NASA Contractor Report 182046

GCS SUPPORT/DEVELOPMENT SYSTEM CONFIGURATION DOCUMENT

Douglas S. Lowman

RESEARCH TRIANGLE INSTITUTE
Research Triangle Park, North Carolina

Contract NAS1-17964
May 1990

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

GCS SUPPORT/DEVELOPMENT SYSTEM CONFIGURATION DOCUMENT

Guidance and Control Software

RTCA DO-178A Document Number 9

Release number: 1.0

Prepared for:

NASA-Langley Research Center under contract
NAS1-17964; Task Assignment No. 8.

Prepared by:

Author(s): Douglas S. Lowman
Reviewer(s): RTI – Janet R. Dunham
Stephen E. Duncan
NASA – George B. Finelli

Software R & D Department
Center for Digital Systems Research
Research Triangle Institute
Research Triangle Park, North Carolina 27709

Preface

The GCS Support/Development System Configuration is document # 9 in a series of documents which fulfill the Radio Technical Commission for Aeronautics RTCA/DO-178A guidelines, "Software Considerations in Airborne Systems and Equipment Certification." These documents were prepared under contract with NASA-Langley Research Center.

This project consists of two complementary goals: first, to develop software for use in the Research Triangle Institute (RTI) software error studies research program sponsored by NASA-Langley Research Center [1]; second, to use and assess the RTCA/DO-178A guidelines for the Federal Aviation Administration (FAA). The two goals are complementary in that the use of the structured DO-178A guidelines in the development of the software will ensure that the test specimens of software have been developed according to the industry standards for flight critical software. The error studies research analyses will then be conducted using high quality software specimens.

The implementations will be subjected to two different software testing environments: verification of each implementation according to the RTCA/DO–178A guidelines and replicated random testing in a configuration which runs more than one test specimen at a time. The term *implementations* refers to bodies of code written by different programmers, while a *version* is a piece of code at a particular state (i.e., version 5.0 is the result of frame testing). This research effort involves the gathering of product and process data from every phase of software development for later analysis. More information on the goals of the Guidance and Control Software (GCS) project are available in the *GCS Plan for Software Aspects of Certification.*

The series consists of the following documents:

- *GCS Configuration Index* Document no. 1
- *GCS Development Specification* Document no. 2
- *GCS Design Descriptions* One for each software implementation. Document no. 3

- *GCS Programmer's Manual* Document no. 4, includes Software Design Standards, document no. 12.
- *GCS Configuration Management Plan* Document no. 5A
- *Software Quality Assurance Plan for GCS* Document no. 5B
- *GCS Source Listing* One for each software implementation. Document no. 6
- *GCS Source Code* One for each software implementation. Document no. 7
- *GCS Executable Object Code* One for each software implementation. Not available on hardcopy. Document no. 8
- *GCS Support/Development System Configuration Description* Document no. 9
- *GCS Accomplishment Summary* Document no. 10
- *Software Verification Plan for GCS* Document no. 11
- *GCS Development Specification Review Description* Document no. 11A
- *GCS Simulator (GCS_SIM) System Description* Document no. 13
- *GCS Simulator (GCS_SIM) Certification Plan* Document no. 13A
- *GCS Plan for Software Aspects of Certification* Document no. 14

Contents

1 RTI System Environment

The Research Triangle Institute (RTI) Center for Digital Systems Research (CDSR) Software R & D Department (SRDD) computer system environment consists of eight VAXstation 2000, and four VAXstation II/GPX workstations. All SRDD workstations are using the VAX/VMS 5.1 operating system and are all part of a Local Area VAXcluster (LAVC) that is connected over a ThickWire Ethernet backbone to several ThinWire multidrop Ethernet segments.

