figure 6-1 Printer banner example denoting minimum protect as classification usage

Name Information Label Encodings

In some encodings files, some classification and word names themselves may be classified. If so, the optional NAME INFORMATION LABELS: section specifies the information label of names specified in the encodings. An information label can be specified for every classification and word name, including prefixes and suffixes. This section is entirely optional. If not included, the information labels of all names will be assumed to be the minimum information label. Even if this section is included, it is necessary to specify information labels for only those names whose information label is other than the minimum information label.

The NAME INFORMATION LABELS: section consists of zero or more information label specifications. Each information label specification consists of one or more name= keywords followed by one il= specification. All of the names specified are assigned the single information label specified. The names can be classification names, snames, or anames, or word names or snames (including prefix or suffix names or snames).

Each information label specified must be well formed according to the encodings in the CLASSIFICATIONS: and INFORMATION LABELS: sections. Furthermore, each information label must be in canonical form. An information label is in canonical form if it begins with a classification name (not sname or aname) followed by the name of zero or more INFORMATION LABELS: WORD:, in the order in which the words appear in the INFORMATION LABELS: section.

The following examples, drawn from the same encodings in Appendix B, "Annotated Sample Encodings," serve to illustrate the usage of name information label specifications. For example, the information label specification:

```
name= bravo1;            il= confidential b;
```

assigns the information label confidential b to the name bravo1, which appears only in the INFORMATION LABELS: section. The specification:

```
name= alpha1;
name= alpha2;
name= alpha3;            il= confidential a;
```

assigns the information label confidential a to the names alpha1, alpha2, and alpha3, which appear only in the INFORMATION LABELS: section. The specification:

name= sa; il= top secret sa;

assigns the information label top secret sa to the name sa, which appears in the INFORMATION LABELS:, SENSITIVITY LABELS:, and CLEARANCES:, sections. Finally, the specification:

name= (CH A); il= confidential a;

assigns the information label confidential a to the name (CH A), which appears multiple times in the CHANNELS: section.

General Considerations for Specifying Encodings

7

The sections above describe how each section of the encodings file should be specified. However, when specifying an encoding file, there are some important considerations regarding the relationships among entries in the various sections to be kept in mind. This section describes such considerations.

The Minimum Information Label

The minimum possible information label is indirectly specified in the encodings file, and care must be taken to insure that it is well formed. The minimum information label's classification is the lowest specified classification. The minimum information label's compartments equal the initial compartments of the lowest specified classification, but with all inverse compartment bits set to 0. The minimum information label's markings equal the initial markings of the lowest specified classification, but with all inverse marking bits set to 0.

The human-readable form of the minimum information label contains the lowest specified classification along with those default words specified by the initial compartments and/or markings of this classification, but *without* any inverse words specified by the initial compartments and/or markings.

The Maximum Sensitivity Label

The maximum possible sensitivity label is indirectly specified in the encodings file and care must be taken to insure that it is well formed. The maximum sensitivity label's classification is the highest specified classification. The maximum sensitivity label's compartments contain 1 bits for every compartment bit referenced in *any* initial compartments= specification or *any* compartments= specification in the encodings file, and 0 bits for all other bits.

Consistency of Word Specification among Different Types of Labels

Many words must be specified as being components of all three types of labels: information labels, sensitivity labels, and clearances. In fact, in most cases, words that appear in sensitivity labels also appear in clearances[1] and information labels.[2] When the same word appears in multiple types of labels, extreme care must be taken to ensure that the words are specified as consistently as possible in each label. In particular, the words should have the same minclass, maxclass, and the same required combinations and combination constraints with respect to combinations with words that also appear in multiple labels. Any inconsistencies may have undesired results.

For example, consider a system that facilitates downgrading the sensitivity label of an object by setting it equal to the classification and compartments of the object's information label. Consider also the encodings in Figure 7-1. With these encodings, CONFIDENTIAL A would be a valid information label, and SECRET A B would be a valid sensitivity label, both for the same object. However, if the system's “downgrade sensitivity label to information label classification and compartments” function is performed, the sensitivity label would become CONFIDENTIAL A. Such a sensitivity label is invalid for two reasons: 1) the word A in a sensitivity label has a minimum classification of SECRET, and 2) the word A requires the word B in a sensitivity label. Consistently encoding the word A for both information and sensitivity labels would have avoided this problem.

1. Sometimes the word may have a different name or prefix in a clearance, but has the same meaning as the sensitivity label word because it is associated with the same compartment bits. See Chapter 5, “Sensitivity Label, Clearance, Channels, and Printer Banner Encodings” for a discussion of why a clearance word might have a different prefix than an otherwise equivalent sensitivity label word.

2. Sometimes the word may have a different name in an information label, but has the same meaning as the sensitivity label word because it is associated with the same compartment bits. In other cases, the word may not appear in an information label, but one or more other words that specify the same compartment bit pattern do appear.

```
CLASSIFICATIONS:
    NAME= CONFIDENTIAL;   SNAME= C; VALUE= 4;
    NAME= SECRET;         SNAME= C; VALUE= 5;
INFORMATION LABELS:
    WORDS:
        NAME= A;   COMPARTMENTS= 2;  MINCLASS= C;
        NAME= B;   COMPARTMENTS= 3;  MINCLASS=  C;
    REQUIRED COMBINATIONS:
    COMBINATION CONSTRAINTS:
SENSITIVITY LABELS:
    WORDS:
        NAME= A;   COMPARTMENTS= 2;  MINCLASS= S;
        NAME= B;   COMPARTMENTS= 3;  MINCLASS= C;
    REQUIRED COMBINATIONS:
        A  B
    COMBINATION CONSTRAINTS:
```

Figure 7-1 Inconsistent encodings example

Mandatory Access Control Considerations When Encoding Words

Before encoding each word, the meaning of the word with respect to national policy must be determined. If national policy dictates that mandatory access control (MAC) must be performed based on the word (which is the case for compartments, subcompartments, SAPs, and SAPIs), or if a policy decision is made to treat a word as a compartment,[1] then the word should be associated with compartment bits in the clearances and sensitivity labels sections of the encodings file, and possibly in the information label section as well. Such a word is called a *MAC word.* Instead, if the word does not directly enter into MAC decisions, but implies some other word that does, the word would appear only in information labels, be associated with both compartments and

1. For example, release markings on which it has been decided to perform MAC, such as REL CNTRY1 and REL CNTRY2 in Appendix B, “Annotated Sample Encodings.”

markings, and is called a *MAC-related word*. Finally, if the word has absolutely nothing to do with MAC, the word would appear only in information labels, be associated with only markings, and be called a *non-MAC word*.

Encoding MAC Words

As mentioned above, words on which mandatory access control must be performed must be associated with compartment bits, and must appear in the CLEARANCES: and SENSITIVITY LABELS: sections, and possibly in the CHANNELS:, PRINTER BANNERS:, and INFORMATION LABELS: sections. The word would appear in the CHANNELS: section if the word represents a handling channel. The word would appear in the PRINTER BANNERS: section if the word requires any special printer banner marking other than a handling channel caveat. The word would appear in the INFORMATION LABELS: section if it is desired that the word appear in information labels. It is conceivable that a mandatory access control word not appear in information labels, but that a *codeword* that implies the word could appear instead.

When encoded in the clearances:, sensitivity labels:, channels: and PRINTER BANNERS: sections, a mandatory access control word would be associated with only compartment bits. When encoded in the INFORMATION LABELS: section, the word could have associated both compartment and marking bits.

Consider the word A in Appendix B, “Annotated Sample Encodings.” This word, which appears with the name A in the clearances: and sensitivity labels: sections and the name (CH A) in the channels: section, is associated with compartment bit 0 being 1. Note that the word A in the information labels: section is also associated with compartment bit 0 being 1, but additionally has a marking bit associated, for a reason discussed below.

Some words that represent compartments, and would typically be expected to have only compartment bits associated, nonetheless require association with marking bits in information labelsB to establish a hierarchy with other information label words. In the INFORMATION LABELS: section, A has marking bit 7 associated. The purpose of marking bit 7 in the specification of A is to establish a hierarchy with A above WNINTEL (which is associated *only* with marking bit 7). The reason for this hierarchy is that the word WNINTEL was deemed unnecessary along with any word that directly represents or implies a compartment. The hierarchy prevents WNINTEL from appearing in a label with any such word.

Encoding MAC-Related Words

Words that are not directly used for MAC, yet imply the presence of a compartment or other MAC word, are encoded in the information labels: section using both compartment and marking bits. This situation typically occurs when there are multiple words, sometimes called *codewords*, associated with a compartment. In such a case, users are cleared for the compartment as a whole, not for the individual codewords. However, the presence of the codeword in an information label implies that the data is in the compartment. In such a case, the codeword must have a compartment bit associated to identify the compartment, but must additionally have one or more marking bits associated to distinguish the word as a codeword (as opposed to a MAC word) and to differentiate among the multiple codewords. An example of this case appears in Appendix B, “Annotated Sample Encodings” with the words alpha1, alpha2, and alpha3. All three words are associated with compartment bit 0 (and hence the compartment A), but additionally have marking bits associated. This particular pattern of marking bits determines which of the three codewords are present.

It is also possible to encode MAC-related words in the PRINTER BANNERS: section if desired. There is no such example in Appendix B, “Annotated Sample Encodings.”

Encoding Non-MAC-Related Words

Words having nothing to do with MAC, either directly as compartments or indirectly as codewords, are encoded in the information labels: section using only marking bits. In Appendix B, “Annotated Sample Encodings,” the word WNINTEL is such a word.

It is also possible to encode non-MAC-related words in the PRINTER BANNERS: section if desired. There is no such example in Appendix B, “Annotated Sample Encodings.”

Using Initial Compartments and Markings to Specify Inverse Compartment and Marking Bits

The intended usage of the initial compartments and initial markings specifications when used to specify inverse compartment or marking bits is described below. Potentially, the initial compartments and initial markings keywords can be used more flexibly than described below, but any such usage should be *very* carefully scrutinized to determine that the desired security and labeling properties are represented.

The intended usage is described below in terms of inverse words of the form REL XX, whose meaning is that data associated with the word is releasable to XX, where XX could represent a country, an organizational affiliation, or even a person.

1. It is often true that data of the lowest classification (or possibly classifications) is always releasable without restrictions. Therefore, the internal format of labels with the lowest classification would specify that all such REL XX words be logically present. Having these words present means that their associated bits should be 0. Therefore, it is not necessary to specify these inverse bits as initial compartments or markings in the lowest classification.

2. For classifications greater than the lowest, it would therefore typically follow that the bits associated with the REL XX words should be 1, because data whose label is one of these classifications *standing alone* is not releasable. Therefore, the *same* initial compartments (representing inverse bits)—if any—are intended to be specified for all classification values other than the lowest, and the *same* initial markings (representing inverse bits)—if any—are intended to be specified for all classification values other than the lowest.

3. When allocating compartment and marking bits, careful consideration must be given to deciding how many inverse bits should ever be needed *throughout the life of the system*. All inverse bits ever anticipated to be used on the system should be specified in the initial compartments and markings specifications even if they are not used initially for any marking words in the encodings. The best way to describe why this preallocation of inverse bits is necessary is to show what happens if inverse bits are not preallocated. Assume that marking bit 11 is encoded as an inverse bit whose meaning is REL CNTRY3 (as specified in Appendix B, “Annotated Sample Encodings”).

Further assume that there are no other inverse marking bits and that marking bit 12 is assigned no meaning. Since marking bit 11 is the only inverse marking bit, the initial markings specification would be:

initial markings= 11;

If the system was used with the encoding file in this condition, a large number of information labels would be stored on the system (and on backup tapes) with marking bit 12 (and all other unused bits) having the value 0. Then, if it is later decided that REL CNTRY4 must be encoded using an inverse marking bit and bit 12 (or any other unused bit) is chosen, then all data previously stored on the system would automatically be treated as REL CNTRY4 because that would be the meaning of marking bit 12 being 0. Of course, all of the data would *not* be releasable to CNTRY4, such that *all* data on the system (and on backup tapes) would have to be relabeled. Therefore, it is best to preallocate some range of bits as inverse bits when the encodings file for a system is first loaded. Then, preallocated unused inverse bits can be later assigned meaning without the need to relabel.

Note that the above discussion covers the simplest and most common usage of inverse bits. More complex usage of inverse bits is possible, and needed in some instances. As an example, see the hypothetical bravo4 codeword in Appendix B, “Annotated Sample Encodings.”

Using Prefixes to Specify Special Inverse Compartment and Marking Bits

The intended usage of prefix words that specify compartments or markings is to specify special inverse bits that allow special inverse words. Special inverse words are words that specify a prefix that in turn specifies compartments or markings. The intended purpose and usage of special inverse bits and special inverse words is best described by the example below.

Special inverse words can be used to implement the ORiginator CONtrolled (ORCON) handling caveat with organizations to which the ORCON data can be released specified in the label. For example, given that three organizations use a particular system (ORG1, ORG@, and ORG3), the encodings to handle ORCON for these three organizations might look as follows. Only the SENSITIVITY LABELS words are shown in this example.

SENSITIVITY LABELS:

WORDS:

name=ORCON RELEASABLE TO; sname=OR; compartments=1-4; prefix;

name=ORCON;	minclass=C;	compartments=1-4;	
name=ORG1;	minclass=C;	compartments=~1 4;	prefix=OR;
name=ORG2	minclass=C;	compartments=~2 4;	prefix=OR;
name=ORG3;	minclass=C;	compartments=~3 4;	prefix=OR;

In this example, ORG1, ORG2, and ORG3 are special inverse words, each of which requires the prefix ORCON RELEASABLE TO. This prefix specifies compartments bits 1-4, which are therefore special inverse bits. Bit one is for ORG1, bit 2 for ORG2, bit 3 for ORG3, and bit 4 has meaning of ORCON. If only ORCON RELEASABLE TO ORG1 is present in a label, then bit 1 would be off, and bits 2-4 would be on. If only ORG1 is present in a label, then bit 1 would be off, and bits 2-4 would be on. If only ORCON RELEASABLE TO ORG2 is present in a label, then bit 2 would be off, and bits 1, 3, and 4 would be on. If only ORCON RELEASABLE TO ORG3 is present in a label, then bit 3 would be off, and bits 1, 2, and 4 would be on. If ORCON RELEASABLE TO ORG1/ORG2 is present in a label, then bits 1 and 2 would be off and bits 3 and 4 would be on, and so on. The word ORCON, which dominates the three other words, is not an inverse word. If it appears in a label, the data so labeled is not releasable to any of the three organizations.

Note that a label that does not contain any of the above words has bits 1-3 off and is therefore releasable to all organizations, and has bit 4 off and is therefore not ORCON data. Thus, with the same words as above for information labels, data with an information label of SECRET ORCON RELEASABLE TO ORG1 when combined with data with an information label of TOP SECRET, would become TOP SECRET ORCON RELEASEABLE TO ORG1. Special inverse words can be specified using markings bits also.

Unlike regular inverse bits, special inverse bits should not be preallocated to allow for future usage. Special inverse bits can be safely added to a running system without preplanning.

Choosing Names

The names chosen in the classifications:, information labels:B, sensitivity labels:, and clearances: sections are extremely important. In general, it is best if all short and long names *within* each of the above sections are unique. However, because of the way prefix and suffix words are handled by the system, there are two exceptions to this general rule.

1. A suffix and a non-prefix/non-suffix word can have the same name. This is possible because you can look at labels with both such names and tell them apart. For example, consider suffix SF, word W that requires suffix SF, and regular word SF. The label TS SF contains the regular word SF, because there is no word that requires the suffix SF preceding the SF. The label TS W SF contains the suffix SF, because the word W immediately precedes the SF. Finally, the label TS SF W SF contains both the regular word SF and the suffix SF.
2. A word that requires a prefix and a non-prefix/non-suffix word can have the same name, as long as the non-prefix/non-suffix word is specified before the word that requires the prefix. This is possible because you can look at labels with both such names and tell them apart. For example, consider word W that requires prefix P, and regular word W. The label TS W contains the regular word W, because there is no prefix before the W. The label TS P W contains the word W that requires the prefix P, because the prefix is present. Finally, the label TS W P W contains both the regular word W and the prefix-requiring word W.

Obviously, use of either of these exceptions should be avoided if at all possible because of the probable confusion that will occur.

