Manual M-11, Information Resources Management (Veterans Health Administration)

Chapter 10, Programming Standards and Conventions (SAC)
(Paragraphs 10.01 through 10.05; Appendix 10A and Appendix 10B)

This document includes:

Title page and p. ii for M-11, dated **January 17, 1995**
Contents page for M-11, dated **January 17, 1995**
Rescissions page for M-11, dated **January 17, 1995**

Contents page for Chapter 10, dated **January 17, 1995**
Text for Chapter 10, dated **January 17, 1995**
Text for Appendix 10A and Appendix 10B, dated **January 17, 1995**

Transmittal sheets located at the end of the document:

Sheets dated **January 17, 1995**

DEPARTMENT OF
VETERANS AFFAIRS

INFORMATION RESOURCES MANAGEMENT

M-11
January 17, 1995

Veterans Health Administration
Washington, DC 20420

Department of Veterans Affairs
Veterans Health Administration
Washington, DC 20420

January 17, 1995

The Department of Veterans Affairs, Veterans Health Administration Manual M-11, "Medical Information Resources Management," is published for the information and compliance of all concerned.

Kenneth W. Kizer, M.D., M.P.H.
Under Secretary for Health

Distribution: **RPC: 1461 is assigned**

Printing Date: 1/95

CONTENTS

CHAPTERS

RESCISSIONS

The following material is rescinded:

Circulars/Directives

10-85-93
10-85-112
10-85-116
10-86-147
10-87-19
10-87-119
10-87-122
10-87-123

CONTENTS

CHAPTER 10. PROGRAMMING STANDARDS AND CONVENTIONS (SAC)

PARAGRAPH **PAGE**

APPENDICES

CHAPTER 10. PROGRAMMING STANDARDS AND CONVENTIONS (SAC)

10.01 PURPOSE

The purpose of this chapter is to:

a. Provide a cornerstone for all national software developed and distributed within Veterans Health Administration (VHA);

b. Provide Programming SAC mandate functional soundness and technical correctness of Decentralized Hospital Computer Program (DHCP) and all other programs written in the MUMPS programming language; and

c. Provide a base of utilities and techniques for uniformity.

NOTE: *MUMPS is the acronym for Massachusetts General Hospital Utility Multi-Programming System and is a registered trademark of Massachusetts General Hospital.*

10.02 POLICY

a. All DHCP package software developed within VHA will conform to Programming SAC. In areas where specific VHA standards have yet to be adopted, software development will conform to American National Standards Institute (ANSI) standards.

b. The VHA Programming SAC have been established as a key aspect of Quality Control under Software Management (see Ch. 9).

c. The VHA Standards And Conventions Committee (SACC) defines Programming SAC and serves as the source of appeal whenever issues of conformity with Programming SAC arise.

d. SACC members and other technical experts within VHA will participate in the development of Programming SAC in collaboration with public and private subject experts and with national and international standards organizations.

NOTE: *Membership rules for the SACC are outlined in Chapter 1.*

10.03 DEFINITIONS

a. **Conventions.** Conventions are those programming rules which serve to promote consistency and clarity across all national packages. Elements are defined as conventions when they are determined to be of broad benefit.

b. **PROGRAMMER Mail Group.** The PROGRAMMER mail group is established on FORUM (the national electronic mail system) for discussion of technical issues. This mail group allows self-enrollment and is used to disseminate information to the VHA development community.

c. **Standards and/or Requirements.** Standards and/or requirements are programming rules which all DHCP national packages must observe.

10.04 RESPONSIBILITIES

a. The Director, Information Integration Service, Medical Information Resources Management Office (MIRMO), is responsible for Information Resource Management (IRM) development within VHA and will promulgate all Programming SAC.

b. The Information Systems Center (ISC) Directors Council is responsible for the review and recommendation of SACC proposals.

10.05 PROCEDURES

a. **Updates to Programming SAC**

(1) Proposed changes to Programming SAC shall first be published for comments via the PROGRAMMER mail group on FORUM.

