

AD-A078 438

GAERTNER (W W) RESEARCH INC STAMFORD CONN
PROGRAMMING SUPPORT LIBRARY. VOLUME I. FUNCTIONAL REQUIREMENTS.(U)
OCT 79 C M TURCIO , W M SCHREYER , N A ADAMS F30602-78-C-0103
RADC-TR-79-241-VOL-1 NL

UNCLASSIFIED

1 OF 1
AD
A078438



END
DATE
FILMED
1-80
DDC

✓
RADC-TR-79-241, Vol I (of two)
Final Technical Report
October 1979

12
LEVEL



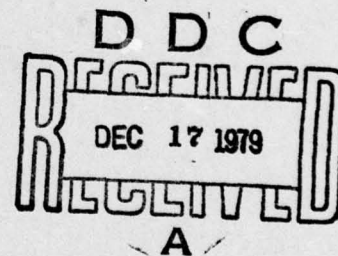
PROGRAMMING SUPPORT LIBRARY

Functional Requirements

W. W. Gaertner Research Inc.

Carolyn M. Turcio
William M. Schreyer
Norman A. Adams
Wolfgang W. Gaertner

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED



ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

79 12 17 009

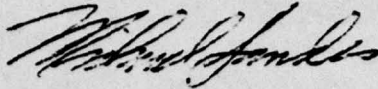
ADA 078438

DDC FILE COPY

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

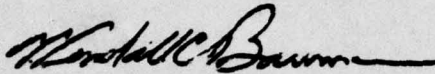
RADC-TR-79-241, Vol I (of two) has been reviewed and is approved for publication.

APPROVED:



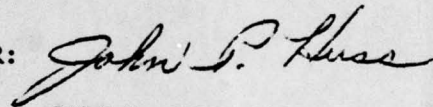
MICHAEL LANDES
Project Engineer

APPROVED:



WENDALL C. BAUMAN, COLONEL, USAF
Chief, Information Sciences Division

FOR THE COMMANDER:



JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIE), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) RADC-TR-79-241, Vol I (of two)	2. GOVT ACCESSION NO. (19) TR-79-242-VOL-2	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) PROGRAMMING SUPPORT LIBRARY, Volume I. Functional Requirements		5. TYPE OF REPORT & PERIOD COVERED (9) Final Technical Report
7. AUTHOR(s) (10) Carolyn M. /Turcio, Norman A. /Adams William M. /Schreyer, Wolfgang W. /Gaertner		8. CONTRACT OR GRANT NUMBER(s) F30602-78-C-0103
9. PERFORMING ORGANIZATION NAME AND ADDRESS W. W. Gaertner Research Inc. 205 Saddle Hill Road Stanford CT 06903		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63728F (12) 42 (16) 25310301 (17) 03
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (ISIE) Griffiss AFB NY 13441		12. REPORT DATE Oct 1979
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same 388 544		13. NUMBER OF PAGES 42
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Michael Landes (ISIE)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Programming Support Library Software Engineering Structured Programming Software Development Management		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report, Vol I contains the generic functional requirements for a programming support library. The requirements are presented at two levels of detail. Chapter 2 contains a brief summary of the requirements separated into Basic and Full PSL requirements. Chapter 3 contains the full wording of the requirements organized into nine categories. The specification for each individual requirement is followed by a brief description of its purpose. This volume may be used as the specification for a Full or Basic PSL under a software development contract. Volume II will be published at a later date.		

DD FORM 1 JAN 73 1473

388 544 UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

Chapter No.	Page No.
1. INTRODUCTION	1
2. SUMMARY OF FUNCTIONAL REQUIREMENTS	2
2.1 Basic PSL	2
2.2 Full PSL	5
3. DETAILED FUNCTIONAL REQUIREMENTS	9
3.1 Source Data Maintenance	10
3.1.1 Basic Source Data Maintenance Requirements	10
3.1.1.a Data File Storage	10
3.1.1.b Data Access	11
3.1.1.c Library Backup Capability	11
3.1.1.d Data Maintenance	11
3.1.2 Full Source Data Maintenance Requirements	12
3.1.2.a Data File Storage	13
3.1.2.b Full Data Maintenance	13
3.2 Output Processing	14
3.2.1 Basic Output Processing Requirements	14
3.2.1.a Library Control Listings	14
3.2.1.b Source Data Listings	14
3.2.1.c Control Data	14
3.2.2 Full Output Processing Requirements	15
3.2.2.a Full Magnetic Tape Output	15
3.2.2.b Full Punched Card Output	15
3.2.2.c Full Management Control Listings	15
3.2.2.d Full File Directory and Listings by Programmer	15
3.2.2.e Full Automatic Indentation of Source Code Listings	15
3.2.2.f Full Data Scanning Capability	16
3.2.2.g Full Directory Availability Printout	16
3.3 Programming Language Support	17
3.3.1 Basic Programming Language Support Requirements	17
3.3.1.a Compiler Interface	17
3.3.1.b Load Module Generation	17