Note:

- Each of the SRDD workstations are *capable* of running all of the tools described in the following section, however due to size constraints, not all of the tools are necessarily installed on all machines.
- For the purposes of this experiment, the operating system, compiler, and validation tools will not be upgraded or modified once GCS implementation validation has begun through the end of the experiment.
- Use of some of the tools described in this document are required. For information regarding the required use of a tool, refer to the *GCS Programmer's Manual.* The use of any tool not listed in the *GCS Programmer's Manual* needs to be approved by the project management team.

2 RTI Software Programming Environment

The remaining sections of this document describe the tools and layered product utilities available to a GCS programmer at RTI. (The software version of each of the tools mentioned in this document can be found in Appendix A.)

Many of the tools described in this section are part of Digital Equipment Corporation's software layered product library or are part of the VAX/VMS baseline system [2]. The purpose of this document is to provide brief abstracts about each of the tools that are available to GCS programmers. It is

not designed as a tutorial or user's manual for these tools. For more information about how to use any of the tools mentioned, consult the references listed at the end of this document.

2.1 Structured Design CASE Tool - Team*work*

Team*work*[1] is a Computer Aided Software Engineering (CASE) tool which allows the user to analyze or design a software system using a variety of structured analysis or design methods.

The team*work* tool will be used to aid in the structured design of the GCS applications. Team*work* is comprised of several tools that are integrated into a single user interface, making all of the tools available to the programmer/designer at the same time.

Some major parts of team*work* include, but are not limited to, the following components:

IM - An information modeling tool for creating entity-relationship diagrams,

SA - The baseline structured analysis tool,

RT - An extension of SA that allows description of real-time systems, and

SD - A structure design tool that follows the Ward and Mellor approach.

The theory behind the underlying team*work* tool components has been adapted from the structured analysis or structured design methods as described by Derek Hatley[3], Tom DeMarco[4], and Paul Ward and Steven Mellor[5].

2.2 Text Editors

There are many different variants of text editors available at RTI. Three of the Digital Equipment Corporation's supported editors, mentioned below, are more commonly used by the SRDD staff.

[1] Team*work* is a registered trademark of Cadre Technologies, Inc.

Information about other public domain editors that are available to users of the CDSR LAVC can be obtained from one of the members of the VAXcluster support staff.

2.2.1 EDIT/EDT

The *Edit/edt* command invokes the VAX EDT [6] interactive text editor. The /EDT qualifier is not required, because EDT is the VAX/VMS default editor.

2.2.2 EDIT/TPU

The *Edit/tpu* command invokes the VAX Text Processing Utility (VAXTPU) [7]. By default, the Extensible VAX Editor (EVE) is used as the interface for VAXTPU. To invoke VAXTPU with the EDT Keypad Emulator interface, define the logical TPUSECINI to point to the section file for that interface.

2.2.3 Language Sensitive Editor

The VAX Language-Sensitive Editor (LSE) [8] is a multi-language advanced text editor (with language templates) specifically designed for software development. LSE's language templates allow a programmer to "expand" symbolic representations of statements into complete statement templates. LSE allows module compilation and analysis without ever leaving the editor through its extended user interface.

2.3 Programmming Language : FORTRAN

Although there are a variety of programming languages available for use with the SRDD workstations, requirements for this project preclude a programmer from using any languages except FORTRAN for the purposes of the Guidance and Control Software project. The VAX/VMS FORTRAN compiler creates object code which can then be linked into an executable image.

VAX FORTRAN [9] is an implementation of full language FORTRAN-77 conforming to American National Standard FORTRAN, ANSI X3.9-

1978. It includes optional support for programs conforming to the previous standard, ANSI X3.9-1966. VAX FORTRAN meets the Federal Information Processing Standard Publication (FIPS-69) requirements by conforming to the ANSI Standard and by including a flagger. The flagger optionally produces diagnostic messages for syntax and/or source form elements, which do not conform to the Full-Level ANSI FORTRAN X3.9-1978 Standard.

The shareable, reentrant compiler operates under the VAX/VMS Operating System. It globally optimizes source programs while taking advantage of the floating point and character string instruction set and the VMS virtual memory system.