(2) The SACC will forward to the PROGRAMMER mail group on FORUM the proposed Programming SAC at least 30 days prior to voting on it.

(a) After voting to accept the proposed Programming SAC, the SACC will forward it to the ISC Directors Council for recommendation.

(b) Then it will be sent to the Director, Information Integration Service, MIRMO, for approval.

(c) After approval the proposed Programming SAC will be available on FORUM as an informational message.

(3) The new Programming SAC becomes effective on the date of approval by the Director, Information Integration Service, MIRMO. After approval, the new Programming SAC will be made available on FORUM.

b. **Conformance to new Programming SAC**

(1) For a period of 6 months following the effective date of the new Programming SAC, packages submitted for verification may adhere to either the new or old Programming SAC.

(2) Packages verified under the old Programming SAC must be resubmitted for verification within 18 months to retain standing as nationally-supported software.

c. **Requests for exemptions from Programming SAC**

(1) If a developer identifies any need for exemptions from Programming SAC for a given package, a request must be made to the SACC either via written notification or the SACC mail group on FORUM.

(2) After review of the need and impact of the proposed exemption, the Chairperson of the SACC will call for a vote on the request. A two-thirds decision by the SACC shall be required to grant an exemption from the Programming SAC.

(3) Notification of the decision of the SACC will be made to the developer requesting the exemption and will also be sent to the Directors, ISC.

d. **Publication of current Programming SAC**

(1) In order to promote rapid dissemination, the current version of VHA Programming SAC will be posted on FORUM for all users to access. All developers should recognize the need for periodic review of Programming SAC on FORUM. The National Center for Documentation (NCD) will maintain a hard copy of the current version of Programming SAC (see App. 1OA).

(2) Programming SAC will be published, at a minimum, once a year by the NCD and will be distributed to ISCs and to Chiefs, IRM Service.

PROGRAMMING STANDARDS AND CONVENTIONS (SAC)

1. The current version was adopted on March 19, 1991.

2. All Decentralized Hospital Computer Program (DHCP) software will, at a minimum, meet the following standards and comply with the spirit of the conventions and extensions.

a. **MUMPS Standards**

NOTE: *MUMPS is the acronym for Massachusetts General Hospital Utility Multi-Programming System and is a registered trademark of Massachusetts General Hospital.*

(1) **MUMPS.** The 1990 ANSI X11/MUMPS Standard will be adhered to unless modified by this document except for Kernel code relating to specific MUMPS implementations.

(2) **Commands and Functions**

(a) The following may only be used by Kernel or through Kernel-supplied utilities:

1. Commands:	BREAK, JOB, VIEW, OPEN, CLOSE, HALT, USE (with parameters), and Z*
2. Functions:	$VIEW, $Z*
3. System Variables:	$Z*, $I

(b) Exclusive KILL or argumentless KILL commands shall not be used except by Kernel Logic (XUS), TaskMan (%ZTM*), Error Trap (%XTER*) and stand-alone VA FileMan startup (DI).

(c) Rules applying to the NEW command are:

1. The argumentless form of the NEW command may not be used.

2. The exclusive form of the NEW command may not be used.

3. Every NEW command must have an explicit, corresponding QUIT command.

(d) Read Commands

1. All user-input READs must have timeouts.

a. If the duration of the timeout is not specified by the variable DTIME and the duration exceeds 300 seconds, each occurrence must be documented in the technical manual.

b. If the argument of the read is in any way evaluated by the application, upon timeout the program must return to MenuMan with no more than one intervening READ which also times out or halt through H XUS.

2. User input shall be terminated by a carriage return character.

3. All user-input READs which are in any way evaluated by the application must be escapable by entering an up-arrow ("^") which takes an action other than a reread.

4. All prompts requesting user input must provide additional help when the user enters a question-mark ("?").

a. Any unrecognized or inappropriate response must be handled properly i.e., at a minimum, in a manner similar to the way VA FileMan handles responses (see VA FileMan User's Manual).

b. Responses to READs that are in no way evaluated by the application are excluded from this standard.