Accession For	
NTIS GRA&I	
FOR AS	
Unannounced	
Justification	
By	
Date	
Initials	
A	

TABLE OF CONTENTS

(continued)

Chapter No.	Page No.
3.3.2 Full Programming Language Support Requirements	18
3.3.2.a Full Top Down Structured Programming Support	18
3.4 Library System Maintenance	19
3.4.1 Basic Library System Maintenance Requirements	19
3.4.1.a Library Installation Support	19
3.4.1.b Library Maintenance Support	19
3.4.1.c Library Termination Support	19
3.4.2 Full Library System Maintenance Requirements	19
3.4.2.a Full Library Installation Support	19
3.5 Data Security	21
3.5.1 Basic Data Security Requirements	21
3.5.2 Full Data Security Requirements	21
3.5.2.a Full Data Integrity	21
3.5.2.b Full Data Protection	22
3.6 Management Data Collection and Reporting	23
3.6.1 Basic Management Data Collection and Reporting Requirements	23
3.6.2 Full Management Data Collection and Reporting Requirements	23
3.6.2.a Full Collecting	23
3.6.2.b Full Updating	24
3.6.2.c Full Accumulating	25
3.6.2.d Full Archiving	25
3.6.2.e Full Reporting	26
3.7 Documentation Support	28
3.7.1 Basic Documentation Support Requirements	28
3.7.2 Full Documentation Support Requirements	28

TABLE OF CONTENTS

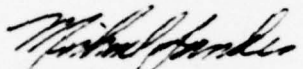
(continued)

Chapter No.	Page No.
3.8 On-Line Terminal Implementation Requirements	30
3.8.1 Basic On-Line Terminal Implementation Requirements	30
3.8.2 Full On-Line Terminal Implementation Requirements	31
3.9 General Requirements	33
3.9.1 Basic General Requirements	33
3.9.2 Full General Requirements	33
3.9.2.b Full Subroutine Support for User Programs	33

EVALUATION

This report, Vol I, contains the generic functional requirements for a programming support library (PSL). The requirements are presented at two levels of detail. Chapter Two contains a brief summary of the requirements separated into Basic and Full PSL requirements. Chapter Three contains the full wording of the requirements organized into nine categories. The specification for each individual requirement is followed by a brief description of its purpose.

This volume will serve as a complete functional specification for a full or Basic program support library in a software development contract. It's content will also be used to provide a completely generic functional description of the requirements for a program support library in an appendix of the Rome Air Development Center Software Development Specification.


MICHAEL LANDES
Project Engineer

1. INTRODUCTION

This Report on the requirements for a Programming Support Library is organized into two volumes.

Volume I contains the Functional Requirements and Volume II contains the Guidelines for their Implementation, including many examples generated on a variety of computer systems.

In this Volume I, the Functional Requirements are presented at two different levels of detail: Chapter 2 contains a brief summary of the requirements, separated into the requirements for a Basic PSL (Chapter 2.1) and a Full PSL (Chapter 2.2).

Chapter 3 contains the full wording of the functional requirements organized into 9 different categories. The specification for each individual requirement is followed by a brief description of the purpose of the particular paragraph and its classification (Basic or Full).

Volume I of this Final Report may therefore be used as the specification for the implementation of a Basic or Full Programming Support Library (PSL) under a Government software development contract.

Volume II of this Final Report follows the organizational structure of Volume I and provides implementation guidelines for each paragraph, with examples drawn from several different types of computer systems ranging from minicomputers to large in-house mainframes and time-share systems. The examples serve as further illustrations of the intent of each PSL requirement and can therefore be used by a DP manager to rapidly assess whether his present installation already complies with a specific PSL requirement. If not, the guidelines and examples will allow him to implement a requirement in a cost effective manner.