There are two additional considerations in specifying names.

1. Classification names should never be the same as information label, sensitivity label, or clearance names.
2. If the same name appears in both the sensitivity labels: and clearances: sections, the words with this name should refer to the same compartment, and should therefore have an identical specification in the encodings file.

Specifying Aliases

A word in the information label, sensitivity label, or clearance sections whose specified compartment or marking bits include all of the bits of one or more words *above in the encodings* is called an *alias*. The simplest case of an alias is a word that duplicates the compartment and marking bit specifications of the word above it. Such an alias—in effect—simply adds more names to the word above it. The word WARNING in Appendix B, "Annotated Sample Encodings" is such an alias for the word WNINTEL. Using an input name (iname=) is the preferred method of associating more than two names with a word. See "The Iname= Keyword" in Chapter 4, "Information Label Encodings."

A more complex type of alias is a word whose compartment and/or marking bits includes bits specified in *multiple* words that appear above it. The word SYSHI in Appendix B, "Annotated Sample Encodings" is an example of this type of alias. Entering SYSHI is the same as entering the following words from Appendix B, "Annotated Sample Encodings": CC SB bravo1 bravo3 SA alpha1 project X/project Y LIMDIS ORCON org x/org Y D/E all eyes NOFORN.

Aliases can be used while entering labels or adding to labels (e.g., by entering +alias to add alias to an existing label), but cannot be used for removing words from labels (e.g., by entering -alias to remove alias from an existing label) and will never appear in output labels (assuming the alias and the words being aliased have the same flags= specification). For example, given the above alias WARNING for the word WNINTEL, the following table shows how the label TOP SECRET can and cannot be modified using the alias.

Table 7-1 Modifying with Alias

LABEL	**TYPED CHANGE**	**COMMENTS**
TOP SECRET	+WARNING	Alias added to existing label; aliased word (WNINTEL) will appear in label instead of alias itself
TOP SECRET WNINTEL	-WARNING	Produces an error because "WARNING" is not in the label
TOP SECRET WNINTEL	-WNINTEL	Aliased word will be removed
TOP SECRET		Aliased word was removed

Alias words can be combined with flags to produce aliases that can optionally be used in output labels. The system does not use the flags feature, but applications can be specifically written to use this feature. As an example, consider the case where you have a word that normally appears as NORMAL NAME, but that must, under certain conditions, appear in labels as ALTERNATE NAME. This could be accomplished with the following encodings:

name= NORMAL NAME; markings= 34;

name= ALTERNATE NAME; markings= 34;flags= 1;

Under normal circumstances NORMAL NAME would appear in labels, but if the translation software is explicitly told to use only words with flag 1, then ALTERNATE NAME would appear in labels. See [DDS-2600-6215-91] for information on how applications can use the flags feature in this manner.

Avoiding "Loops" In Required Combinations

Extreme care must be taken in specifying required combinations to ensure that there are no "loops" in the specifications. A "loop" occurs when, through a series of required combination specifications, a word requires *itself*. The simplest case of a loop is:

A B

B A

whereby word A requires word B, which in turn requires word A. Such a specification makes no sense. If words A and B must always appear together, why are they encoded as separate words? A more complex case of a loop occurs in the following specification:

A B

B C

C A

whereby word A requires word B, which in turn requires word C, which in turn requires word A.

Visibility Restrictions for Required Combinations

The fact that information labels must be dominated by their associated sensitivity label, and that sensitivity labels specified by a user must be dominated by that user's clearance, places some constraints on what words can be added to certain labels. For example, if adding a word to an information label raises the information label such that it is no longer dominated by the associated sensitivity label, then that word is not *visible* in the information label. Similarly, if adding a word to a sensitivity label raises the sensitivity of the label such that it is no longer dominated by the associated user's clearance, then that word is not *visible* in the sensitivity label.

It is important that any word required by another word in a required combination be *visible* whenever the requiring word is visible. For example, given the required combination:

A B

which means A requires B, word B must be visible whenever word A is visible. If B were not visible at some point when A was visible, a situation could occur whereby A could legally be added to a label, were it not for the fact that doing so would require also adding B, which would violate a dominance relationship. Such a situation must be prevented by careful construction of required combinations. There are no restrictions on required combinations of words with only marking bits (i.e., no compartment bits) associated, because marking bits do not participate in the dominance relationships mentioned above.

One practical ramification of this restriction is that 1) sensitivity label required combinations should not be more restrictive than the equivalent clearance restrictions, and that 2) information label required combinations should not be more restrictive than the equivalent sensitivity label restrictions. A concrete example of this problem can be taken from the sample encodings in Appendix B, "Annotated Sample Encodings."

Consider the SA and CC compartments in the CLEARANCES: and SENSITIVITY LABELS: encodings. The REQUIRED COMBINATIONS: in both of these sections are:

SB B

SA A

Now, consider the same where an additional required combination is added to *only* the SENSITIVITY LABELS: encodings:

SA CC

This additional required combination, which makes the sensitivity label required combinations more restrictive than those for clearances, specifies that if SA is present in a sensitivity label, CC must also be present. Now consider the case of a user with the clearance TS A B SA SB. Such a clearance is perfectly valid according to the encodings, but such a user can never put SA in a sensitivity label because SA requires CC, yet the user is not cleared for CC.

Relationships between Required Combinations and Combination Constraints

It is possible for a valid required combination, when combined with a valid combination constraint, to yield an anomalous situation. Consider the required combination:

A B

combined with the combination constraint:

A ! B

These specifications say that word A requires word B, yet words A and B cannot be combined. Such contradictory specifications must be avoided.

Restrictions on Specifying Information Label Combination Constraints

Information label combination constraints are used by the labeling software to ensure that no invalid combinations of words are allowed to be specified in a single information label. However, any two valid information labels can be combined by the system by bitwise or-ing the compartment and marking bits. Thus if a combination constraint is specified that (using examples from Appendix B, “Annotated Sample Encodings”) subcompartment SA and subcompartment SB cannot be combined, an inconsistent situation has arisen. The inconsistency is that SA and SB cannot be combined by entering them in a single information label, yet two separate information labels, each with one of the subcompartments, can be combined to produce a new information label with both subcompartments. Therefore, to avoid such inconsistencies, you should never specify any combination constraints that are not automatically enforced on combinations by the encodings.

Examples of constraints automatically enforced on combinations by the encodings abound when considering inverse words. If two inverse words IW1 and IW2 are constrained not to be combined with the combination constraint:

IW1 ! IW2

then you can be assured that IW1 and IW2 can never be put together as a result of the combination of two labels. Why? Because inverse words combine by having only those inverse words in both of the labels being combined appear in the resulting label. Therefore, if both IW1 and IW2 cannot appear in any single information label, then no combination of information labels can combine IW1 and IW2 together.

Thus, you can be assured of avoiding inconsistencies if only inverse words are used in ! constraints and in the left hand side of & constraints.

Modifying Encodings Already Used by the System

Extreme care must be taken when modifying an encodings file that has already been loaded and run on a CMW system. The reason for concern is the fact that once the system has run with the encodings, many objects will become labeled with sensitivity labels and information labels that are well formed with respect to the encodings loaded. If the encodings are subsequently changed, it is possible that the existing labels will no longer be well formed unless care is taken. Changing the bit patterns associated with words will cause existing objects whose labels contain the words to have possibly invalid labels. Raising the minimum classification or lowering the maximum classification associated with words will likely cause existing objects whose labels contain the words to no longer be well formed.

Therefore, changes to encodings that have already been used should generally be limited to adding new classifications or words, or changing the *names* of existing words only. However, as described above, it is important to reserve extra inverse bits when the encodings file is first created to allow for later expansion of the encodings to incorporate new inverse words. If an inverse word is added not using reserved inverse bits, all existing objects on the system will erroneously have labels that include the new inverse word.

Consistency of Default Word Specification

A *default word* is a word whose presence is specified by the initial compartments and initial markings associated with a classification value. In other words, a default word appears in all labels containing the classification(s) whose initial compartments and markings specify the presence of the word.

As with all other words, an output minimum classification can be specified (ominclass=) with a default word, in which case the word will appear in human-readable labels at or above the output minimum classification only. Also, a minimum classification can be specified (minclass=) with a default word, as long as the minimum classification is less than or equal to each classification for which the word is default. For example, the following encodings would be in error.

CLASSIFICATIONS:

name= SECRET; sname= S; value= 5; initial markings= 3;

name= TOP SECRET; sname= TS; value= 6; initial markings= 3;

INFORMATION LABELS:

WORDS:

name= word1; markings= 3; minclass= TS;

The error is that word1 is a default word for the classification SECRET, but has a minimum classification of TOP SECRET, which is greater than SECRET.

Care must be taken in the specification of default words to ensure consistency between the default words specified and any combination constraints involving a default word. If a combination constraint prevents a default word from being combined with a second word, then the second word should not be specified as a default word for the same classifications for which the first word is default. For example, the following encodings would be in error.

CLASSIFICATIONS:

name= SECRET; sname= S; initial markings= 3 4;

INFORMATION LABELS:

WORDS:

name= word1; markings= 3;

name= word2; markings= 4;

REQUIRED COMBINATIONS:

COMBINATION CONSTRAINTS:

word1 ! word2

The error is that word1 and word2 are both default words, but are constrained not to be combined together.

Enforcing Proper Label Adjudications

8

This section describes how to use compartments and/or markings to effect proper label adjudication by the system. Each label adjudication example described in Chapter 1, “Introduction” is expanded below using examples in the proper format for inclusion in an encodings file. Only marking bits are used in the examples below, although compartment bits could alternatively have been used in the examples. From the standpoint of label adjudication, there is no difference between compartment bits and marking bits.

Normal Words

Figure 8-1 shows the encoding of a normal (non-inverse) word not necessarily in a hierarchy. A single marking (or compartment) bit is necessary, and this bit should not be specified as an initial marking (or compartment) bit[1]. This example is the simplest and most common example of the association between a human-readable word and the internal bit format. Normal words can also have multiple associated bits, and may be in hierarchies with other words.

```
name= word1;  markings= 0;
```

Figure 8-1 Normal Words

1. Except in the rare case that this word is a default word that should *always* be present with some classification.

Appendix B, “Annotated Sample Encodings” has the following examples of normal information label words: CC, SB, bravo1, bravo2, bravo3, B, alpha1, alpha2, alpha3, A, project x LIMDIS, project y LIMDIS, ORCON org x, ORCON org y, D/E, all eyes, p1 eyes only, p2 eyes only, WNINTEL, and NOFORN.

Inverse Words

Figure 8-2 shows the encoding of an inverse word that is not necessarily in a hierarchy. A single marking (or compartment) bit is necessary, and this bit must be specified as an initial marking (or compartment) bit, as described in Chapter 3, “Classification Encodings.” This example is the simplest and most common example of the association between a human-readable inverse word and the internal format. Inverse words can also have multiple associated bits, and may be in hierarchies with other words.

```
name= word2;   markings= ~1;
```

Figure 8-2 Inverse Words

Appendix B, “Annotated Sample Encodings” contains examples of inverse information label words: bravo4, charlie, REL CNTRY1, REL CNTRY2, REL CNTRY3.

Hierarchies of Words

The establishment of hierarchies among words is an extremely important feature of the translation software, because the software enforces that two words in the same hierarchy cannot appear together in a label. The software infers a hierarchy between two words W1 and W2 whenever *both* of the following two conditions are true: 1) if the bits explicitly specified for W1 includes all of the bits explicitly specified for W2, and 2) if the values of the bits specified for W1 dominates the values of the same bits for W2, with unspecified bits in W2 taken to be 0. The following series of examples highlights these concepts.

Figure 8-3 shows the encoding of two normal words in a hierarchy, with word5 hierarchically above word4. This is the most common way to encode a hierarchy of normal words, with 1) the words in decreasing hierarchical order and 2) without any 0 bits explicitly encoded (using a ~). It is unnecessary to explicitly specify that for word4 bit 4 must be 0 (using ~4) because bit 4 was not specified as 1 in the initial marking bits. Note that if bit 4 is not specified as 0 for word4, that word4 must appear after word5 in the encodings. If the order of the two words were reversed, a label with bits 3 and 4 as 1 would have word4 shown in the human-readable label, because although both the word4 and word5 bits match the internal format, word4 would be encountered first by the translation software.

```
name= word5;   markings= 3  4;
name= word4;   markings= 3;
```

Figure 8-3 Two Normal Words in a Hierarchy

Figure 8-4 is identical to the above example, except it demonstrates that while the ~4 is not necessary, it is a valid alternative representation. Note that bit 4 is *not* an inverse bit, and therefore need not be specified as 1 in the initial marking bits.

```
name= word5;  markings= 3  4;
name= word4;  markings= 3 ~4;
```

Figure 8-4 Normal Words Alternative Representation

Figure 8-5 is identical to the above example, but has the order of the words reversed. This example works properly because the ~4 is present. As above, note that bit 4 is *not* an inverse bit, and therefore need not be specified as 1 in the initial marking bits.

```
name= word4;  markings= 3 ~4;
name= word5;  markings= 3  4;
```

Figure 8-5 Normal Words Reversed Order

Figure 8-6 shows a hierarchy of inverse words, with word12 above word13 in the hierarchy. In this example, both bits 3 and 4 are inverse bits, and are specified as 1 in the initial marking bits.

```
name= word12;  markings= 3 ~4;
name= word13;  markings= ~3  ~4;
```

Figure 8-6 Hierarchy of Inverse Words

Figure 8-7 does not represent any hierarchy between the two words. Bit 4 is an inverse bit, and is specified as 1 in the initial marking bits. Word12 is not in a hierarchy above word13 because the bits specified for word12 (3) do not include those specified for word13 (3 and 4). Word13 is not in a hierarchy above word12 because the values of the bits specified for word13 (1 and 1 for bits 3 and 4, respectively) do not dominate the values of the same bits for word12 (1 and 0, with the 0 for bit 4 implicitly assumed).

```
name= word12;  markings= 3;
name= word13;  markings= ~3  ~4;
```

Figure 8-7 No Hierarchy

It is always possible to specify all bits involved in hierarchies (leaving no bits implicitly 0). Allowing bits to be implicitly assumed to be 0, as the the first example above, is a convenience that can be used only for hierarchies—or portions of hierarchies—involving normal bits. If an inverse bit appears in a word in a hierarchy, all words *above* that word in the hierarchy must have the inverse bit explicitly specified.

Appendix B, "Annotated Sample Encodings" contains the following examples of information label words in hierarchies: CC, SB, bravo1, bravo2, bravo3, bravo4, B, alpha1, alpha2, alpha3, A, all eyes, p1 eyes only, p2 eyes only, WNINTEL, NOFORN, REL CNTRY1, REL CNTRY2, and REL CNTRY3. Appendix B, "Annotated Sample Encodings" also contains a graphical representation of the hierarchical relationships among these words.

Composite Words

Figure 8-8 shows a composite word, word9, which is a composite of words word7 and word8. This example works assuming that bits 6 and 7 are not specified as 1 in the initial marking bits. Functionally, with this encoding, if word8 is added to a label with word7 already present, both words are automatically replaced by word9. Note that having the composite word precede the other words is critical in this example. If instead word9 was below word7 and word8, it would never appear in human-readable labels translated from internal labels; instead, word7 word8 would appear, and word9 would become an alias for word7 word8.

```
name= word9;  markings= 6  7;
name= word7;  markings= 6;
name= word8;  markings=    7;
```

Figure 8-8 Composite Words

Figure 8-9 shows an alternative method for encoding a composite word. In this example, because of the presence of the explicit 0 bits (~6 and ~7) in the encodings of word7 and word8, the order of the words does not matter. Note that the 0 bits are *not* inverse bits, and therefore need not be specified as 1 in the initial marking bits.

```
name= word7;  markings= 6 ~7;
name= word8;  markings= ~6 7;
name= word9;  markings= 6  7;
```

Figure 8-9 Composite Word Alternative Method

From an adjudication standpoint, this alternative is identical to the previous example. However, there is one important functional difference between this example and the preceding one in the way that the translation software will work. In the preceding example, if word8 is added to a label containing word7 (or vice versa), the label changes to contain word9 instead of word7 or word8. But in this example, the explicit specification of the 0 bits makes the encodings of word7 and word8 mutually exclusive, so (as explained in Chapter 7, “General Considerations for Specifying Encodings”) they must be prevented from appearing together by a combination constraint, for example:

word7 ! word8

Therefore, adding word8 to a label that already contains word7 does *not* replace both words with word9 as in the above example; instead, the addition of word8 is ignored. Note that in both examples, adding word9 to a label with either word7 or word8 replaces the word with word9.