5. In scrolling mode, defaults must be so indicated with a double slash ("//") or "replace" indicating that "replace/with" editing is allowed. The null response (i.e. typing only the RETURN key) shall select the default.

(3) **Name Requirements.** Routine, global, security key, option, template, bulletin names, functions, and help frames must be consistent with the assigned namespace, except as approved by the DBIC.

(4) **Local Variables**

(a) No routines are allowed to SET, KILL, or NEW the variable DUZ or any DUZ-array element except Kernel security routines (XUS*, XUP), TaskMan (%ZTM*), and VA FileMan (DICRW).

(b) A package may declare namespaced, local variables as package-wide. The variables and all array elements must be defined in the technical documentation.

(c) Documentation on how to create and kill package-wide variables on an option that is removed from its normal menu path must be included in the technical manual.

(d) The following must not exist after exiting an option:

1. All documented output variables and scratch global nodes created by a called utility.

2. All local variables and scratch global nodes directly created by the application.

(e) The variables U, DT, and DTIME, have no array elements and shall be initially defined by Kernel and will not be KILLed or NEWed. DTIME, if changed, must be restored to its original value before exiting the option and documented in the package technical manual.

(f) Except as documented in the various Kernel technical/programmer manuals, the variables IO* and any of their array elements may only be created, changed, or KILLed by Kernel routines or utilities.

(g) Only Kernel may SET or KILL % variables. Exceptions to this are the single character variable, %, and the variables set for and returned by documented Kernel utilities.

(h) All utilities must leave the local symbol and lock tables unchanged upon exit with the exception of documented input and output variables, utility-namespaced variables and those variables composed of a single upper case letter followed optionally by one numeric, all of which may be either created, changed,

or killed. These callable utilities must be documented in the technical manual with a descriptive list of all input and resulting output variables.

(5) **Routines**

(a) Maximum routine size, as determined by %ZOSF("SIZE"), is 5000 characters. Without routine mapping, the combination of a routine and symbol table must run in a 16K partition using DSM-11 and a 24K partition using InterSystems M/11+.

(b) Routine lines must not exceed 245 characters in length and must only contain the ASCII character values 32-126. Lower case will only be used for text, not in local variable names, commands, functions, program names, or global names.

(c) Vendor-specific subroutines will not be used, except by the Kernel.

(d) Line Format of Exported Routines:

1. The first line of a routine must be in the following format: routine name<ls>;domain(site)/programmer-brief description<space>;date/(time is optional).

2. The second line must be in the following format: (TAG-optional)<ls>;;version number;package name;**pm,...pn**; version date where pm,...pn are the applied patch numbers separated by commas, this ";" piece is null if there are no patches.

a. The version date must be the same on all of the package-namespaced routines including inits.

b. The package version number must be the same on all of the package-namespaced routines including inits.

3. Lines referenced by $TEXT must be in the following format: (TAG-optional)<ls>;;text or MUMPS code.

(e) TAG+OFFSET for label references shall not be used except for $TEXT references.

(f) All applications will use documented TaskMan utilities to interface with TaskMan globals.

(g) Line tags must be limited to eight characters.

(h) When a naked indicator is not defined on the same line as the corresponding naked reference, it must be documented.

1. Occurrences of this within a routine must be documented in the routine; all other occurrences must be documented in the technical manual.

2. All documentation must include the location that defines the naked indicator.

3. Uses of naked references in calls to utilities are exempt.

(i) Package routine names of the following forms will not be used:

1. NAMESPACE_I* (with the exception of routines created to support the INIT process); and

2. NAMESPACE _NTE* (with the exception of the package integrity routines).

(j) Percent Routines

1. Except for the Kernel, no application will distribute percent routines.

2. No percent-routines shall execute variables which could be set by a programmer prior to executing the code.

3. No percent-routine or MGR-routine will issue VIEW commands using variables as arguments which could be set by a programmer prior to executing the code.