2.4 VAX Debugger

The VAX/VMS Debugger [10] is a full-screen, interactive program debugging tool that allows a programmer to observe and manipulate a program's behavior as it executes. The VAX/VMS Debugger allows a variety of advanced full-screen debugging features (e.g., line by line code execution, examination of variable contents, visual verification of code execution paths).

2.5 Module Management Tools

In any large software project, it is important to maintain some form of configuration control in order to better organize module management. It is also helpful to have some way to automatically rebuild a series of related modules into a program, re-compiling only those modules that have changed since the last program re-build.

For the purposes of the GCS project, DEC/CMS (Code Management System) and DEC/MMS (Module Management System) will be used to assist the GCS programmers and the project management team in maintaining each GCS programmer's code. (For instructions detailing the use of CMS and MMS see the *GCS Configuration Management Plan*). Through the use of CMS, programmers will be able to re-create any ***version*** of their code at any stage during its development. MMS will allow the automatic rebuilding of any programmer's system or ***implementation*** based on rule-specified criteria.

2.5.1 DEC/CMS

VAX DEC/CMS (Code Management System) [11] is a software library system that facilitates the development and maintenance of software systems. Software systems are divided into different functional components that are, in turn, organized into sets of one or more files. During development, one or more programmers continually make changes to these files. VAX DEC/CMS helps manage the files during development (and later during maintenance) by storing the files in a project library, tracking changes, and monitoring access to the library. VAX DEC/CMS also supplies a means of manipulating different combinations of files within a library. The ability to formalize these combinations provides a focus for system design and a means of organizing the files within a library.

2.5.2 DEC/MMS

VAX DEC/MMS [12] is a software tool designed to enhance programmer productivity. It determines what components in a described software system have changed and rebuilds the system in an optimal way. When some modules of a software system are modified, dependent modules may need to be recompiled. VAX DEC/MMS determines which modules need to be recompiled and performs the appropriate actions to ensure that the software system is recompiled and linked with all the latest changes.

Additionally, VAX DEC/MMS has the ability to interact with VAX DEC/CMS (Code Management Systems) to provide an enhanced software development package by providing programmers with a mechanism for optimal system re-builds as well as providing configuration control through project libraries.

2.6 Testing and Validation Tools

This section describes some of the testing and validation tools that may be used by the GCS programmers during the development of their code for the GCS project. The GCS project management team has developed the *Software Verification Plan for GCS* that utilizes some of the testing/validation tools listed below.

2.6.1 DEC/TEST MANAGER

VAX DEC/Test Manager [13] automates regression testing by executing user-supplied tests and automatically comparing the results with the expected test results. VAX DEC/Test Manager gives the software engineer flexibility in organizing tests, selecting tests for execution, and verifying and reviewing test results. Batch applications or interactive applications which input and output text to terminals can be tested using VAX DEC/Test Manager. VAX DEC/Test Manager lets the user create a library area for test result storage.

It allows users to:

- Create descriptions of their software tests and store these descriptions in CMS libraries,
- Group these test descriptions into meaningful combinations for later runs,
- Modify or display the test description or groups,
- Execute specific tests, groups of tests, or combinations of groups of tests,
- Record and replay a terminal session and save the image of terminal screen as a benchmark for interactive applications which input and output text to terminals,
- Compare the results of the executed tests with benchmark test results to determine differences,
- View test results interactively, and
- Update benchmarks as needed.

2.6.2 VAX Performance and Coverage Analyzer (PCA)

The VAX Performance and Coverage Analyzer (PCA) [14] is used to help analyze the execution behavior of application programs. The VAX Performance and Coverage Analyzer has two functions. First, it can pin-point

execution bottlenecks and other performance problems in applications programs. Second, it provides test coverage analysis by measuring the parts of a program that are executed or not executed by a given set of test data.