Appendix B, "Annotated Sample Encodings" contains the following examples of composite words: all eyes, which is a composite of p1 eyes only and p2 eyes only.

Non-Hierarchical Composite Words

Figure 8-10 shows a non-hierarchical composite word, word12, which is a composite of words word10 and word11. This example works assuming that all specified bits are not specified as 1 in the initial marking bits. Functionally, with this encoding, if word11 is added to a label with word10 already present (or vice verse), word12 is automatically added to the label. Non-hierarchical composite words must *always* preceed the words of which they are composite. Note that a combination constraint cannot prevent a non-hierarchical composite word from being combined with a word of which it is a composite.

```
name= word12;  markings= 6 7;
name= word10;  markings= 0 6;
name= word11;  markings= 1 7;
```

Figure 8-10 Non-Hierarchical Composite Word

A Complex Example

Combinations of the above types of words can yield more complex examples. Figure 8-11 shows two words in a hierarchy, one of which is inverse. This example has the interesting adjudication effect that if the inverse word (word13) is combined with *any* label that does not contain the word, the result is that the inverse word is replaced by the word above it in the hierarchy (word14). Because word13 is inverse, bit 1 must be specified as 1 in the initial marking bits.

```
name= word13;  markings= 8 ~9;
name= word14;  markings= 8  9;
```

Figure 8-11 Two Words in Hierarchy with Inverse Word

Appendix B, "Annotated Sample Encodings" contains the following examples of complex information label words similar to this example: bravo4, which is an inverse word below bravo2 in a hierarchy.

Encodings Specifications Error Messages

A

This appendix documents the error messages produced by the software described in *Compartmented Mode Workstation Labeling: Source Code and User Interface Guidelines* [DDS-2600-6215-91] when errors in the encodings file are encountered. If the encodings file itself is not found, the message

> Encodings file “XXX” not found.

is produced, where XXX is the name of the encodings file on the system. All other possible error messages are listed below in boldface in alphabetical order, followed by an indented description of the conditions under which the error can occur. In all error messages that contain actual text from the encodings, XXX is used to indicate such text. If a line longer than the maximum allowed (N) is found in the encodings file, the message

> <<<Line longer than N characters>>>

will appear after Found instead: in one of the error messages below containing Found instead:. Where appropriate, error messages contain the line number in the encodings where the error appears or the line just after where the error appears.

ACCREDITATION RANGE CLASSIFICATION “XXX” not found.

> This error occurs if the text following a classification= keyword does not represent a classification long, short, or alternate name in the CLASSIFICATIONS: section.

ACCREDITATION RANGE specifier “XXX” is invalid.

This error occurs if the keyword following a classification= keyword is not one of the following keywords: ALL COMPARTMENT COMBINATIONS VALID EXCEPT:;, ALL COMPARTMENT COMBINATIONS VALID, or ONLY VALID COMPARTMENT COMBINATIONS:;;.

Can't allocate NNN bytes for checking labels.

This catastrophic error occurs if the system cannot dynamically allocate the memory it needs to process the encodings. It does not mean there was any kind of error made in specifying the encodings.

Can't allocate NNN bytes for encodings.

This catastrophic error occurs if the system cannot dynamically allocate the memory it needs to store the encodings. It does not mean there was any kind of error made in specifying the encodings. It could theoretically occur because of a *very very* long encodings file.

Can't find ACCREDITATION RANGE CLASSIFICATION specification. Found instead: XXX.

This error occurs when no classification= keyword follows the ACCREDITATION RANGE: keyword.

Can't find ACCREDITATION RANGE specification. Found instead: “XXX”.

This error occurs when the ACCREDITATION RANGE: keyword does not follow the printer banners section.

Can't find any CLASSIFICATIONS NAME specification. Found instead: “XXX”.

This error occurs if no classifications are specified.

Can't find CHANNELS WORDS specification. Found instead: “XXX”.

This error occurs when the WORDS: keyword does not immediately follow the CHANNELS: keyword.

Can't find CLASSIFICATIONS specification. Found instead: “XXX”.

This error occurs if the CLASSIFICATIONS: keyword is not found after the VERSION= keyword.

Can't find CLEARANCES COMBINATION CONSTRAINTS specification. Found instead: “XXX”.

This error can occur *only* if end of file is reached before the CLEARANCES: COMBINATION CONSTRAINTS: keyword is found.

Can't find CLEARANCES REQUIRED COMBINATIONS specification. Found instead: "XXX".

This error occurs when a keyword not expected in a clearance WORDS: subsection is found, and that keyword is not REQUIRED COMBINATIONS:, which would begin the next subsection. This error typically occurs when a keyword is misspelled, or particularly when a blank is placed before a = in an otherwise valid keyword.

Can't find CLEARANCES WORDS specification. Found instead: "XXX".

This error occurs when the WORDS: keyword does not immediately follow the CLEARANCES: keyword.

Can't find INFORMATION LABELS COMBINATION CONSTRAINTS specification. Found instead: "XXX".

This error can occur *only* if end of file is reached before the INFORMATION LABELS: COMBINATION CONSTRAINTS: keyword is found.

Can't find INFORMATION LABELS REQUIRED COMBINATIONS specification. Found instead: "XXX".

This error occurs when a keyword not expected in an information label WORDS: subsection is found, and that keyword is not REQUIRED COMBINATIONS:, which would begin the next subsection. This error typically occurs when a keyword is misspelled, or particularly when a blank is placed before a = in an otherwise valid keyword.

Can't find INFORMATION LABELS specification. Found instead: "XXX".

This error occurs in the INFORMATION LABELS: keyword does not follow the classifications section.

Can't find INFORMATION LABELS WORDS specification. Found instead: "XXX".

This error occurs when the WORDS: keyword does not immediately follow the INFORMATION LABELS: keyword.

Can't find MINIMUM CLEARANCE specification. Found instead: "XXX".

This error occurs if the minimum clearance= keyword does not appear after the accreditation range specifications in the ACCREDITATION RANGE: section.

Can't find MINIMUM PROTECT AS CLASSIFICATION specification. Found instead: “XXX”.

This error occurs if the minimum protect as classification= keyword does not appear after the minimum sensitivity label= keyword in the ACCREDITATION RANGE: section.

Can't find MINIMUM SENSITIVITY LABEL specification. Found instead: “XXX”.

This error occurs if the minimum sensitivity label= keyword does not appear after the minimum clearance= keyword in the ACCREDITATION RANGE: section.

Can't find PRINTER BANNERS specification. Found instead: “XXX”.

This error occurs when the PRINTER BANNERS: keyword does not follow the channels section.

Can't find PRINTER BANNERS WORDS specification. Found instead: “XXX”.

This error occurs when the WORDS: keyword does not immediately follow the PRINTER BANNERS: keyword.

Can't find SENSITIVITY LABELS COMBINATION CONSTRAINTS specification. Found instead: “XXX”.

This error can occur *only* if end of file is reached before the SENSITIVITY LABELS: COMBINATION CONSTRAINTS: keyword is found.

Can't find SENSITIVITY LABELS REQUIRED COMBINATIONS specification. Found instead: “XXX”.

This error occurs when a keyword not expected in a sensitivity label WORDS: subsection is found, and that keyword is not REQUIRED COMBINATIONS:, which would begin the next subsection. This error typically occurs when a keyword is misspelled, or particularly when a blank is placed before a = in an otherwise valid keyword.

Can't find SENSITIVITY LABELS WORDS specification. Found instead: “XXX”.

This error occurs when the WORDS: keyword does not immediately follow the SENSITIVITY LABELS: keyword.

Can't find VERSION specification. Found instead: “XXX”.

This error occurs if the first keyword found in the encodings is not the VERSION= keyword.

Classification “XXX” does not have a VALUE.

This error occurs when a classification name= keyword is specified without a corresponding value= keyword. Classifications must have values specified.

Classification “XXX” does not have an SNAME.

This error occurs when a classification name= keyword is specified without a corresponding sname= keyword. Classifications must have both long and short names specified.

Classification “XXX” has an invalid VALUE: “XXX” (max is N).

This error occurs when a classification value= keyword specifies a value greater than the maximum value allowed (N).

End of file not found where expected. Found instead: “XXX”.

This error occurs if there is extraneous text at the end of the encodings file.

In ACCREDITATION RANGE, classification “XXX”: invalid sensitivity label “XXX”.

This error occurs if a sensitivity label specified after an ALL COMPARTMENT COMBINATIONS VALID EXCEPT: keyword or an ONLY VALID COMPARTMENT COMBINATIONS: keyword in the ACCREDITATION RANGE: section cannot be understood as a valid sensitivity label based on the encodings. This error would typically occur if the sensitivity label contained an invalid word or classification.

In ACCREDITATION RANGE, classification “XXX”: No sensitivity labels allowed after ALL COMPARTMENT COMBINATIONS VALID.

This error occurs if a classification= keyword or the minimum clearance= keyword does not follow an ALL COMPARTMENT COMBINATIONS VALID keyword in the accreditation range: section.

In ACCREDITATION RANGE, classification “XXX”: SENSITIVITY LABEL “XXX” not in canonical form. Is YYY what was intended:

This error occurs if a sensitivity label specified after an ALL COMPARTMENT COMBINATIONS VALID EXCEPT: keyword or an ONLY VALID COMPARTMENT COMBINATIONS: keyword in the ACCREDITATION RANGE: section, while understood, is not in canonical form. This additional canonicalization check ensures that no errors are made in specifying the accreditation range. For example, given the sample encodings in Appendix B, “Annotated Sample Encodings.” if the sensitivity label C SA were specified, the system would understand the label, but would automatically convert it to TS A SA, because the SA subcompartment has a minimum classification of TS, and requires compartment A. However, the fact that the label had to be converted means it was not in canonical form. A human-readable label is in canonical form if it can be converted into internal format and coverted back to human-readable format with the two human-readable formats being identical. In the error messasge, YYY is the canonical form of the sensitivity label represented by XXX.

In ACCREDITATION RANGE, classification “XXX”: Classification in sensitivity label “XXX” must be “XXX”.

This error occurs during the specification of the accreditation range for a particular classification when a sensitivity label specified for that classification has a different classification.

In ACCREDITATION RANGE, classification “XXX”: Duplicate sensitivity label “XXX”.

This error occurs if a sensitivity label specified after an ALL COMPARTMENT COMBINATIONS VALID EXCEPT: keyword or an ONLY VALID COMPARTMENT COMBINATIONS: keyword in the ACCREDITIATION RANGE: section is a duplicate of a sensitivity label previoulsy specified after the same keyword.

In ACCREDITATION RANGE: Invalid MINIMUM CLEARANCE “XXX”.

This error occurs if the clearance specified after the minimum clearance= keyword in the ACCREDITATION RANGE: section cannot be understood as a valid clearance based on the encodings. This error would typically occur if the clearance contained an invalid word or classification.

In ACCREDITATION RANGE: invalid MINIMUM PROTECT AS CLASSIFICATION “XXX”.

This error occurs if the classification specified after the minimum protect as classification= keyword in the ACCREDITATION RANGE: section does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In ACCREDITATION RANGE: Invalid MINIMUM SENSITIVITY LABEL “XXX”.

This error occurs if the sensitivity label specified after the minimum sensitivity label= keyword in the ACCREDITATION RANGE: section cannot be understood as a valid sensitivity label based on the encodings. This error would typically occur if the sensitivity label contained an invalid word or classification.

In ACCREDITATION RANGE: MINIMUM CLEARANCE “XXX” not in canonical form. Is YYY what was intended?

This error occurs if the clearance specified after the minimum clearance= keyword in the ACCREDITATION RANGE: section while understood, is not in canonical form. This additional canonicalization check ensures that no errors are made in specifying the minimum clearance. For example, given the sample encodings in Appendix B, “Annotated Sample Encodings,” of the clearance C SA were specified, the system would understand the label, but would automatically convert it to TS SA, because the SA subcompartment has a minimum classification of TS, and requires compartment A. However, the fact that the label had to be converted means it was not in canonical form. A canonical clearance has a short classification name and long word names. A human-readable label is in canonical form if it can be converted into internal format and coverted back to human-readable format with the two human-readable formats being identical. In the error message, YYY is the canonical form of the clearance represented by XXX.

In ACCREDITATION RANGE: MINIMUM PROTECT AS CLASSIFICATION “XXX” greater than classification in MINIMUM CLEARANCE.

This error occurs in the ACCREDITATION RANGE: section if the minimum protect as classification specified is greater than the classification in the minumum clearance specified, but is an otherwise valid classification.

In ACCREDITATION RANGE: MINIMUM SENSITIVITY LABEL “XXX” not in canonical form. Is YYY what is intended?

This error occurs if the sensitivity label specified after the minimum sensitivity label= keyword in the ACCREDITATION RANGE: section while understood, is not in canonical form. This additional canonicalization check ensures that no errors are made in specifying the minimum sensitivity label. For example, given the sample encodings in Appendix B, “Annotated Sample Encodings,” of the sensitivity label C SA were specified, the system would understand the label, but would automatically convert it to TS SA, because the SA subcompartment has a minimum classification of TS, and requires compartment A. However, the fact that the label had to be converted means it was not in canonical form. A canonical sensitivity label has a short classification name and long word names. A human-readable label is in canonical form if it can be converted into internal format and coverted back to human-readable format with the two human-readable formats being identical. In the error messasge, YYY is the canonical form of the sensitivity label represented by XXX.

In ACCREDITATION RANGE: MINIMUM SENSITIVITY LABEL must be dominatd by MINIMUM CLEARANCE.

This error occurs in the ACCREDITATION RANGE: section if the minimum sensitivity label specified is not dominated by the minimum clearance specified, but is an otherwise valid sensitivity label.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “COMPARTMENTS= XXX”.

This error occurs when multiple COMPARTMENTS= keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “FLAGS= XXX”.

This error occurs when multiple FLAGS keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “MAXCLASS= XXX”.

This error occurs when multiple MAXCLASS= keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “MINCLASS= XXX”.

This error occurs when multiple MINCLASS= keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “OMINCLASS= XXX”.

This error occurs when multiple OMINCLASS= keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “PREFIX”.

This error occurs when multiple PREFIX keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “PREFIX=”.

This error occurs when multiple PREFIX= keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “SNAME= XXX”.

This error occurs when multiple SNAME= keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “SUFFIX”.

This error occurs when multiple SUFFIX keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “SUFFIX= XXX”.

This error occurs when multiple SUFFIX= keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Invalid COMPARTMENTS specification “XXX”.

This error occurs whenever the text that follows a compartments= keyword in not in the proper format or specifies a bit position larger than the system-specific maximum compartment bit.

In CHANNELS WORDS, word “XXX”: Invalid FLAGS specification “XXX”.

This error occurs whenever the text that follows a flags= keyword is not in the proper format or specifies a bit position larger than 14.

In CHANNELS WORDS, word "XXX": Keyword ACCESS RELATED does not apply to CHANNELS words.

This error occurs whenever an ACCESS RELATED keyword is specified for a channel word.

In CHANNELS WORDS, word "XXX": Keyword MARKINGS does not apply to CHANNELS WORDS.

This error occurs whenever a MARKINGS= keyword is specified for a channel word.

In CHANNELS WORDS, word "XXX": MAXCLASS "XXX" is less than MINCLASS "XXX".

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In CHANNELS WORDS, word "XXX": MAXCLASS "XXX" is less than OMINCLASS "XXX".

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified OMINCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In CHANNELS WORDS, word "XXX": MINCLASS "XXX" is greater than MAXCLASS "XXX".

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In CHANNELS WORDS, word "XXX": OMINCLASS "XXX" is greater than MAXCLASS "XXX".

This error occurs when an OMINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “OMAXCLASS= XXX”.

This error occurs when multiple OMAXCLASS= keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Keyword INAME does not apply to CHANNELS words.

This error occurs whenever an INAME= keyword is specified for a channel word.