(6) **Globals**

(a) The full evaluated length of a global reference must not exceed 127 characters and a single subscript must not exceed 63 characters.

(b) Application programs must not KILL entire (unsubscripted) globals.

(c) Except for the Kernel, SETting or KILLing of percent globals is not allowed.

(d) All globals except for UTILITY and TMP must be VA FileMan-compatible.

(e) Fields which contain executable code must have write protection in DD with "@", i.e., DD(file,field,9)="@" or be VA FileMan MUMPS data type.

(f) Only VA FileMan may directly create, change, or KILL any portion of the DD-global.

(g) All global variables executed by percent-routines must be in write-protected globals.

(7) **Options**

(a) Option selection must be made through MenuMan.

(b) All options in a package must be path-independent once the steps described in the technical manual for creating and killing package-wide variables have been taken. **NOTE:** *The utility XUP will be used to verify compliance with this standard.*

(8) **Device Handling**

(a) Routines designed to allow the user to select the output device must allow queuing.

1. Internal device selection must be designated by name; and

2. References to $I or internal number of the device name file shall not be used.

(b) Any output directed to a printer must leave the device at top-of-form when the output is finished.

(9) **Miscellaneous**

(a) Deletion of a data value, if permissible, must be executed with the entrance of the at-sign ("@").

(b) Except for Kernel implementation-specific routines, application software must use documented Kernel utilities to perform all implementation-specific functions.

(c) Any data element which may be interpreted as a number must contain no more than 15 significant digits.

(d) $NEXT shall not be used. All occurrences of $NEXT should have been removed by December 31, 1992.

(e) Packages may phase out documented (as callable from outside the package) utilities by providing a minimum 18-month warning to the PROGRAMMERS, CHIEF PROJECT MANAGERS, and SITEMANAGERS NATIONAL mail groups on Forum.

b. **Conventions**

(1) Kernel/VA FileMan convention should be used for editing data and for formatting of date and time.

(2) All entry points into a routine, except for the first line should be meaningful as mnemonics and include comments.

(3) Fields which contain executable code should be VA FileMan MUMPS data type.

(4) Options requiring package-wide variables should be able to create them should the option be removed from its normal menu path.

c. **Extensions.** The global TMP($J should be used as a scratch area within a routine. The global TMP("namespace",$J, should be used as scratch area between parts of a package.

SYSTEM-WIDE VARIABLES

If a variable is declared to be a system-wide variable, then any application setting this variable must conform to the definition of system-wide variable names:

1. DFN is the internal entry number to the Patient File.

2. VA ("PID") is the patient identifier up to 15 characters.
   VA ("BID") is the brief patient identifier up to 6 characters.
   VA ("PID") and VA ("BID") are defined through calls to the VADPT utilities.

3. AGE is the patient age in years from DT expressed as an integer.

4. DOB is the patient date of birth expressed in VA FileMan format.

5. SEX is the patient sex, either "M" or "F".

6. U is the "^" character.

7. DUZ is the internal entry of the user file.

8. DUZ(O) is this user's VA FileMan access code.

9. DUZ(2) is the internal entry of the institution file.

10. DUZ("AG") is the agency code (from the Kernel site param file).

11. DT is the current date in VA FileMan internal format.

12. DTIME is the integer value of the number of seconds the user has to respond to a timed READ (see App. 10A, subpar. 2(d)1).

13. IO is the hardware name of the last selected in/output device.

14. ION is the logical name of the IO device.

15. IOST is the name of the terminal type of the IO device.

16. IOM is the width of the IO device.

17. IOSL is the length of the IO device.

18. IOF is the code to start output at the top of a page (e.g., W @IOF).

Department of Veterans Affairs
Veterans Health Administration
Washington, DC 20420

M-11
Chapter 10

January 17, 1995

1. Transmitted is a new chapter to the Department of Veterans Affairs, Veterans Health Administration Manual, M-11, "Information Resources Management," Chapter 10, "Programming Standards and Conventions (SAC)."