PCA has the capability of collecting and analyzing a variety of performance data. Some of the kinds of data that can be collected and analyzed with PCA include: (1) Program Counter (PC) sampling data, (2) CPU sampling data, (3) Page Fault data, (4) System Services data, (5) Input/Output data, (6) Execution counts, (7) Test coverage, and (8) Tasking data. The VAX Performance and Coverage Analyzer is an aid in tuning the performance and testing of applications programs. It is not a tool for the analysis of operating system performance or for use as an aid in hardware resource planning.

2.6.3 VAX Source Code Analyzer (SCA)

The VAX Source Code Analyzer (SCA) [8] is an interactive, multilanguage, source code cross-reference and static analysis tool that is designed to aid developers in understanding the complexities of large-scale software systems. SCA can (1) provide call-tree information, (2) preform module interface consistency checks, (3) locate all symbol definitions and references, (4) and locate file references.

SCA provides a programmer with the ability to statically analyze the above mentioned components of programs prior to executing them.

2.7 Project/Programmer Communication Tools

This section describes which tools will be used by the GCS programmers and the GCS project management team to interact, distribute information electronically, and document activities throughout the project.

2.7.1 VAX Notes

VAX Notes [15] is a computer conferencing software product designed to provide users with the capability of creating and accessing online conferences or meetings. Computer conferencing is a new electronic messaging technology which lets users conduct meetings with people in different geographic locations via computer so that participants can join in a discussion

from their own desks at a time of their own choice. It also offers the advantage of keeping a detailed record of the proceedings of a meeting, which can be searched by a variety of criteria, such as name of participant, subject, or keyword.

VAX Notes can also be used for a variety of similar applications, such as an "electronic bulletin board" or collaborative document authoring and review.

CDSR VAX Notes Conferences Information about the CDSR VAXcluster, new products, and other general information is available by subscribing to any or all of the following CDSR conferences:

- CDSR-NOTES-INTRODUCTION contains introductory information about CDSR conferences.
- NEW_CONFERENCES announces all new conferences that are open to the CDSR VAX/VMS user public.
- TEX_NOTES is a question/answer bulletin board that discusses LaTeX and TeX questions as they apply to the VAXcluster.

2.7.2 MAIL

The *MAIL* command invokes the VAX/VMS Personal Mail Utility (MAIL) [16], which is used to send messages to other users of the system. For a complete description of the VAX/VMS Personal Mail Utility, including information about the MAIL command and its qualifiers, see the VAX/VMS Mail Utility Reference Manual.

2.8 LaTeX: A Documentation Preparation System

LaTeX [17] is a high-level document preparation facility by Leslie Lamport built on top of Donald Knuth's document formatting and typesetting system, TeX. LaTeX is designed to add to and simplify the typesetting commands of TeX while allowing the user to concentrate on writing rather than text formatting.

2.9 Finding Additional Information about CDSR Tools

More information about tools the may be used in the CDSR LAVC may be obtained by either examining reference manuals or by checking with an SRDD staff member. (Note: Use of any tools not mentioned in this document or in the *GCS Programmer's Manual* must be approved by the GCS project management team).

2.9.1 Reference Manual Locations

Below is a list VAX/VMS reference manual locations:

- Manuals for the Digital Equipment Corporation Software Library are stored in Room 210.
- VAX/VMS General Information and System Reference Manuals can be found in Room 219.
- Manuals for LaTeX are available for use on request.

A Appendix – Software Versions Used

Software	*Version*
VAX/VMS Operating System	4.7
team*work*	V2.3.2

Editors	*Version*
EDT	EDT V3.10-00
Text Processing Utility	TPU Version V1.2
Language Sensitive Editor	LSEDIT V2.1-39

Software	*Version*
FORTRAN	V4.8-283
Linker	VAX-11 Linker V04-00
Debug	DEBUG V4.7-1

Module Management Tools	*Version*
Code Management System	DEC/CMS V2.3-00
Module Management System	DEC/MMS V2.2

Testing and Validation Tools	*Version*
DEC Test Manager	TEST MGR. V2.2
Performance Coverage Analyzer	PCA V2.0-1
Static Code Analyzer	VAX SCA V1.1-90

Project Communication Tools	*Version*
Notes	VAX NOTES V01.2
Mail	X-9
LaTeX	LaTeX Version 2.09

References

[1] George B. Finelli. Results of software error-data experiments. In *AIAA/AHS/ASEE Aircraft Design, Systems and Operations Conference*, Atlanta, GA, September 1988.