In CHANNELS WORDS, word “XXX”: OMAXCLASS “XXX” is less than MINCLASS “XXX”.

This error occurs whenever an OMAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The OMAXCLASS must always be greater than or equal to the MINCLASS.

In CHANNELS WORDS, word “XXX”: OUTPUT MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMAXCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In CHANNELS WORDS, word “XXX”: MINCLASS “XXX” is greater than OMAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified OMAXCLASS for the same word. The MINCLASS must always be less than or equal to the OMAXCLASS.

In CHANNELS WORDS, word “XXX”: OUTPUT MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMINCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In CHANNELS WORDS, word “XXX”: PREFIX “XXX” not found.

This error occurs whenever the text that follows a prefix= keyword does not represent a long or short name defined with the prefix keyword in the same WORDS: subsection.

In CHANNELS WORDS, word “XXX”: SUFFIX “XXX” not found.

This error occures whenever the text that follows a suffix= keyword does not represent a long or short name defined with the suffix keyword in the same WORDS subsection.

In CLASSIFICATION “XXX”: Invalid INITIAL COMPARTMENTS specification “XXX”.

This error occurs whenever the text that follows an initial compartments= keyword is not in the proper format or specifies a bit pisition larger than the system-specific maximum compartment bit.

In CLASSIFICATION “XXX”: Invalid INITIAL MARKINGS specification “XXX”.

This error occurs whenever the text that follows an initial markings= keyword is not in the proper format or specifies a bit position larger than the system-specific maximum marking bit.

In CLEARANCES COMBINATION CONSTRAINTS: Keyword “CHANNELS:” cannot start a continuation line.

This error occurs if a clearance combination constraint specification line ends with a \ and the next non-blank line (which is supposed to continue the specification) starts with the CHANNELS: keyword.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “PREFIX”.

This error occurs when multiple PREFIX keywords appear in the specification of a channel word.

In CHANNELS WORDS, word “XXX”: Duplicate keyword “PREFIX= XXX”.

This error occurs when multiple PREFIX= keywords appear in the specification of a channel word.

In CLEARANCES WORDS, word “XXX”: Default word for XXX has a greater minimum classification (XXX).

This error occurs when a default word has a minimum classification greater than some classification whose initial compartments and initial markings specify the word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “COMPARTMENTS= XXX”.

This error occurs when multiple COMPARTMENTS= keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “FLAGS= XXX”.

This error occurs when multiple FLAGS keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “MAXCLASS= XXX”.

This error occurs when multiple MAXCLASS= keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “MINCLASS= XXX”.

This error occurs when multiple MINCLASS= keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “OMINCLASS= XXX”.

This error occurs when multiple OMINCLASS= keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “PREFIX”.

This error occurs when multiple PREFIX keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “PREFIX= XXX”.

This error occurs when multiple PREFIX= keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “SNAME= XXX”.

This error occurs when multiple SNAME= keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “SUFFIX”.

This error occurs when multiple SUFFIX keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “SUFFIX= XXX”.

This error occurs when multiple SUFFIX= keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: Invalid COMPARTMENTS specification “XXX”.

This error occurs whenever the text that follows a compartments= keyword is not in the proper format or specifies a bit position larger than the system-specific maximum compartment bit.

In CLEARANCES WORDS, word “XXX”: Invalid FLAGS specification “XXX”.

This error occurs whenever the text that follows a flags= keyword is not in the proper format or specifies a bit position larger than 14.

In CLEARANCES WORDS, word “XXX”: Keyword ACCESS RELATED does not apply to CLEARANCES words.

This error occurs whenever an ACCESS RELATED keyword is specified for a clearance word.

In CLEARANCES WORDS, word “XXX”: Keyword MARKINGS does not apply to CLEARANCES words.

This error occurs whnever a MARKINGS= keyword is specified for a clearance word.

In CLEARANCES WORDS, word “XXX”: MAXCLASS “XXX” is less than MINCLASS “XXX”.

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In CLEARANCES WORDS, word “XXX”: MAXCLASS “XXX” is less than OMINCLASS “XXX”.

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified OMINCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In CLEARANCES WORDS, word “XXX”: MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified MAXCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In CLEARANCES WORDS, word “XXX”: MINCLASS “XXX” is greater than MAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In CLEARANCES WORDS, word “XXX”: MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified MINCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In CLEARANCES WORDS, word “XXX”: No corresponding inverse compartment found in SENSITIVITY LABELS WORDS.

This error occurs when a clearance inverse compartment word does not have a corresponding sensitivity label inverse compartment word whose compartment bits are dominated by the clearance word’s compartment bits

In CLEARANCES WORDS, word “XXX”: OMINCLASS “XXX” is greater than MAXCLASS “XXX”.

This error occurs when an OMINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In CLEARANCES WORDS, word “XXX”: Word contains default bits in combination with non-default bits.

This error occurs when a word specifying default bits also specifies other bits that are not default.

In CLEARANCES WORDS, word “XXX”: Duplicate keyword “OMAXCLASS= XXX”.

This error occurs when multiple OMAXCLASS= keywords appear in the specification of a clearance word.

In CLEARANCES WORDS, word “XXX”: OMAXCLASS “XXX” is less than MINCLASS “XXX’.

This error occurs when an OMAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The OMAXCLASS must always be greater than or equal to the MINCLASS.

In CLEARANCES WORDS, word “XXX”: OUTPUT MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMAXCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In CLEARANCES WORDS, word “XXX”: MINCLASS “XXX” is greater than OMAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified OMAXCLASS for the same word. Ths MINCLASS must always be less than or equal to the OMAXCLASS.

In CLEARANCES WORDS, word “XXX”: OUTPUT MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMINCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In CLEARANCES WORDS, word “XXX”: PREFIX “XXX” not found.

This error occurs whenever the text that follows a prefix= keyword does not represent a long or short name defined with the prefix keyword in the same WORDS: subsection.

In CLEARANCES WORDS, word “XXX”: SUFFIX “XXX” not found.

This error occurs whenever the text that follows a suffix= keyword does not represent a long or short name defined with the suffix keyword in the same WORDS: subsection.

In INFORMATION LABEL WORDS, word “XXX”: MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified MAXCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In INFORMATION LABEL WORDS, word “XXX”: MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified MINCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In INFORMATION LABELS COMBINATION CONSTRAINTS: Keyword “SENSITIVITY LABELS:” cannot start a continuation line.

This error occurs if an information label combination constraint specification line ends with a \ and the next non-blank line (which should continue the specification) starts with the SENSITIVITY LABELS: keyword.

In INFORMATION LABELS WORDS, word “XXX”: A word that requires a prefix with compartments or markings must also specify special inverse compartment or marking bits that correspond to bits in the required prefix’s compartments or markings.

This error occurs when a word specifies a prefix with special inverse bits, but the word does not specify any of the special inverse bits specified by the prefix.

In INFORMATION LABELS WORDS, word “XXX”: A word that requires a prefix with compartments or markings must specify a subset of the bits in the prefix.

This error occurs when a word specifies a prefix with special inverse bits, but the word specifies at least one bit not specified by the prefix.

In INFORMATION LABELS WORDS, word “XXX”: Default word for XXX has a greater minimum classification (XXX).

This error occurs when a default word has a minimum classification greater than some classification whose initial compartments and initial markings specify the word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “ACCESS RELATED”.

This error occurs when multiple ACCESS RELATED keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “COMPARTMENTS= XXX”.

This error occurs when multiple COMPARTMENTS= keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “FLAGS= XXX”.

This error occurs when multiple FLAGS keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “MARKINGS= XXX”.

This error occurs when multiple MARKINGS= keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “MAXCLASS= XXX”.

This error occurs when multiple MAXCLASS= keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “MINCLASS= XXX”.

This error occurs when multiple MINCLASS= keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “OMINCLASS= XXX”.

This error occurs when multiple OMINCLASS= keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “PREFIX”.

This error occurs when multiple PREFIX keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “PREFIX= XXX”.

This error occurs when multiple PREFIX= keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “SNAME= XXX”.

This error occurs when multiple SNAME= keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “SUFFIX”.

This error occurs when multiple SUFFIX keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “SUFFIX= XXX”.

This error occurs when multiple SUFFIX= keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: Invalid COMPARTMENTS specification “XXX”.

This error occurs whenever the text that follows a compartments= keyword is not in the proper format or specifies a bit position larger than the system-specific maximum compartment bit.

In INFORMATION LABELS WORDS, word “XXX”: Invalid FLAGS specification “XXX”.

This error occurs whenever the text that follows a flags= keyword is not in the proper format or specifies a bit position larger than 14.

In INFORMATION LABELS WORDS, word “XXX”: Invalid MARKINGS specification “XXX”.

This error occurs whenever the text that follows a markings= keyword is not in the proper format or specifies a bit position larger than the system-specific maximum marking bit.

In INFORMATION LABELS WORDS, word “XXX”: MAXCLASS “XXX” is less than MINCLASS “XXX”.

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In INFORMATION LABELS WORDS, word “XXX”: MAXCLASS “XXX” is less than OMINCLASS “XXX”.

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified OMINCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In INFORMATION LABELS WORDS, word “XXX”: MINCLASS “XXX” is greater than MAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In INFORMATION LABELS WORDS, word “XXX”: OMINCLASS “XXX” is greater than MAXCLASS “XXX”.

This error occurs when an OMINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In INFORMATION LABELS WORDS, word “XXX”: Word contains default bits in combination with non-default bits.

This error occurs when a word specifying default bits also specifies other bits that are not default.

In INFORMATION LABELS WORDS, word “XXX”: Duplicate keyword “OMAXCLASS= XXX”.

This error occurs when multiple OMAXCLASS=keywords appear in the specification of an information label word.

In INFORMATION LABELS WORDS, word “XXX”: OMAXCLASS “XXX” is less than MINCLASS “XXX”.

This error occurs when an OMAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The OMAXCLASS must always be greater than or equal to the MINCLASS.

In INFORMATION LABELS WORDS, word “XXX”: OUTPUT MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMAXCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In INFORMATION LABELS WORDS, word “XXX”: MINCLASS “XXX” is greater than OMAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified OMAXCLASS for the same word. The MINCLASS must always be less than or equal to the OMAXCLASS.

In INFORMATION LABELS WORDS, word “XXX”: OUTPUT MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMINCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In INFORMATION LABELS WORDS, word “XXX”: PREFIX “XXX” not found.

This error occurs whenever the text that follows a prefix= keyword does not represent a long or short name defined with the prefix keyword in the same WORDS: subsection.

In INFORMATION LABELS WORDS, word “XXX”: SUFFIX “XXX” not found.

This error occurs whenever the text that follows a suffix= keyword does not represent a long or short name defined with the suffix keyword in the same WORDS: subsection.

In NAME INFORMATION LABELS: A NAME= keyword must always be followed by an IL= keyword.

This error occurs if an IL= keyword is missing following a name= keyword in the NAME INFORMATION LABELS: section.

In NAME INFORMATION LABELS: A NAME= keyword must precede an IL= keyword.

This error occurs when an IL= keyword appears at the beginning of the NAME INFORMATION LABEL: section or when two IL= keywords appear in a row. An IL= keyword must always have one or more name= keywords preceding it.

In NAME INFORMATION LABELS: INFORMATION LABEL “XXX” not in canonical form. Is YYY what was intended?

This error occurs if an information label specified in the NAME INFORMATION LABELS: section while understood, is not in canonical form. This additional canonicalization check ensures that no errors are made in specifying the name information labels. For example, given the sample encodings in Appendix B, “Annotated Sample Encodings,” if the information label C SA were specified, the system would understand the label, but would automatically convert it to TS SA, because the SA subcompartment has a minimum classification of TS, and the canonical form of informatin labels includes the classification name. However, the fact that the label had to be converted means it was not in canonical form. A canonical information label has a short classification name and long word names. A human-readable label is in canonical form if it can be converted into internal format and coverted back to human-readable format with the two human-readable formats being identical. In the error messasge, YYY is the canonical form of the information label represented by XXX.

In NAME INFORMATION LABELS: Invalid INFORMATION LABEL “XXX”.

This error occurs if an information label specified in the NAME INFORMATION LABELS: section cannot be understood as a valid information label based on the encodings. This error would typically occur if the information label contained an invalid word or classification.

In NAME INFORMATION LABELS: NAME “XXX” not found.

This error occurs when the name following a name= keyword in the NAME INFORMATION LABEL: section cannot be found as a classification name, sname, or aname, or a word name or sname (including prefix or suffix names or snames).

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “COMPARTMENTS= XXX”.

This error occurs when multiple COMPARTMENTS= keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “FLAGS= XXX”.

This error occurs when multiple FLAGS keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “MARKINGS= XXX”.

This error occurs when multiple MARKINGS= keywords appear in the specification of an information label word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “MAXCLASS= XXX”.

This error occurs when multiple MAXCLASS= keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “MINCLASS= XXX”.

This error occurs when multiple MINCLASS= keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “OMINCLASS= XXX”.

This error occurs when multiple OMINCLASS= keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “PREFIX”.

This error occurs when multiple PREFIX keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “PREFIX= XXX”.

This error occurs when multiple PREFIX= keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “SNAME= XXX”.

This error occurs when multiple SNAME= keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “SUFFIX”.

This error occurs when multiple SUFFIX keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “SUFFIX= XXX”.

This error occurs when multiple SUFFIX= keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Invalid COMPARTMENTS specification “XXX”.

This error occurs whenever the text that follows a compartments= keyword is not in the proper format or specifies a bit position larger than the system-specific maximum compartment bit.

In PRINTER BANNERS WORDS, word “XXX”: Invalid FLAGS specification “XXX”.

This error occurs whenever the text that follows a flags= keyword is not in the proper format or specifies a bit position larger than 14.

In PRINTER BANNERS WORDS, word “XXX”: Invalid MARKINGS specification “XXX”.

This error occurs whenever the text that follows a markings= keyword is not in the proper format or specifies a bit position larger than the system-specific maximum marking bit.

In PRINTER BANNERS WORDS, word “XXX”: Keyword ACCESS RELATED does not apply to PRINTER BANNERS words.

This error occurs whenever an ACCESS RELATED keyword is specified for a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: MAXCLASS “XXX” is less than MINCLASS “XXX”.

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In PRINTER BANNERS WORDS, word “XXX”: MAXCLASS “XXX” is less than OMINCLASS “XXX”.

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified OMINCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In PRINTER BANNERS WORDS, word “XXX”: MINCLASS “XXX” is greater than MAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In PRINTER BANNERS WORDS, word “XXX”: OMINCLASS “XXX” is greater than MAXCLASS “XXX”.

This error occurs when an OMINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In PRINTER BANNERS WORDS, word “XXX”: Duplicate keyword “OMAXCLASS= XXX”.

This error occurs when multiple OMAXCLASS= keywords appear in the specification of a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: Keyword INAME does not apply to PRINTER BANNERS words.

This error occurs whenever an INAME= keyword is specified for a printer banner word.

In PRINTER BANNERS WORDS, word “XXX”: OMAXCLASS “XXX” is less than MINCLASS “XXX”.

This error occurs when an OMAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The OMAXCLASS must always be greater than or equal to the MINCLASS.

In PRINTER BANNERS WORDS, word “XXX”: OUTPUT MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMAXCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In PRINTER BANNERS WORDS, word “XXX”: MINCLASS “XXX” is greater than OMAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified OMAXCLASS for the same word. The MINCLASS must always be less than or equal to the OMAXCLASS.

In PRINTER BANNERS WORDS, word “XXX”: OUTPUT MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMINCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In PRINTER BANNERS WORDS, word "XXX": PREFIX "XXX" not found.

This error occurs whenever the text that follows a prefix= keyword does not represent a long or short name defined with the prefix keyword in the same WORDS: subsection.

In PRINTER BANNERS WORDS, word "XXX": SUFFIX "XXX" not found.

This error occurs whenever the text that follows a suffix= keyword does not represent a long or short name defined with the suffix keyword in the same WORDS: subsection.

In SENSITIVITY LABEL WORDS, word "XXX": MINIMUM CLASSIFICATION "XXX" not found.

This error occurs whenever a specified MINCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In SENSITIVITY LABELS COMBINATION CONSTRAINTS: Keyword "CLEARANCES:" cannot start a continuation line.