All references to the "Structured Programmed Series" pertain to RADC-TR-74-300 which consists of 15 volumes. It is available from the NTIS under catalog number AD/A-003 339.

2. SUMMARY OF FUNCTIONAL REQUIREMENTS

2.1 Basic PSL

A Basic PSL shall include the following:

- a. Overview. A data repository shall be maintained in two forms: Data is stored in an Internal Library which is computer-resident in machine readable form. The identical data is also stored in an External Library which is in hard copy form, in organized project notebooks or post binders. Procedures shall be established for this repository to define:
 1. The delegation of clerical and record keeping operations associated with the programming process and the maintenance of both the Internal and External Libraries.
 2. The delegation of all machine operations with regard to such areas as project initiation/termination, program-test philosophy, output media/frequency, etc.
 3. The requirements for the recording, cataloging, and filing of all code generated in the project, both intermediate and final; for the retention of superceded, corrected code for stated retention periods.
- b. Source Data Maintenance. The PSL shall be structurally organized to provide the following capabilities:
 1. Data File Storage - The capability to store within the Internal Library files of source and object code, control language, code enabling the loading and execution of several files stored, storage of files in a compressed form from which they can be restored, etc.
 2. Data Access - Direct access of a single unit of stored data in a timely fashion.

3. Library Backup - A capability to recover from inadvertent loss or destruction of data. This involves backup storage on a storage unit independent of the Internal Library (e.g., magnetic tape, disk pack, punched cards) and also the regeneration of the library data files from this backup storage. This backup capability shall selectively regenerate and restore data so that it is possible to recover portions of the total PSL data base without the need to perform full storage dump and restore operations.
 4. Data Maintenance - The capability to add, delete, or replace one or more units of data in an Internal Library file; to copy one or more segments of data from one library file to another.
- c. Output Processing. This involves the output of data stored in the file(s) and the output of related control data for use by both programming and management personnel. As a minimum, data output shall include:
1. Project Status Listings - Status information with regard to a section within a project to include such items as segment name, date/time information retrieved, storage space allocated and used, segment language, creation/last-update date, number of lines per segment.
 2. Source Listings - Printed output of any segment which is in card-image format such as Source, PDL, Link, Job, Test and Text. The output contains such items as print date/time, language, segment type, line numbering and nesting-indentation, segment update information.
- d. Programming Language Support. This capability includes the validation and compilation of program source code stored in the PSL. The PSL shall interface with or invoke precompilers and compilers (JOVIAL, FORTRAN, and COBOL) for the purposes of syntax checking of both structured and unstructured source code, compiling the source code, and storing the resultant object module within the library. Further, to provide

load module generation to execute the necessary computer programs (i.e., to convert object modules into executable programs, to store the resulting machine code in the library, to load the program for subsequent execution).

- e. Library System Maintenance. To provide procedures to generate and maintain the PSL system. This shall include PSL installation; initialization of PSL projects; creation of any of the eight standard PSL sections (data files); termination (deletion) of a section, library, or project; backup/restore either on a project, library, or section level.
- f. Privacy/Access Constraints. The PSL shall provide for the integrity of the data stored within the PSL. This includes maintaining the separation of different versions of projects/programs under development, control over inadvertent destruction of data, and protection of data from unauthorized access.
- g. Management Data Collection and Reporting. No Basic Requirements.
- h. Documentation Support. Basic requirements are met by Basic Source Data Maintenance.
- i. On-Line Terminal Implementation Requirements. Use of on-line terminals requires: Support of terminals (both CRT and typewriter) connected to the computer directly or via communications links, a Remote Job Entry facility, inter-terminal communications, minimal data loss in system failure, and a facility to easily scan data files at the terminal.

2.2 Full PSL

A full PSL with management data collection and reporting shall include all the functions defined for the Basic PSL above and the additional capability of providing full privacy/access constraints, documentation support and management data collection/reporting as specified herein:

a. Source Data Maintenance

1. Data File Maintenance - A Full PSL will allow record modification without retyping the entire record (specific character selection/replacement) temporary changes in a file, automatic generation of program stubs, and a facility to nondestructively merge two or more PSL files from two or more PSL Libraries.

b. Output Processing. A Full PSL will support:

1. Magnetic Tape Output - copying one or more records, or, one or more files to magnetic tape.
2. Punched Card Output - punching cards from one or more records, or, one or more files.
3. Management Control listings - covered in paragraph on Management Data Collection and Reporting.
4. List by Programmer - generate a list of all files and/or listings thereof, that are identified with a specific programmer.
5. Automatic Indentation of Source Code.
6. Data Scanning - searching for a particular character string in every line of every file in a PSL.
7. Directory Availability - listing of pertinent information about a particular directory.

c. Programming Language Support

1. Top Down Structured Programming (TDSP) Support - flagging source code constructions that do not conform to TDSP philosophy; such as, explicit branches, large program blocks, and source lines with more than one statement.

d. Library Maintenance Support - this includes:

1. Library Installation Support - definition of hardware configuration at installation time, and system generation facilities allowing user to generate a PSL.

- e. Privacy/Access Constraints. The PSL shall provide control over the integrity and security of the data stored within the PSL; namely control over different versions of projects/programs under development, control over the inadvertent destruction of data, and control with regard to the protection of data from unauthorized access. The design requirements listed herein are not intended to support the protection of military classified data. Protection of classified data is left primarily to physical security measures (i.e., control of computer log-on, installation of computer terminals within secure areas). (See Sec. 5.5 and 5.7.5 of the Specification for military classification provisions.)
1. Data Integrity - Procedures shall be established for maintaining control over multiple versions of a program or program system and also control over various change levels of program source segments within a specific version. A version is defined as a complete program or program system as it exists at a specific point in time, usually related to a specific milestone in the development cycle. The objective of version and level control is to allow a development and/or maintenance group to manipulate multiple versions of the same program system' (e.g., operational version, development version, maintenance version) stored in separate libraries without unnecessary duplication of source or object code. With version control, it is possible to merge data from several libraries for the purpose of performing tests.
 2. Data Protection - The PSL shall contain facilities for the protection of data stored in a project from unauthorized access and update. Procedures shall be established to limit data destruction (and likewise data recovery) to authorized personnel only. At a minimum, these procedures shall establish controls at the project, library, and data file positions as defined for the designated PSL installation.

- b. Management Data Collection and Reporting (MDCR). This capability involves the collection and storage of data related to program development and maintenance and subsequently, the generation of management reports containing the data and/or summaries of data. The minimum requirements for the MDCR are as follows:
1. Collecting - An automatic function of the PSL that must include: Counting input source code lines, gathering and storing pertinent information about files, gathering and storing pertinent information about PSL functions, and allowing addition of user routines.
 2. Updating - A PSL function allowing modification of the data base in which collection data was stored. This function must: Allow allocation of data areas, adding data, deleting data, replacing data, and termination of data areas.
 3. Accumulating - A PSL facility to extract, summarize, or process management data stored in the PSL. This facility must: Summarize data at various hierarchical levels, process data for maxima, minima, and averages, retain data produced.
 4. Archiving - Historical information storing facility that must provide: Storage (on secondary storage devices) all data accumulated at the system level, and backup of management data.
 5. Reporting - A report generation facility to output information in a convenient and meaningful way. Besides producing the various reports needed, this facility must: Produce reports on an exception basis, produce cyclic versions of reports, and allow adding of user routines.
- g. Documentation Support. The basic requirements include the storage, update and output of PSL data as specified for the MDCR. Other desirable, but not required, capabilities are as follows:
1. Print in a user specified sequence one or more segments of program stored in the PSL. There also must be a capability to merge segments of program design language, program source code and textual data into one output listing.

2. Format an output listing in accordance with the following user supplied information:
Header to be printed at top of page; spacing between segments of data; spacing between lines of output; and number of lines to print on a page.
 3. Automatic page numbering beginning with a user-supplied page number for Page #1.
 4. Print a title page containing: Document title; Document date; Author's name; Organization and address.
 5. Generate a magnetic tape in print image form. This allows the distribution of documentation in machine readable form for easy storage and reproduction.
- h. On-Line Terminal Implementation Requirements. Full PSL terminal support must include: Reporting terminal usage statistics, promoting security through access limitation, and a HELP facility.
- i. General Requirements. A Full PSL must give the user conveniences such as allowing more than one file to be specified in all jobs requiring access to data files, and a subroutine library to perform routine PSL functions.

3.0 DETAILED FUNCTIONAL REQUIREMENTS

The description of the PSL functional requirements is divided into nine functional areas listed below. The order of the listing has no particular significance except that the basic requirements are identified only for (a) through (e).