2. Chapter 10 describes and provides procedures for the Information Resources Management (IRM) planning and acquisition process

3. **Filing Instructions**

Remove pages	Insert pages
	10-i
	10-1 through 10-3
	10A-1 through 10A-5
	10B-1

4. **RESCISSIONS:** VHA Circulars/Directives 10-85-93, 10-85-112, 10-85-116, 10-86-147, 10-87-19, 10-87-119, 10-87-122, and 10-87-123.

Kenneth W. Kizer, M.D., M.P.H.
Under Secretary for Health

Distribution: **RPC 1470 is assigned.**
FD

Printing Date: 1/95

1. Transmitted is a new manual to the Department of Veterans Affairs, Veterans Health Administration Manuals, M-11, "Information Resources Management," Chapters 1 through 17. **NOTE:** *Due to the length of this part, each Chapter will be transmitted separately.*

2. The principal changes include:

a. **Chapter 1:** Defines the over arching Information Resource Management (IRM) policy and the responsibilities of various offices, committees, and directorates for carrying out policy.

b. **Chapter 2:** Describes and provides procedures for the IRM planning and acquisition process.

c. **Chapter 3:** Sets forth Veterans Health Administration (VHA) policy relating to data administration and reports management, including reports control procedures.

d. **Chapter 4:** Establishes policy for the management of VHA's database, and procedures for design integrity and overall conformance to programming goals and standards.

e. **Chapter 5:** Establishes policy and guidance for the procurement of office automation equipment and software as part of VHA's health care information systems.

f. **Chapter 6:** Defines the responsibility for managing and administering VHA telecommunications resources.

g. **Chapter 7:** Provides guidance concerning the role of technology assessment as it relates to the management and operation of medical information systems.

h. **Chapter 8:** Defines the role and responsibility of Applications Requirements Groups in the development, design, and maintenance of VHA Decentralized Hospital Computer Program software.

i. **Chapter 9:** Sets forth the VHA policy regarding software management standards and requirements for the development, maintenance, and support of all software packages designated for national distribution.

j. **Chapter 10:** Describes and provides procedures for the IRM planning and acquisition process.

k. **Chapter 11:** Defines application documentation, documentation standards, and management of documentation of all VHA software.

l. **Chapter 12:** Establishes policy and procedures related to ensuring the quality of VHA developed software.

m. **Chapter 13:** Provides policy and guidance governing the archiving and purging of data from the VHA computer systems to ensure the ability to store current data in the system.

n. **Chapter 14:** Establishes policy for the provision of support to VHA facilities for the acquisition, implementation, and maintenance of automated hospital information systems to increase the effectiveness and quality of patient care.

o. **Chapter 15:** Establishes policy and responsibilities for training to support VHA IRM activities, both at the local and national level.

p. **Chapter 16:** Provides policy and procedures to ensure the protection of data, hardware, software, and storage media.

q. **Chapter 17:** Establishes operational guidelines for and defines the responsibilities of IRM Service, which unifies automated data processing, telecommunications, office automation, information collection, information management, and systems development.

3. **Filing Instructions**

Remove pages	Insert pages
	Cover through iv
	1-i through 1-19
	2-i through 2-8
	3-i through 3-6
	4-i through 4-5
	5-i through 5-3
	6-i through 6-4
	7-i through 7-6
	8-i through 8-7
	9-i through 9-21
	10-i through 10B-1
	11-i through 11A-5
	12-i through 12-4
	13-i through 13-2
	14-i through 14-4
	15-i through 15-3
	16-i through 16A-3
	17-i through 17-4

4. **RESCISSIONS:** VHA Circulars/Directives 10-85-93, 10-85-112, 10-85-116, 10-86-147, 10-87-19, 10-87-119, 10-87-122, and 10-87-123.

Kenneth W. Kizer, M.D., M.P.H.
Under Secretary for Health

Distribution: **RPC 1461 is assigned.**
FD

Printing Date: 1/95