This error occurs if a sensitivity label combination constraint specification line ends with a \ and the next non-blank line (which should continue the specification) starts with the CLEARANCES: keyword.

In SENSITIVITY LABELS WORDS, word "XXX": A word that requires a prefix with compartments or markings must also specify special inverse compartment or marking bits that correspond to bits in the required prefix's compartments or markings.

This error occurs when a word specifies a prefix with special inverse bits, but the word does not specify any of the special inverse bits specified by the prefix.

In SENSITIVITY LABELS WORDS, word "XXX": Default word for XXX has a greater minimum classification (XXX).

This error occurs when a default word has a minimum classification greater than some classification whose initial compartments and initial markings specify the word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “COMPARTMENTS= XXX”.

This error occurs when multiple COMPARTMENTS= keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “FLAGS= XXX”.

This error occurs when multiple FLAGS keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “MAXCLASS= XXX”.

This error occurs when multiple MAXCLASS= keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “MINCLASS= XXX”.

This error occurs when multiple MINCLASS= keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “PREFIX”.

This error occurs when multiple PREFIX keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “PREFIX= XXX”.

This error occurs when multiple PREFIX= keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “SNAME= XXX”.

This error occurs when multiple SNAME= keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “SUFFIX”.

This error occurs when multiple SUFFIX keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “SUFFIX= XXX”.

This error occurs when multiple SUFFIX= keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Invalid COMPARTMENTS specification “XXX”.

This error occurs whenever the text that follows a compartments= keyword is not in the proper format or specifies a bit position larger than the system-specific maximum compartment bit.

In SENSITIVITY LABELS WORDS, word “XXX”: Invalid FLAGS specificatin “XXX”.

This error occurs whenever the text that follows a flags= keyword is not in the proper format or specifies a bit position larger than 14.

In SENSITIVITY LABELS WORDS, word “XXX”: Keyword ACCESS RELATED does not apply to SENSITIVITY LABELS words.

This error occurs whenever an ACCESS RELATED keyword is specified for a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: Keyword MARKINGS does not apply to SENSITIVITY LABELS words.

This error occurs whenever a markings= keyword is specified for a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: MAXCLASS “XXX” is less than MINCLASS “XXX”.

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In SENSITIVITY LABELS WORDS, word “XXX”: MAXCLASS “XXX” is less than OMINCLASS “XXX”.

This error occurs when a MAXCLASS= keyword is specified with a classification whose value is less than a previously-specified OMINCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In SENSITIVITY LABELS WORDS, word “XXX”: MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified MAXCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In SENSITIVITY LABELS WORDS, word “XXX”: MINCLASS “XXX” is greater than MAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the MINCLASS.

In SENSITIVITY LABELS WORDS, word “XXX”: No corresponding inverse compartment found in INFORMATION LABELS WORDS.

This error occurs when asensitivity label inverse compartment word does not have a corresponding information label inverse compartment word whose compartment bits are dominated by the sensitivity label word’s compartment bits, and whose markings contain no normal bits.

In SENSITIVITY LABELS WORDS, word “XXX”: OMINCLASS “XXX” is greater than MAXCLASS “XXX”.

This error occurs when an OMINCLASS= keyword is specified with a classification whose value is greater than a previously-specified MAXCLASS for the same word. The MAXCLASS must always be greater than or equal to the OMINCLASS.

In SENSITIVITY LABELS WORDS, word “XXX”: Word contains default bits in combination with non-default bits.

This error occurs when a word specifying default bits also specifies other bits that are not default.

In SENSITIVITY LABELS WORDS, word “XXX”: Duplicate keyword “OMAXCLASS= XXX”.

This error occurs when multiple OMAXCLASS= keywords appear in the specification of a sensitivity label word.

In SENSITIVITY LABELS WORDS, word “XXX”: OMAXCLASS “XXX” is less than MINCLASS “XXX”.

This error occurs when an OMAXCLASS= keyword is specified with a classification whose value is less than a previously-specified MINCLASS for the same word. The OMAXCLASS must always be greater than or equal to the MINCLASS.

In SENSITIVITY LABELS WORDS, word “XXX”: OUTPUT MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMAXCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In SENSITIVITY LABELS WORDS, word “XXX”: MINCLASS “XXX” is greater than OMAXCLASS “XXX”.

This error occurs when a MINCLASS= keyword is specified with a classification whose value is greater than a previously-specified OMAXCLASS for the same word. The MINCLASS must always be less than or equal to the OMAXCLASS.

In SENSITIVITY LABELS WORDS, word “XXX”: OUTPUT MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified OMINCLASS does not appear as a classification long, short, or alternate name in the CLASSIFICATIONS: section.

In SENSITIVITY LABELS WORDS, word “XXX”: PREFIX “XXX” not found.

This error occurs whenever the text that follows a prefix= keyword does not represent a long or short name defined with the prefix keyword in the same WORDS: subsection.

In SENSITIVITY LABELS WORDS, word “XXX”: SUFFIX “XXX” not found.

This error occurs whenever the text that follows a suffix= keyword does not represent a long or short name defined with the suffix keyword in the same WORDS: subsection.

Invalid characters in CLASSIFICATION value specification “XXX”.

This error occurs if the text following a value= keyword does not represent an integer value.

Invalid COMPARTMENTS specification “XXX”.

This error occurs whenever the text that follows a compartments= keyword is not in the proper format, or specifies a bit position larger than the system-specific maximum compartment bit.

Invalid FLAGS specification “XXX”.

This error occurs whenever the text that follows a flags= keyword is not in the proper format, or specifies a bit position larger than 14.

Invalid INITIAL COMPARTMENTS specification “XXX”.

This error occurs whenever the text that follows an initial compartments= keyword is not in the proper format, or specifies a bit position larger than the system-specific maximum compartment bit.

Invalid INITIAL MARKINGS specification “XXX”.

This error occurs whenever the text that follows an initial markings= keyword is not in the proper format, or specifies a bit position larger than the system-specific maximum marking bit.

Invalid MARKINGS specification “XXX”.

This error occurs whenever the text that follows a markings= keyword is not in the proper format, or specifies a bit position larger than the system-specific maximum marking bit.

Invalid MINIMUM CLEARANCE “XXX”.

This error occurs if the clearance specified after the minimum clearance= keyword in the ACCREDITATION RANGE: section cannot be understood as a valid clearance based on the encodings. This error would typically occur if the clearance contained an invalid word or classification.

Invalid MINIMUM PROTECT AS CLASSIFICATION “XXX”.

This error occurs if the classification specified after the minimum protect as classification= keyword in the ACCREDITATION RANGE: section does not appear as a classification long or short name in the CLASSIFICATIONS: section.

Invalid MINIMUM SENSITIVITY LABEL "XXX".

This error occurs if the sensitivity label specified after the minimum sensitivity label= keyword in the ACCREDITATION RANGE: section cannot be understood as a valid sensitivity label based on the encodings. This error would typically occur if the sensitivity label contained an invalid word or classification.

Invalid sensitivity label "XXX".

This error occurs if a sensitivity label specified after an ALL COMPARTMENT COMBINATIONS VALID EXCEPT:; keyword or an ONLY VALID COMPARTMENT COMBINATIONS:;; keyword in the ACCREDITATION RANGE: section cannot be understood as a valid sensitivity label based on the encodings. This error would typically occur if the sensitivity label contained an invalid word or classification.

Keyword "CHANNELS:" cannot start a continuation line.

This error occurs if a clearance combination constraint specification line ends with a \, and the next non-blank line (which is supposed to continue the specification) starts with the CHANNELS: keyword.

Keyword "CLEARANCES:" cannot start a continuation line.

This error occurs if a sensitivity label combination constraint specification line ends with a \, and the next non-blank line (which is supposed to continue the specification) starts with the CLEARANCES: keyword.

Keyword "SENSITIVITY LABELS:" cannot start a continuation line.

This error occurs if an information label combination constraint specification line ends with a \, and the next non-blank line (which is supposed to continue the specification) starts with the SENSITIVITY LABELS: keyword.

Keyword ACCESS RELATED does not apply to CHANNELS words.

This error occurs whenever an ACCESS RELATED keyword is specified for a channel word.

Keyword ACCESS RELATED does not apply to CLEARANCES words.

This error occurs whenever an ACCESS RELATED keyword is specified for a clearance word.

Keyword ACCESS RELATED does not apply to PRINTER BANNERS words.

This error occurs whenever an ACCESS RELATED keyword is specified for a printer banner word.

Keyword ACCESS RELATED does not apply to SENSITIVITY LABELS words.

This error occurs whenever an ACCESS RELATED keyword is specified for a sensitivity label word.

Keyword MARKINGS does not apply to CHANNELS words.

This error occurs whenever a MARKINGS= keyword is specified for a channel word.

Keyword MARKINGS does not apply to CLEARANCES words.

This error occurs whenever a MARKINGS= keyword is specified for a clearance word.

Keyword MARKINGS does not apply to SENSITIVITY LABELS words.

This error occurs whenever a MARKINGS= keyword is specified for a sensitivity label word.

MAXIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified maxclass does not appear as a classification long or short name in the CLASSIFICATIONS: section.

Maximum sensitivity label not well formed.

This error occurs whenever the maximum sensitivity label is not well formed. The maximum sensitivity label is taken to be the highest classification along with all of the compartment bits referenced in all words and initial compartments being 1.

MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified minclass does not appear as a classification long or short name in the CLASSIFICATIONS: section.

MINIMUM CLEARANCE “XXX” not in canonical form. Is YYY what was intended?

This error occurs if the clearance specified after the minimum clearance= keyword in the ACCREDITATION RANGE: section, while able to be understood, is not in canonical form. This additional canonicalization check is made to ensure that no errors are made in specifying the minimum clearance. For example, given the sample encodings in Appendix B, “Annotated Sample Encodings,” if the clearance C SA were specified, the system would understand the label, but would automatically convert it to TS SA, because the SA subcompartment has a minimum classification of TS. However, the fact that the label had to be converted means it was not in canonical form. A canonical clearance has a short classification name and long word names. A human-readable label is in canonical form if it can be converted into internal format and converted back to human-readable format, with the two human-readable formats being identical. In the error message, YYY is the canonical form of the clearance represented by XXX.

MINIMUM PROTECT AS CLASSIFICATION “XXX” greater than classification in MINIMUM CLEARANCE.

This error occurs in the ACCREDITATION RANGE: section if the minimum protect as classification specified is greater than the classification in the minimum clearance specified, but is an otherwise valid classification.

MINIMUM SENSITIVITY LABEL “XXX” not in canonical form. Is YYY what is intended?

This error occurs if the sensitivity label specified after the minimum sensitivity label= keyword in the ACCREDITATION RANGE: section, while able to be understood, is not in canonical form. This additional canonicalization check is made to ensure that no errors are made in specifying the minimum sensitivity label. For example, given the sample encodings inAppendix B, “Annotated Sample Encodings,” if the sensitivity label C SA were specified, the system would understand the label, but would automatically convert it to TS SA, because the SA subcompartment has a minimum classification of TS. However, the fact that the label had to be converted means it was not in canonical form. A canonical sensitivity label has a short classification name and long word names. A human-readable label is in canonical form if it can be converted into internal format and converted back to human-readable format, with the two human-readable formats being identical. In the error message, YYY is the canonical form of the sensitivity label represented by XXX.

MINIMUM SENSITIVITY LABEL must be dominated by minimum clearance.

This error occurs in the ACCREDITATION RANGE: section if the minimum sensitivity label specified is not dominated by the minimum clearance specified, but is an otherwise valid sensitivity label.

Minimum information label not well formed. The initial compartments or initial markings for “XXX” are specified incorrectly.

This error occurs when the minimum information label is not well formed. The minimum information label is taken to be the lowest classification, and those initial compartments and markings bits for the lowest classification that are not inverse bits (i.e., represent default words).

Missing ! or & in CLEARANCES COMBINATION CONSTRAINTS “XXX”.

This error occurs if a clearance combination constraint specification line does not contain a ! or a & character. All combination constraints must have one of these two characters to be meaningful.

Missing ! or & in INFORMATION LABELS COMBINATION CONSTRAINTS: “XXX”.

This error occurs if an information label combination constraint specification line does not contain a ! or a & character. All combination constraints must have one of these two characters to be meaningful.

Missing ! or & in SENSITIVITY LABELS COMBINATION CONSTRAINTS “XXX”.

This error occurs if a sensitivity label combination constraint specification line does not contain a ! or a & character. All combination constraints must have one of these two characters to be meaningful.

Missing or unrecognized word in CLEARANCES COMBINATION CONSTRAINTS “XXX”.

This error occurs if a clearance combination constraint specification line is missing a required word (e.g., after a ! or a |) or contains a word that cannot be found in the clearance WORDS: subsection.

Missing or unrecognized word in INFORMATION LABELS COMBINATION CONSTRAINTS “XXX”.

This error occurs if an information label combination constraint specification line is missing a required word (e.g., after a ! or a |) or contains a word that cannot be found in the information label WORDS: subsection.

Missing or unrecognized word in SENSITIVITY LABELS COMBINATION CONSTRAINTS “XXX”.

This error occurs if a sensitivity label combination constraint specification line is missing a required word (e.g., after a ! or a |) or contains a word that cannot be found in the sensitivity label WORDS: subsection.

Missing |, !, or & in CLEARANCES COMBINATION CONSTRAINTS “XXX”.

This error occurs if a clearance combination constraint specification line contains two words not separated by a |, !, or & character.

Missing |, !, or & in INFORMATION LABELS COMBINATION CONSTRAINTS “XXX”.

This error occurs if an information label combination constraint specification line contains two words not separated by a |, !, or & character.

Missing |, !, or & in SENSITIVITY LABELS COMBINATION CONSTRAINTS “XXX”.

This error occurs if a sensitivity label combination constraint specification line contains two words not separated by a |, !, or & character.

Multiple &s and/or !s in CLEARANCES COMBINATION CONSTRAINTS “XXX”.

This error occurs if a clearance combination constraint specification line contains more than one ! or & character. Only one of these characters is allowed in each constraint specification.

Multiple &s and/or !s in INFORMATION LABELS COMBINATION CONSTRAINTS “XXX”.

This error occurs if an information label combination constraint specification line contains more than one ! or & character. Only one of these characters is allowed in each constraint specification.

Multiple &s and/or !s in SENSITIVITY LABELS COMBINATION CONSTRAINTS “XXX”.

This error occurs if a sensitivity label combination constraint specification line contains more than one ! or & character. Only one of these characters is allowed in each constraint specification.

No CHANNELS WORDS non-prefix/suffix entries.

This error occurs when a channel WORDS: subsection contains at least one prefix or suffix word, but no other words that are not prefixes or suffixes. It is valid to have no words at all, but it makes no sense to specify a prefix or a suffix and no other words.

No CLEARANCES WORDS non-prefix/suffix entries.

This error occurs when a clearance WORDS: subsection contains at least one prefix or suffix word, but no other words that are not prefixes or suffixes. It is valid to have no words at all, but it makes no sense to specify a prefix or a suffix and no other words.

No INFORMATION LABELS WORDS non-prefix/suffix entries.

This error occurs when an information label WORDS: subsection contains at least one prefix or suffix word, but no other words that are not prefixes or suffixes. It is valid to have no words at all, but it makes no sense to specify a prefix or a suffix and no other words.

No PRINTER BANNERS WORDS non-prefix/suffix entries.

This error occurs when a printer banner WORDS: subsection contains at least one prefix or suffix word, but no other words that are not prefixes or suffixes. It is valid to have no words at all, but it makes no sense to specify a prefixB or a suffix and no other words.

No SENSITIVITY LABELS WORDS non-prefix/suffix entries.

This error occurs when a sensitivity label WORDS: subsection contains at least one prefix or suffix word, but no other words that are not prefixes or suffixes. It is valid to have no words at all, but it makes no sense to specify a prefix or a suffix and no other words.

No sensitivity labels allowed after ALL COMPARTMENT COMBINATIONS VALID.

This error occurs if a classification= keyword or the minimum clearance= keyword does not follow an ALL COMPARTMENT COMBINATIONS VALID keyword in the ACCREDITATION RANGE: section.

OUTPUT MINIMUM CLASSIFICATION “XXX” not found.