[2] *Introduction to VAX/VMS.* Digital Equipment Corporation, Maynard, Massachusetts, September 1984.

[3] Derek J. Hatley and Imtiaz A. Pirbhai. *Strategies for Real-Time System Specification.* Dorset House Publishing Company, New York, New York, 1987.

[4] Tom DeMarco. *Structured Analysis and System Specification.* YOURDON Inc., 1133 Avenue of the Americas, New York, NY 10036, 1978.

[5] Paul Ward and Steven Mellor. *Strucured Development for Real-Time Systems.* Prentice-Hall Inc., Englewood Cliffs, NJ, 1985.

[6] *Text Processing: EDT Reference Manual.* Digital Equipment Corporation, Maynard, Massachusetts, September 1984.

[7] *Text Processing: VAXTPU Reference Utility Manual.* Digital Equipment Corporation, Maynard, Massachusetts, April 1986.

[8] *Guide to VAX Language Sensitive Editor and VAX Source Code Analyzer.* Digital Equipment Corporation, Maynard, Massachusetts, August 1987.

[9] *Programming in VAX FORTRAN.* Digital Equipment Corporation, Maynard, Massachusetts, September 1984.

[10] *VAX/VMS Debugger Reference Manual.* Digital Equipment Corporation, Maynard, Massachusetts, April 1986.

[11] *Guide to VAX DEC/Code Management System.* Digital Equipment Corporation, Maynard, Massachusetts, April 1987.

[12] *Guide to VAX DEC/Module Management System.* Digital Equipment Corporation, Maynard, Massachusetts, April 1987.

[13] *Guide to VAX DEC/Test Manager.* Digital Equipment Corporation, Maynard, Massachusetts, October 1986.

[14] *Guide to VAX Performance and Coverage Analyzer.* Digital Equipment Corporation, Maynard, Massachusetts, August 1987.

[15] *Guide to VAX Notes.* Digital Equipment Corporation, Maynard, Massachusetts, March 1986.

[16] *Mail Utility Reference Manual.* Digital Equipment Corporation, Maynard, Massachusetts, September 1984.

[17] Leslie Lamport. LaTeX: *A Document Preparation System.* Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1986.

Report Documentation Page

NASA National Aeronautics and Space Administration

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-182046		

4. Title and Subtitle	5. Report Date
GCS Support/Development System Configuration Document	May 1990
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Douglas S. Lowman	
	10. Work Unit No. 505-66-21-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Research Triangle Institute Research Triangle Park, NC 27709	NAS1-17964
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Technical Monitor: George B. Finelli, Langley Research Center
Task 8 Report

16. Abstract

This document describes in detail the software programming environment used in the development of Guidance and Control Software (GCS) implementations used in a software error studies experiment conducted by the Research Triangle Institute (RTI) and the NASA Langley Research Center. This document fulfills the Radio Technical Commission for Aeronautics RTCA/DO-178A guidelines, "Software Considerations in Airborne Systems and Equipment Certification" requirements for document #9 in which the hardware, software, and processes used to develop and maintain the software for the GCS project are described. The software programming environment for GCS largely consists of tools that are included in Digital Equipment Corporation's software layered product library or are a part of the VAX/VMS baseline system.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Software Programming Environment Guidance and Control Software (GCS) Support Environment System Configuration	Unclassified-Unlimited Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	16	A03

NASA FORM 1626 OCT 86