This error occurs whenever a specified ominclass does not appear as a classification long or short name in the CLASSIFICATIONS: section.

PREFIX “XXX” not found.

This error occurs whenever the text that follows a prefix= keyword does not represent a long or short name defined with the prefix keyword in the same WORDS: subsection.

Sensitivity label “XXX” not in canonical form. Is YYY what was intended?

This error occurs if a sensitivity label specified after an ALL COMPARTMENT COMBINATIONS VALID EXCEPT:; keyword or an ONLY VALID COMPARTMENT COMBINATIONS:;; keyword in the ACCREDITATION RANGE: section, while able to be understood, is not in canonical form. This additional canonicalization check is made to ensure that no errors are made in specifying the accreditation range. For example, given the sample encodings in Appendix B, “Annotated Sample Encodings,” if the sensitivity label C SA were specified, the system would understand the label, but would automatically convert it to TS SA, because the SA subcompartment has a minimum classification of TS. However, the fact that the label had to be converted means it was not in canonical form. A human-readable label is in canonical form if it can be converted into internal format and converted back to human-readable format, with the two human-readable formats being identical. In the error message, YYY is the canonical form of the sensitivity label represented by XXX.

SUFFIX “XXX” not found.

This error occurs whenever the text that follows a suffix= keyword does not represent a long or short name defined with the suffix keyword in the same WORDS: subsection.

The compartment bits specified for channels are not dominated by those specified for information labels, sensitivity labels, and clearances.

This error occurs whenever the words in the CHANNELS: section specify some compartment bits not specified by words in the INFORMATION LABELS:, SENSITIVITY LABELS:, and CLEARANCES: sections.

The compartment bits specified for clearances do not equal those specified for information labels and sensitivity labels.

This error occurs whenever the words in the CLEARANCES: section do not, in the aggregate, specify the same compartments bits as the words in the INFORMATION LABELS: and SENSITIVITY LABELS: sections specify in the aggregate.

The compartment bits specified for printer banners are not dominated by those specified for information labels, sensitivity labels, and clearances.

This error occurs whenever the words in the PRINTER BANNERS: section specify some compartment bits not specified by words in the INFORMATION LABELS:, SENSITIVITY LABELS:, and CLEARANCES: sections.

The compartment bits specified for sensitivity labels do not equal those specified for information labels.

This error occurs whenever the words in the SENSITIVITY LABELS: section do not, in the aggregate, specify the same compartments bits as the words in the INFORMATION LABELS: section specify in the aggregate.

The first keyword after CLASSIFICATIONS must be NAME.

This error occurs if a sname=, value=, initial compartments=, or initial markings= keyword appears immediately after the CLASSIFICATIONS: keyword.

The marking bits specified for printer banners are not dominated by those specified for information labels.

This error occurs whenever the words in the PRINTER BANNERS: section specify some marking bits not specified by words in the INFORMATION LABELS: section.

Unrecognized CLEARANCES REQUIRED COMBINATION “XXX”.

This error occurs if a clearance required combination specification does not contain exactly two words (including required prefixes or suffixes) that can be found in the clearance WORDS: subsection. This error would also probably occur if the COMBINATION CONSTRAINTS: keyword was misspelled.

Unrecognized INFORMATION LABELS REQUIRED COMBINATION “XXX”.

This error occurs if an information label required combination specification does not contain exactly two words (including required prefixes or suffixes) that can be found in the information label WORDS: subsection. This error would also probably occur if the COMBINATION CONSTRAINTS: keyword were misspelled.

Unrecognized SENSITIVITY LABELS REQUIRED COMBINATION “XXX”.

This error occurs if a sensitivity label required combination specification does not contain exactly two words (including required prefixes or suffixes) that can be found in the sensitivity label WORDS: subsection. This error would also probably occur if the COMBINATION CONSTRAINTS: keyword were misspelled.

Annotated Sample Encodings B

This appendix contains a sample encodings file, along with annotations that describe the purpose of most specifications within the file. The sample file, which is similar though not identical to the sample encodings in [DDS-2600-6215-91], is designed to illustrate a number of realistic examples. All annotations in the file appear within boxes.

> The **VERSION** specification is useful for identifying different versions of the encodings. It is stored by the system when the encodings are loaded. It can be used to facilitate interoperability among multiple CMW systems.

VERSION= DISTRIBUTED DEMO VERSION

CLASSIFICATIONS:

```
*
* Comments can be placed in the encodings file any place a keyword can start.
* Comments begin with a * and continue to the end of the line.
*
```

The classification specifications below define the common classifications. The values specified represent the proper hierarchy among the classifications and leave room for later expansion below **UNCLASSIFIED**, between **UNCLASSIFIED** and **CONFIDENTIAL**, and above **TOP SECRET**. There are no initial compartments or markings specifications for **UNCLASSIFIED** because all compartment and marking bits are intended to be 0 in **UNCLASSIFIED** labels. However, the initial compartment and marking specifications for the remaining (classified) classifications all specify those bits that are used inversely in the information label, sensitivity label, and clearances encodings below, plus extra bits reserved for future use as inverse bits. Compartment bits 4-5 are used for the release compartments **REL CNTRY1** and **REL CNTRY2**. These bits being 0 in an **UNCLASSIFIED** label means that the label indicates releasability to both countries. In other labels, the bits being 1 as specified means that unless the words **REL CNTRY1** or **REL CNTRY2** are explicitly added to the label, the data is not releasable to those countries. Marking bit 11 is the inverse bit used in the **REL CNTRY3** release marking. Marking bit 17 is the inverse bit used in the inverse word **charlie**. Marking bit 12 is the inverse bit used in the inverse codeword **bravo4**. These words will be discussed in more detail below. Compartments and marking bits 100-127 are reserved for future expansion as inverse bits.

name= UNCLASSIFIED;sname= U; value= 1;

name= CONFIDENTIAL; sname= C; value= 4; initial compartments= 4-5 100-127;

initial markings= 11 12 17 100-127

name= SECRET; sname= S; value= 5; initial compartments= 4-5 100-127;

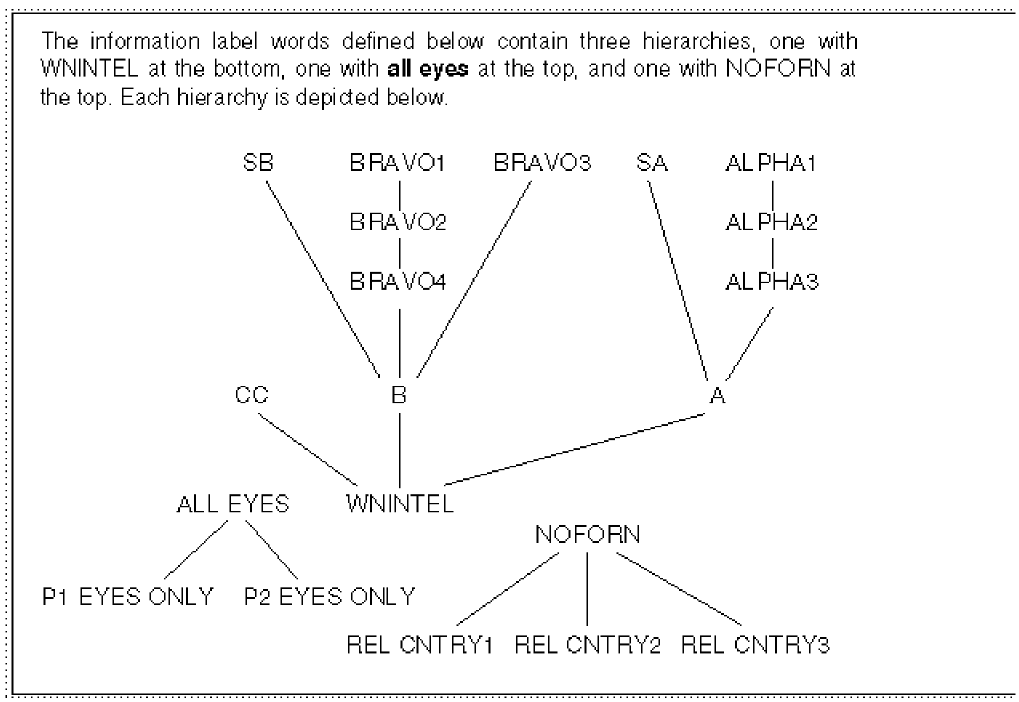
initial markings= 11 12 17 100-127

name= TOP SECRET; sname= TS; value= 6; initial compartments= 4-5 100-127;

initial markings= 11 12 17 100-127

INFORMATION LABELS:

> The information label words defined below contain three hierarchies, one with **WNINTEL** at the bottom, one with all eyes at the top, and one with **NOFORN** at the top. Each hierarchy is depicted below.



WORDS:

> Note that all of the prefixes and suffixes appear at the beginning of the **WORDS** subsection. Note also that the case used in specifying names does not matter.

name= REL; prefix;

name= LIMDIS; sname= LD;suffix;

name= ORCON; sname= OC; prefix;

name= eyes only; sname= eo; suffix;

After the prefixes and suffixes are specified, those words that represent compartments, subcompartments, and codewords are specified. Note that the words are in order of decreasing importance. **CC**, **B**, and **A** are main compartments, also commonly called *channels*. **SB** and **SA** are subcompartments of **B** and **A**, respectively. **bravo1** through **bravo4** are **B** codewords, and **alpha1** through **alpha3** are **A** codewords. Note that all of the compartments, subcompartments, and codewords specify marking bit 7. This bit, when in a label with no compartment bits on, specifies the marking **WNINTEL** (see below). Since it is invalid to have **WNINTEL** in a label if a compartment, subcompartment, or codeword is present, putting the **WNINTEL** bit in each of these words creates a hierarchy whereby **WNINTEL** is hierarchically below all compartments, subcompartments, and codewords. In effect, all compartments, subcompartments, and codewords “mean” **WNINTEL**, but the word **WNINTEL** is shown only for non-compartment/subcompartment/codeword **WNINTEL** data.

name= CC; minclass= TS; compartments= 6; markings= 7;

name= SB; minclass= TS; compartments= 1 3; markings= 7;

Subcompartment **SB** specifies compartment bits 1 and 3. Bit 3 is the bit for subcompartment **SB**, whereas bit 1 is the bit for its main compartment, **B**. This is specified because, by convention for information labels, specifying a subcompartment should automatically protect the information as being in the main compartment (channel).

name= bravo1; sname= b1; minclass= TS; compartments= 1; markings= 3-4 7 12;

name= bravo2; sname= b2; minclass= S; compartments= 1; markings= 3 7 12;

The use of marking bits 3 and 4 in the above two words specifies a hierarchy with **bravo1** above **bravo2**. If two information labels, each with one of the words, are combined, the result will contain only the higher word in the hierarchy—**bravo1**. Marking bit 12 is specified in **bravo2** to assure that **bravo2** is hierarchically above **bravo4** (see below). Marking bit 12 must therefore also be present in **bravo1** to assure that **bravo1** is hierarchically above **bravo2**.

name= bravo3; sname= b3; minclass= S; compartments= 1; markings= 5 7;

bravo3 is a codeword independent of **bravo1, bravo2,** and **bravo4**.

name= bravo4; sname= b4; minclass= S;

maxclass= S; compartments= 1; markings= 3 7 ~12;

bravo4 is a compartment **B** codeword which has some inverse qualities because bit 12 is off. It acts like an inverse word in that it persists through the combination of two information labels only if it is present in both labels. However, because not all of its compartment and marking bits are 0, it does not appear in **UNCLASSIFIED** labels, and therefore does not require an **ominclass.** It has a maximum classification of **SECRET**. Also, note that it is in a hierarchy with **bravo2**. Thus, if **bravo4** data is combined with any non-**bravo4** data (which includes all non-**SECRET** data), the result is automatically **bravo2**, because bit 12 (which is one of the initial markings) will turn on.

name= B; minclass= C; compartments= 1; markings= 7;

B represents non-codeword compartment **B** data. If none of the marking bits defined above for **bravo1** through **bravo4** (bits 3, 4, 5, and 12) are present in a label with compartment bit 1, the word **B** will be used to mark the data.

name= SA; minclass= TS; compartments= 0 2; markings= 7;

Subcompartment **SA** specifies compartment bits 0 and 2. Bit 2 is the bit for subcompartment **SA**, whereas bit 0 is the bit for its main compartment, **A**. This is specified because, by convention for information labels, specifying a subcompartment should automatically protect the information as being in the main compartment (channel).

name= alpha1; sname= a1; minclass= TS; compartments= 0; markings= 0-2 7;

name= alpha2; sname= a2; minclass= S; compartments= 0; markings= 0-1 7;

name= alpha3; sname= a3; minclass= S; compartments= 0; markings= 0 7;

The use of marking bits 0, 1, and 2 in the above three words specifies a hierarchy with **alpha1** above **alpha2** above **alpha3**.

name= A; minclass= C; compartments= 0; markings= 7;

A represents non-codeword compartment **A** data. If none of the marking bits defined above for **alpha1** through **alpha3** (bits 0, 1, and 2) are present in a label with compartment bit 0, the word **A** will be used to mark the data.

After the compartments, subcompartments, and codewords are specified, those words that represent markings are specified, in order of decreasing importance. Note that some of the words below do contain compartment bit references (**NOFORN**, **REL CNTRY1**, and **REL CNTRY2**). These were placed below because **NOFORN** and release markings—by convention—appear towards the end of the label.

name= project x; sname= px; minclass= C; markings= 14;

suffix= LIMDIS; access related;

The flags= keyword to the left serves a purpose only if the system has assigned some particular meaning to flag bit 3. It is included here only as an example of how flags are specified.

flags= 3;

name= project y; sname= py; minclass= C; markings= 6;

suffix= LIMDIS; access related;

The two words above both require the suffix **LIMDIS**. They represent projects whose data should only be shown to people with need-to-know for the project. There is another common usage of **LIMDIS** whereby no project name is specified. Such a usage would have **LIMDIS** as a base word, not a suffix, and would assign a unique marking bit for **LIMDIS**.

name= charlie; sname= ch; ominclass= c;

minclass= s; maxclass= S; markings= ~17;

charlie is included as an example of an extremely complicated word specification, to show some of the advanced specification features. **charlie** is an inverse marking, which is present when marking bit 17 (one of the marking bits with an initial value of **1**) is 0. Because of its **minclass** and **maxclass** specifications, it can appear only with the classification **SECRET**. As is the case with all inverse markings, **charlie** includes an **ominclass** specification, which prevents **charlie** from appearing in labels below **CONFIDENTIAL**. However, since its **minclass** is **SECRET**, why can't the **ominclass** be **SECRET** or omitted entirely? It can't be omitted because to do so would cause **charlie** to be displayed with **UNCLASSIFIED** labels (because it is an inverse marking). It *could* be **SECRET** however. With the **ominclass SECRET**, **charlie** could not be added to a **CONFIDENTIAL** label. In other words, entering “+**charlie**” to modify a **CONFIDENTIAL** label would fail. With **ominclass CONFIDENTIAL** however, entering “+**charlie**” to modify a **CONFIDENTIAL** label would force the classification to **SECRET** and add the marking **charlie**. The final thing to note about **charlie** is that it requires the codeword **alpha2** to be present (see **REQUIRED COMBINATIONS** below).

name= org x; sname= ox; minclass= C; markings= 9;

prefix= ORCON; access related;

name= org y; sname= oy; minclass= C; markings= 15;

prefix= ORCON; access related;

The two words above both require the prefix **ORCON**. They represent an extension of the typical usage of **ORCON**. The purpose of the extension is to indicate via the base word name the originator of the **ORCON** data. Thus **ORCON org x** indicates **ORCON** with **org x** as the originator, and **ORCON org x/org y** indicates data that is a combination of **ORCON org x** and **ORCON org y** data. To specify the more typical **ORCON** marking, **ORCON** would be a base word without a prefix or suffix, and would use a single marking bit.

name= D/E; minclass= C; markings= 16;

access related;

The word **D/E** is included in these encodings as an example of a word that contains a **/**. Even though **/** is used as the separator of multiple words that require the same prefix or suffix, the **/** character can be included in word names themselves. Care should be taken in any such usage of **/** to avoid confusion.

name= all eyes; access related; markings= 8 10;

The above word is a composite of the two words that follow.

name= p1; markings= 8;

suffix= eyes only; access related;

name= p2; markings= 10;

suffix= eyes only; access related;

> The above two words both require the suffix **eyes only**. They serve as an example of the fact that blanks can be included in word names, even in suffix names. These words represent an extension of the more typical encoding of **eyes only**, in that they allow a specification through the base word name of who can view the data. To specify the more typical **eyes only** marking, **eyes only** would be a base word without a prefix or suffix, and would use a single marking bit.

name= WNINTEL; sname= WN; minclass= C; markings= 7;

access related;

> Note the relationship between the **WNINTEL** marking above and the compartment, subcompartment, and codewords at the top of the information label words, all of which include marking bit 7 to form a hierarchy with **WNINTEL** at the bottom.

name= WARNING; minclass= C; markings= 7;

> Because the above word specifies the same compartments and markings as the word before it, it simply adds a third input-only name to **WNINTEL**.

The four words below comprise the release markings and their related marking **NOFORN**. In these encodings, **NOFORN** is encoded such that it cannot appear in the same label with a release marking. There are alternative encodings whereby **NOFORN** is totally independent of the release markings. In this example, **REL CNTRY1** and **REL CNTRY2** are actually release compartments, whereas **REL CNTRY3** is just a release marking. Such encodings might be used if citizens of **CNTRY1** and **CNTRY2** were direct users of this or a connected system whose access to data was mandatorily controlled through release compartments, and citizens of **CNTRY3** were not users, but could receive hardcopy system output of marked **REL CNTRY3**. The encoding of the **NOFORN** word is such that it is hierarchically above all of the release compartments and markings. Marking bit 13 was specifically specified as **1** in **NOFORN** and as **0** in the release compartments and markings to ensure this hierarchy. Because they are inverse words, **REL CNTRY1**, **REL CNTRY2**, and **REL CNTRY3** all have an **ominclass** of **CONFIDENTIAL**. This **ominclass** specification prevents these words from appearing in human-readable labels below **CONFIDENTIAL**. Therefore, even though the bit representations of these three release compartments/markings indicate that they should be present with **UNCLASSIFIED**, by convention they are not shown in **UNCLASSIFIED** labels. A useful way to think about the bit assignments involved in these release compartments/markings is as follows. Compartment bit 4 is the (inverse) bit for **REL CNTRY1**. Compartment bit 4 being **0** means that the data is releasable to **CNTRY1**. Compartment bit 5 is the (inverse) bit for **REL CNTRY2**. Compartment bit 5 being **0** means that the data is releasable to **CNTRY2**. Marking bit 11 is the (inverse) bit for **REL CNTRY3**. Marking bit 11 being **0** means that the data is releasable to **CNTRY3**. Finally, marking bit 13 is the **NOFORN** bit. Marking bit 13 being **1** means that the data is NOFORN. If the data is neither **NOFORN** nor releasable to any of the countries, compartment bits 4 and 5 will be **1**, marking bit 11 will be **1**, and marking bit 13 will be **0**.

name= NOFORN; sname= NF; minclass= C; compartments= 4-5; markings= 11 13;

access related;

name= CNTRY1; sname= c1; ominclass= C; compartments= ~4; markings= ~13;

prefix= REL

name= CNTRY2; sname= C2; ominclass= C; compartments= ~5; markings= ~13;

prefix= REL;

name= CNTRY3; sname= c3; ominclass= C; markings= ~11 ~13;

prefix= REL;

> The following word acts as an alias for the following combination of the above words: **CC SB bravo1 bravo3 SA alpha1 project X/project Y LIMDIS ORCON org x/org Y D/E all eyes NOFORN**. The alias has associated all of the compartment and marking bits of the aliased words, and no others. It also has a **minclass** equal to the highest **minclass** of any of the aliased words. Because it follows these words in the encodings, it can never appear in an output label; it can be used only as a shorthand on input for entering or adding to a label. It is intended to represent the “system high” set of information label words.

name= SYSHI; minclass= TS; compartments= 0-6; markings= 0-16;

> The **REQUIRED COMBINATIONS** below specify two constraints about the above information label words. The first specification requires that **NOFORN** be present in a label whenever subcompartment **SB** is present. The second specification requires that the codeword **alpha2** be present in a label whenever the marking **charlie** is present.

REQUIRED COMBINATIONS:

SB NF

charlie alpha2

> The **COMBINATION CONSTRAINTS** below specify three constraints about the above information label words. The first specification requires that codeword **bravo4** must stand alone in a label (along with the classification SECRET as forced by the specification above for **bravo4**). The second specification requires that the marking **charlie** can be combined *only* with the codeword **alpha2**. Note that this specification, when combined with the second required combination above, requires that the marking **charlie** , if present in a label, must appear along with **alpha2** and *only* **alpha2** and the classification **SECRET** (as forced by the specification above for **charlie**). The third specification requires that if data is marked releasable to **CNTRY3**, it cannot also be releasable to **CNTRY1** or **CNTRY2**. Note that there is no restriction on marking data releasable to **CNTRY1** and **CNTRY2**.

COMBINATION CONSTRAINTS:

bravo4 &

charlie & alpha2

> The line to the left is continued onto the next line by ending the line with a \.
> This is done as an example of the line continuation feature that might be required on long combination constraints.

REL CNTRY3 ! REL CNTRY1 | \

REL CNTRY2

SENSITIVITY LABELS:

WORDS:

> The **PREFIX** keyword to the left is shown in upper case as an example of the case insensitivity of the encodings. Note that the prefix comes at the beginning of the words.

name= REL; PREFIX;

The sensitivity label compartments below are ordered in terms of increasing importance, with the exception of the release compartments, which are at the end by convention. Most of the compartments require the specification of a single compartment bit. However, **SB** and the release compartments are a special case. Since subcompartment **SB** must appear with **NOFORN**, and since **NOFORN** cannot appear with release compartments or markings (see the encodings above), **SB** cannot appear in a sensitivity label with release compartments. This constraint is enforced below by creating a hierarchy using compartment bits with **SB** at the top of the hierarchy above **REL CNTRY1** and **REL CNTRY2**. Compartment bit 3 is the bit that means **SB**. The compartments for **SB** include bits 4 and 5 to force them to 1 when **SB** is specified. Since bits 4 and 5 are the inverse bits for the release compartments, specifying **SB** ensures that no release compartments are present. The **~3** specification in the release compartments is redundant, but serves to emphasize the hierarchy present. With this hierarchy specified, it is possible to add **SB** to a sensitivity label that contains a release compartment, thereby automatically removing the release compartment. As an alternative to the specification below, it would have been possible to enforce the fact that **SB** cannot be combined with release compartments via a combination constraint of **SB ! REL CNTRY1 | REL CNTRY2**. However, such an encoding forms no hierarchy, such that trying to add **SB** to a sensitivity label that contains a release compartment would be considered an error. Because they are inverse words, **REL CNTRY1** and **REL CNTRY2** have an **ominclass** of **CONFIDENTIAL**. This **ominclass** specification prevents these words from appearing in human-readable labels below **CONFIDENTIAL**. Therefore, even though the bit representations of these two release compartments indicate that they should be present with **UNCLASSIFIED**, by convention they are not shown in **UNCLASSIFIED** labels.

name= A; minclass= C; compartments= 0;

name= B; minclass= C; compartments= 1;

name= SA; minclass= TS; compartments= 2;

name= SB; minclass= TS; compartments= 3-5;

name= CC; minclass= TS; compartments= 6;

name= CNTRY1; sname= c1; ominclass= C; compartments= ~3 ~4;

prefix= REL;

name= CNTRY2; sname= c2; ominclass= C; compartments= ~3 ~5;

prefix= REL;

Because of the system invariant that the compartment bits in sensitivity labels must always dominate the compartment bits in associated information labels, the presence of one of the above two words in a sensitivity label *forces* the same word to appear in an associated information labels.

The **REQUIRED COMBINATIONS** below specify that if subcompartment **SB** is present in a sensitivity label, compartment **B** must also be present. Similarly, if subcompartment **SA** is present in a sensitivity label, compartment **A** must also be present. Note how differently this requirement is met in this sensitivity label encoding compared to how it was met above in the information label encoding. In the sensitivity label—by convention—both compartments and subcompartments can appear, which is accomplished by this encoding. In the information label, the presence of a subcompartment automatically forces the appropriate main compartment bit to be present, but does not include the main compartment name in the human-readable representation of the label—again by convention.

REQUIRED COMBINATIONS:

SB B

SA A

There are no combination constraints for sensitivity label words, so the subsection below has no constraints specified. Note that the subsection must be present even if it is empty.

COMBINATION CONSTRAINTS:

The **CLEARANCES** section below is similar to the **SENSITIVITY LABELS** section above, but with two differences. First, the prefix used for the release compartments is different. Whereas it makes sense to mark data REL COUNTRY, when the same concept is applied to clearances, and therefore related to users, it makes more sense to refer to the nationality of the user, rather than having REL COUNTRY in the user's clearance. Therefore, this section uses the prefix **NATIONALITY:** before the country words. Second, there is a combination constraint specified. Since the release compartments **NATIONALITY: CNTRY1** and **NATIONALITY: CNTRY2** in a clearance mean that the user is a citizen of the country, the constraint specifies that a clearance cannot specify that a user is a citizen of more than one country. Note that no such constraint is needed for sensitivity labels, because the meaning of the release compartments in a sensitivity label is that the data is releasable to citizens of the country, and data can be releasable to more than one country. Because they are inverse words, **NATIONALITY: CNTRY1** and **NATIONALITY: CNTRY2** have an **ominclass** of **CONFIDENTIAL**. This **ominclass** specification prevents these words from appearing in human-readable labels below **CONFIDENTIAL**. Therefore, even though the bit representations of these two release compartments indicate that they should be present with **UNCLASSIFIED**, by convention they are not shown in **UNCLASSIFIED** labels.

CLEARANCES:

WORDS:

name= NATIONALITY:; sname= N:; prefix;

name= A; minclass= C; compartments= 0;

name= B; minclass= C; compartments= 1;

name= SA; minclass= TS; compartments= 2;

name= SB; minclass= TS; compartments= 3-5;

name= CC; minclass= TS; compartments= 6;

name= CNTRY1; sname= c1; ominclass= C; compartments= ~3 ~4;

prefix= NATIONALITY:;

name= CNTRY2; sname= c2; ominclass= C; compartments= ~3 ~5;

prefix= NATIONALITY:;

REQUIRED COMBINATIONS:

SB B

SA A

COMBINATION CONSTRAINTS:

NATIONALITY: c1 ! NATIONALITY: c2

The **CHANNELS** section specifies the **HANDLE VIA...** caveats associated with the main compartments (channels) specified above, for use by the system in producing printer banner pages. If the sensitivity label indicates only one channel present, the caveat should be of the form **HANDLE VIA (CHANNEL NAME) CHANNELS ONLY**. If the sensitivity label indicates multiple channels present, the caveat should be of the form **HANDLE VIA (CHANNEL NAME)/(CHANNEL NAME)/... CHANNELS JOINTLY**. The encodings could specify a unique word for each channel and each combination of channels, but such an encoding would be extremely long with a large number of encodings. Rather, the encodings below takes full advantage of the fact that words can require both a prefix and a suffix to shorten the specifications. To fully understand the encodings below, you must know how the system uses the channel words in producing the caveat string. The words are scanned in the order specified, with all words whose compartment bits are present in the sensitivity label placed into the caveat string in the order in which they are encountered. Once a compartment bit has been matched in the sensitivity label, it is "forgotten" as the rest of the words are scanned. Note that none of the words below contains an **sname**, because only long names are used for producing the channel caveat string.

CHANNELS:

WORDS

:

The encodings below define a single prefix, **HANDLE VIA**, which is the prefix for *every* word in the encodings. Two suffixes are defined: **CHANNELS ONLY** for the case when only one channel is present, and **CHANNELS JOINTLY** for the case when more than one channel is present. Each main word below requires the prefix and one of the suffixes.

name= CHANNELS JOINTLY; suffix;

name= CHANNELS ONLY; suffix;

name= HANDLE VIA; prefix;

The first three main words cover the case where only a single channel is present. The compartment bit specifications of each will match a sensitivity label only if a single channel is present. Note that all non-channel bits are ignored. For example, the compartments specification for the word (**CH A**) is **0 ~1 ~6**, which will match only a sensitivity label with bit 0 (for channel **A**) on and bits 1 and 6 (for channels **B** and **CC**) off. These first three entries all require the suffix **CHANNELS ONLY**. Once a compartment bit is matched by one of these words, it will be "forgotten" as the remaining words are scanned, so that none of the final three words will be placed in the caveat string if one of the first three are. Note that the order of these first three words does not matter, because at most one of them will ever match a sensitivity label.

name= (CH A); prefix=HANDLE VIA; compartments= 0 ~1 ~6;

suffix= CHANNELS ONLY;

name= (CH B); prefix=HANDLE VIA; compartments= ~0 1 ~6;

suffix= CHANNELS ONLY;

name= (CH C); prefix=HANDLE VIA; compartments= ~0 ~1 6;

suffix= CHANNELS ONLY;

The last three main words cover the case where multiple channels are present. Any of these words that match the sensitivity label will be placed in the caveat string, preceded by **HANDLE VIA**, separated by /, and followed by **CHANNELS JOINTLY**. Note that these words are in order of decreasing sensitivity, and must follow the single channel encodings above.

name= (CH C); prefix=HANDLE VIA; compartments= 6;

suffix= CHANNELS JOINTLY;

name= (CH B); prefix=HANDLE VIA; compartments= 1;

suffix= CHANNELS JOINTLY;

name= (CH A); prefix=HANDLE VIA; compartments= 0;

suffix= CHANNELS JOINTLY;

The **PRINTER BANNERS** section specifies the nonchannel-related caveats associated with compartments and markings, for use by the system in producing printer banner pages. Note that none of the words below contains an **sname**, because only long names are used for producing the printer banner caveat string. Note also that these words are in order of decreasing sensitivity .

PRINTER BANNERS:

WORDS:

name= ORCON; prefix;

name= (FULL SB NAME); compartments= 3

name= (FULL SA NAME); compartments= 2

These first two words specify caveats associated with the subcompartments defined above. Note that all main compartments (channels) are ignored by the encodings in this section. Each word specifies the name to be placed in the printer banner caveat string if the specified compartments (in this case subcompartments) match the sensitivity label. Note that the compartments specifications could also have included the associated main compartment bits, because they are forced to be present along with the subcompartment bits (i.e., **compartments= 3** could have been **compartments= 1 3**).

name= org x; prefix= ORCON; markings= 9;

name= org y; prefix= ORCON; markings= 15;

These two words specify caveats associated with certain markings defined above. Each word specifies the name to be placed in the printer banner caveat string if the specified markings match the information label.

The **ACCREDITATION RANGE** section specifies the system and user accreditation ranges and related constants. The user accreditation range is the set of sensitivity labels at which normal system users can operate. In the general case, not all possible sensitivity labels containing the compartments defined for the system are in the user accreditation range. The encodings allow for the specification of the user accreditation range in the most compact manner possible, rather than having to list every possible valid sensitivity label. The valid sensitivity labels for each classification are specified separately. Since no specification for the classification **UNCLASSIFIED** appears below, the sensitivity label **UNCLASSIFIED** is not in the user accreditation range.

ACCREDITATION RANGE:

In this example, the most compact way to specify the valid **CONFIDENTIAL** sensitivity labels is to list only those sensitivity labels that are *invalid*, presumably because the list of invalid labels is shorter or more meaningful.

classification= c; all compartment combinations valid except:;

c

c a

c b

In this example, the most compact way to specify the valid **SECRET** sensitivity labels is to state only those sensitivity labels that are *valid*, presumably because the list of valid labels is shorter or more meaningful.

classification= s; only valid compartment combinations:; ;

s a b

In this example, all **TOP SECRET** sensitivity labels are valid.

classification= ts; all compartment combinations valid;

Below the minimum clearance that can be associated with a user is specified. The system will not allow a clearance that is below the minimum to be specified. Note that the clearance specified below represents **TOP SECRET** with all compartment bits 0. Note also that this clearance is *not* a legal clearance according to the encodings above, but does represent a useful minimum, being the only clearance immediately below **TS NATIONALITY: CNTRY1** and **TS NATIONALITY: CNTRY2**.

minimum clearance= ts NATIONALITY: CNTRY1/CNTRY2;

Below the minimum sensitivity label for the system is specified. The system will not allow a sensitivity label that is below the minimum to be specified. Note that the sensitivity label specified below represents **CONFIDENTIAL** with all compartment bits 0. There should be no sensitivity labels in the user accreditation range specification below the minimum sensitivity label, but the minimum sensitivity label does *not* have to be in the user accreditation range, though it should be the greatest lower bound of all sensitivity labels in the user accreditation range. In this case it is in fact the lowest sensitivity label in the user accreditation range.

minimum sensitivity label= c REL CNTRY1/CNTRY2;

Below the minimum classification that can appear on the top and bottom of printer banner pages is specified. This classification is also the minimum that will appear in the printer banner warning statement that specifies how the data must be protected unless it is manually reviewed and downgraded.

minimum protect as classification= ts;

CMW Labeling Software C1.0 Release Notes, 6/8/93

C

Release C1.0 is a modification of Release 2.1.1 that incorporates minor bug fixes, portability improvements, and a new functional capability. This appendix is an addendum to the original document.

The new functional capability is supported for SPECIAL_INVERSE words. SPECIAL_INVERSE words are distinctly different from the inverse words supported by Release 2.1.1 (herein called regular inverse words). Regular inverse words use inverse bits specified in the initial compartments or initial markings associated with one or more classifications. These regular inverse words, typically specified with an ominclass specification, do not appear in human-readable labels below the specified ominclass. In contrast, SPECIAL_INVERSE words use inverse bits specified by a prefix word, and do not appear in human-readable labels unless one or more of the SPECIAL_INVERSE words associated with the same prefix are present in the label.

SPECIAL_INVERSE words can be used to implement the ORiginator CONtrolled (ORCON) handling caveat with organizations to which the ORCON data can be released specified in the label. For example, given that three organizations use a particular system (ORG1, ORG2, and ORG3), the encodings to handle ORCON for these three organizations might look as follows. Only the SENSITIVITY LABELS words are shown in this example.

```
SENSITIVITY LABELS:

WORDS:

name=ORCON RELEASABLE TO; sname=OR; compartments=1-4; prefix;

name=ORCON; minclass=C; compartments=1-4;
name=ORG1; minclass=C; compartments=~1 4; prefix=OR;
name=ORG2; minclass=C; compartments=~2 4; prefix=OR;
name=ORG3; minclass=C; compartments=~3 4; prefix=OR;
```

In this example, ORG1, ORG2, and ORG3 are SPECIAL_INVERSE words, each of which requires the prefix ORCON RELEASABLE TO. This prefix specifies compartments bits 1-4. Bit one is for ORG1, bit 2 for ORG2, bit 3 for ORG3, and bit 4, has a meaning of ORCON. If only ORCON RELEASABLE TO ORG1 is present in a label, then bit 1 would be off, and bits 2-4 would be on. If only ORCON RELEASABLE TO ORG2 is present in a label, then bit 2 would be off, and bits 1, 3, and 4 would be on. If only ORCON RELEASABLE TO ORG3 is present in a label, then bit 3 would be off, and bits 1, 2, and 4 would be on. If ORCON RELEASABLE TO ORG1 / ORG2 is present in a label, then bits 1 and 2 would be off and bits 3 and 4 would be on, etc. The word ORCON, which dominates the three other words, is not an inverse word. If it appears in a label, the data so labeled is not releasable to any of the three organizations. Note that a label that does not contain any of the above words has bits 1-3 off, and is therefore releasable to all organizations, and has bit 43 off, and is therefore not ORCON data. Thus, with the same words as above for information labels, data with an information label of SECRET ORCON RELEASABLE TO ORG1, when combined with data with an information label of TOP SECRET, would become TOP SECRET ORCON RELEASABLE TO ORG1. SPECIAL_INVERSE words can be specified using markings bits also.

In addition to the changes mentioned above, minor improvements to comments in the code were made. The comment changes are not described below.

Files Changed

1. l_init.c

2. std_labels.c
3. CMW Label Setting Demo.c
4. std_labels.h

Changes to Subroutine Interfaces

The interface to internal subroutine turnoff_word was changed in two ways. First, the flag argument can now be passed as RECURSING, for usage when turnoff_word calls itself. Second, the flag argument can be passed as FORCE_OFF_BY_TURNON_WORD, to indicate that turnon_word is calling turnoff_word to force off a word.

Changes to Encodings File

Prefix words can now accept compartments= and markings= specifications, with meanings as described above. In prior releases, prefix words could accept these specifications, but they were ignored.

Changes to std_labels.h

1. On line 36 (of Release 2.1.1), strings.h was changed to string.h for ANSI compatibility. This change was made in a portion of std_labels.h that vendors are encouraged to change to adapt to their own libraries.
2. Added declarations of l_t3_compartments, l_t3_markings, l_t4_compartments, l_t4_markings, l_t5_compartments, and l_t5_markings.
3. The l_w_type flags SPECIAL_INVERSE and SPECIAL_COMPARTMENTS_INVERSE were defined.

Changes to Subroutines

1. CMW Label Setting Demo.c was changed in Macintosh-specific ways. The changes were in no way related to the usage of the labeling subroutines.
2. In std_labels.c and l_init.c all uncast calls to calloc were cast to (char *).
3. In l_init.c (line 2862 of Release 2.1.1), sl_buffer was changed to cl_buffer.

4. In l_init.c (line 1456 of Release 2.1.1), added a cast of (unsigned int) to a subroutine argument. This cast would be required on some compilers/systems.
5. In l_init.c, added subroutine check_special_inverse. This subroutine is called by do_words to error check and set l_w_type flags for SPECIAL_INVERSE words.
6. In l_init.c, added the allocation of a new set of temporary variables to hold compartments/markings: l_t5_compartments, and l_t5_markings.
7. In l_init.c, changed subroutine check_inverse_words to check new l_w_type SPECIAL_COMPARTMENTS_INVERSE as appropriate.
8. In std_labels.c, in the subroutine word_forced_on, fixed check of comparison or normal compartment to the minimum compartments.
9. In std_labels.c, in the subroutine word_forced_on, added check to cause word_forced_on to ignore SPECIAL_INVERSE words.
10. In std_labels.c, in the subroutine l_parse, added support for SPECIAL_INVERSE words.
11. In std_labels.c, in the subroutine turnoff_word, added algorithms to deal with the special case of SPECIAL_INVERSE words. In particular, turnoff_word was made recursive.
12. In std_labels.c, in the subroutine turnon_word, added algorithms to deal with the special case of SPECIAL_INVERSE words.
13. In std_labels.c, in the subroutine l_valid, added support for SPECIAL_INVERSE words.

Changes to Encodings Specification Error Messages

The following encodings specification error messages have been added:

- In INFORMATION LABELS WORDS, word “XXX”:

  A word that requires a prefix with compartments or markings must specify a subset of the bits in the prefix.

- In SENSITIVITY LABELS WORDS, word “XXX”:

A word that requires a prefix with compartments or marking must specify a subset of the bits in the prefix.

- In CLEARANCE WORDS, word “XXX”:

  A word that requires a prefix with compartments or markings must specify a subset of the bits in the prefix.

- In INFORMATION LABELS WORDS, word “XXX”:

  A word that requires a prefix with compartments or markings must also specify inverse compartment or marking bits that correspond to bits in the required prefix’s compartments or markings.

- In SENSITIVITY LABELS WORDS, word “XXX”:

  A word that requires a prefix with compartments or markings must also specify inverse compartment or marking bits that correspond to bits in the required prefix’s compartments or markings.

- In CLEARANCE WORDS, word “XXX”:

  A word that requires a prefix with compartments or markings must also specify inverse compartment or marking bits that correspond to bits in the required prefix’s compartments or markings.

Glossary

Access

A specific type of interaction between a process and an object that results in the flow of information from one to the other.

Accreditation Range

A set of sensitivity labels. See System Accreditation Range and User Accreditation Range.

Adjudication

When two information labels are combined, the determination of the information label that represents the proper combination of the two.

Classification

A designation applied to data indicating the sensitivity of the data with respect to national security. The designation, as described by executive order 12356, is one of the following, given in increasing order of sensitivity: UNCLASSIFIED, CONFIDENTIAL, SECRET, TOP SECRET.

Clearance

A designation applied to a person, indicating the sensitivity of data to which the person is allowed access when the person has an established need to know for the data. The designation consists of one of the classification levels with the possible addition of compartments, such as A or B. "Clearance" is also called security level of a user (or person) in some contexts.

Compartment

A designation applied to a type of sensitive information, indicating the special handling procedures to be used for the information and the general class of people who may have access to the information. As used in this document, “compartment” has the same meaning as the word “category” in the National Computer Security Center's *Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD. As used in this document, “compartment” refers to what the intelligence community calls compartments, subcompartments, SAPs, or SAPIs.

Canonical Form

A standard format for a label. A human-readable sensitivity label is in canonical form if it consists of a short classification name followed by any zero or more words, with any words present appearing in the same order as they appear in the SENSITIVITY LABELS: section of the encodings. A human-readable clearance is in canonical form if it consists of a short classification name followed by any zero or more words, with any words present appearing in the same order as they appear in the CLEARANCES: section of the encodings. A human-readable information label is in canonical form if it consists of a long classification name followed by any zero or more words, with any words present appearing in the same order as they appear in the INFORMATION LABELS: section of the encodings.

Codeword

As used throughout this document, a word on which mandatory access control *is not* directly based (i.e., users are not specifically “cleared” or “briefed into” the word), but which implies a compartment on which mandatory access control *is* directly based.

Default bit. An initial compartment or marking bit that is not inverse. Default bits are associated with default words.

Default word

A word associated with *only* default bits. A default word appears in all labels containing the classification with which the word’s default bits are associated, but can be prevented from appearing in the human-readable representation of a label with an output minimum classification.

Dominate

Security level SL1 dominates security level SL2 if the classification in SL1 is greater than or equal to the classification in SL2 and all the compartments in SL2 are also contained in SL1. A sensitivity label is said to dominate an information label if the security level in the sensitivity label dominates the security level in the information label. More generally, any bit string S1 dominates bit string S2 if all of the bits on in S1 are also on in S2.

IL. Information Label

Information Label. A piece of information that accurately represents the sensitivity of the *data* in a subject or object. An information label consists of an information level and other required security markings (e.g., codewords and handling caveats, control and release markings), to be used for data labeling purposes. The term *information label* is used when referring to both the information level and markings, and the term *information level* is used when referring to only the level portion of the label (not including the markings).

Information Level

The security level in an information label. An information level represents the actual classification and compartments of the *data* in a subject or object with which the level is associated. Information levels are used for data labeling, not for mandatory access control.

Initial bit

A compartment or marking bit specified in the initial compartments or initial markings associated with some classification.

Inverse bit

A compartment or marking bit in the internal representation of a label whose 0 value is associated with the presence of a word in a human-readable label *and* that is specified as 1 in the initial compartments or the initial markings for one or more classifications. An inverse bit is 1 in a label that does not contain any of the inverse words associated with the bit, and is therefore 1 in a label that contains no words.

Inverse word

A component of a human-readable label, other than a classification, whose internal representation contains at least one inverse bit. Adding an inverse word to a label either decreases or changes the sensitivity of the label, but never increases the sensitivity (i.e., changes at least one bit from 1 to 0).

ISSO

Information System Security Officer.

Label

A piece of information that represents either a clearance, a sensitivity level (see Sensitivity Label), or an information level and markings (see Information Label).

Level

See Security Level.

Mandatory Access Control

Control of access to an object by a process, host, or person on the basis of the sensitivity label of the object and the sensitivity label of the process attempting access to the object.

Markings

Information, other than security level, that must be associated with data within a computer system and with human-readable output. Markings include codewords and handling caveats, control and release markings. Markings are a portion of an information label.

Maximum Sensitivity Label

A well-formed sensitivity label that dominates all sensitivity labels in the system accreditation range.

Minimum Clearance

The lowest clearance of any user on the system.

Minimum Sensitivity Label

A well-formed sensitivity label that is dominated by all sensitivity labels in the system accreditation range.

Minimum Information Label

A well-formed information label that is dominated by all other information labels possible on the system. Empty objects are created with the minimum information label.

Normal word

A component of a human-readable label, other than a classification, whose internal representation includes no inverse bits. Adding a normal word to a label increases the sensitivity of the label.

Normal user

Any system user designated by the ISSO as a normal user. Normal users typically include at least those users that are not operators, administrators, or ISSOs. Normal users are constrained to create only sensitivity labels that appear in the accreditation range of the system.

Object

A passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bytes, words, fields, processors, video displays, keyboards, clocks, printers, network nodes, etc.

Process

An independent unit of activity operating on behalf of a specific system user or on behalf of the system itself. A process can be thought of as a program in execution, but different users running the same program have different processes running the programs. Similarly, the same user running multiple different programs can have different processes running the programs, and the same user running a program more than once has different processes running the program.

SAP

Special Access Program.

SAPI

Special Access Program for Intelligence. Those special access programs involving intelligence activities that fall within the statutory authority and responsibility of the Director of Central Intelligence. Within these provisions, only those programs that require, as a condition of access, the signing of a nondisclosure statement are considered to be SAPIs. SAPIs are thereby given the status of SCI compartments and subcompartments in terms of the minimum required security levels necessary for their protection.

SCI

Sensitive Compartmented Information. All information and materials requiring special Community controls indicating restricted handling within present and future Community intelligence collection programs and their end products. These special Community controls are formal systems of restricted access established to protect the sensitive aspects of intelligence sources and methods and analytical procedures of foreign intelligence programs. The term does not include Restricted Data as defined in Section II, Public Law 585, Atomic Energy Act of 1954, as amended.

Security Level

A hierarchical classification and a set of nonhierarchical compartments, and any SAPs and/or SAPIs.

Sensitivity Label

A piece of information that represents the sensitivity level of a subject or an object and that describes the sensitivity (e.g., classification) of the data in the subject or object. Sensitivity labels/levels are used as the basis for mandatory access control decisions. Sensitivity labels/levels, because they are used for mandatory access control, must represent the sensitivity of both the subject or object and the data in the subject or object. Therefore, sensitivity labels/levels sometimes overrepresent the sensitivity of the *data* itself. The information label associated with subjects and objects is generally a more accurate representation of the sensitivity of the *data* in the subject or object. In the intelligence community, sensitivity labels typically represent the classification, compartments, subcompartments, SAPs and/or SAPIs associated with the subject or object.

Sensitivity Level

The security level in a sensitivity label. See Sensitivity Label.

SL

Sensitivity Label.

Special inverse bit

A compartment or marking bit in the internal representation of a label whose 0 value is associated with the presence of a word in a human-readable label and that is specified as 1 in the prefix associated with the word. A special inverse bit is 0 in a label that does not contain any of the special inverse words associated with the prefix that specifies the bit and is therefore 0 in a label that contains no words.

System Accreditation Range

A set of sensitivity labels, denoted by a well-formed minimum sensitivity label and a well-formed maximum sensitivity label, that represents those sensitivity labels that can be processed by the system as a whole. All sensitivity labels and clearances processed by the system must be well formed, must dominate the minimum sensitivity label, and must be dominated by the maximum sensitivity label.

User Accreditation Range

The subset of the system accreditation range that normal (non-authorized) users of the system can set (i.e., those sensitivity labels at which users can create subjects or objects, or to which users can change existing sensitivity labels). The user accreditation range applies only to sensitivity labels to be associated with subjects and objects and used for mandatory access control.

Visible Word

A word is said to be visible in a label if the presence of the word in the label does not cause the label to violate the dominance constraint on the label. The classification and compartments portion of each information label must be dominated by the associated sensitivity label, which in turn must be dominated by the associated user's clearance. If adding a word to an information label increases the sensitivity of the label such that the associated sensitivity label no longer dominates the information label, then the word is not visible in that information label.

Well Formed Label

A label that satisfies the set of well-formedness criteria specified in the encodings. These criteria include 1) the initial compartments and markings associated with each classification; 2) the minimum classification, output minimum classification, and maximum classification associated with each word; 3) the hierarchies defined by the bit patterns chosen for each word; 4) the required combinations of words; and 5) the combination constraints that apply to the words.

Word

A component of a human-readable label other than a classification.

Index