- (a) Source Data Maintenance
- (b) Output Processing
- (c) Programming Language Support
- (d) Library System Maintenance
- (e) Data Security
- (f) Management Data Collection and Reporting
- (g) Documentation Support
- (h) On-Line Terminal Implementation Requirements
- (i) General Requirements.

The functional requirements within each of the nine areas are described as either basic or full. The purpose of this subdivision is to present the functional requirements in a manner which addresses two levels of potential users.

The following paragraphs contain descriptions of the functional requirements.

The 9 functional areas listed above are presented in sub-chapters 3.1 through 3.9 respectively. Within each functional area, the Basic requirements are presented first (in sub-chapters 3.1.1 and 3.2.1, etc.), followed by the Full Requirements (sub-chapters 3.1.2, 3.2.2, etc.). A user who is only interested in the Basic Requirements may therefore ignore the Full Requirements (sub-chapters 3.X.2, etc.), whereas a user who is required to implement a Full PSL must comply with the entire text of Chapter 3.

Volume II of this Report contains guidelines and examples for the implementation of all these requirements, organized according to the same numbering system as this Chapter 3 of Volume I. As an example, guidelines for the implementation of requirement 3.1.1.a.3 in this Volume I, will be found in Chapter 3.1.1.a.3 in Volume II, etc.

3.1 Source Data Maintenance

This functional area addresses the physical storage and maintenance of programming related data. The basic requirements define the minimum capabilities needed to store, access, maintain and provide backup for the data. The full requirements define additional capabilities which increase storage efficiency and improve maintenance support.

3.1.1 Basic Source Data Maintenance Requirements

3.1.1.a Data File Storage

Physical storage must be provided for the following types of data.

1. Program source code - Programmer coded input to a program language compiler or assembler (e.g., COBOL source statements).
2. Program object modules - Program code that results from the execution of a compiler or assembler.
3. Program load modules - Program code in a form ready to be loaded into a computer for execution.
4. Job control data - Control statements used to control the execution of computer jobs.
5. Test data - Data used for testing programs and/or program systems during the development and maintenance of the program/system and for performing system and acceptance testing.
6. Program Design Language (PDL) statements - English-like statements that follow the basic rules of structured programming and are used to define the program structure and logic.
7. Textual data - Standard text which is written primarily by programmers for use as program documentation. (Although the intent is to support program documentation, any textual data supporting the programming process could be stored and maintained in the PSL.)

8. Other data - Any form of data that can be entered via punched cards, magnetic tape, or on-line terminals and stored in a computer. What data is to be stored is determined by the user of the system.
9. Provision must be made for storing data (except for object and load modules) in a compressed form to minimize storage space. This requirement involves:
 - 9.1 The compression of data being stored to eliminate leading and trailing blank characters in each record, as well as intervening blanks when more than three appear contiguously. This will result in storage of significant characters only.
 - 9.2 Restoration of data from its compressed library storage form to its original uncompressed form.

3.1.1.b Data Access

Data access - Direct access of a single record of stored data must be provided via an on-line storage device when access to that data is deemed necessary to the user. Such access is necessary during execution of standard PSL jobs (e.g., data file updating or other maintenance operations).

3.1.1.c Library Backup Capability

Library backup capability - A capability must be provided to recover from inadvertent loss or destruction of data. This involves the following two functions.

1. Backup storage on a storage unit independent of the master file (e.g., magnetic tape, disk pack, punched cards).
2. Regeneration of the library data files from backup storage.

The backup facility must provide the capability to selectively generate and restore backup data so that it is possible to recover portions of the total PSL data base without the need to perform full storage dump and restore operations.

3.1.1.d Data maintenance

Data maintenance - The PSL must provide facilities for generation and updating of library data files. Updating shall be defined to include addition to, deletion from, replacement within the copying of a file or parts thereof, as well as organization of file references to reflect such changes. Specifically, the minimum capability required includes the following functions:

3.1.1.d.1

Add one or more characters to a file without re-entering the entire record which is to contain the new characters.

3.1.1.d.2

Delete one or more characters from a file without re-entering the entire record containing those characters.

3.1.1.d.3

Replace one or more characters from a file without re-entering the entire record containing those characters.

3.1.1.d.4

Make the following changes to one or more records of source data:

- (a) Add one or more records of data.
- (b) Delete one or more records of data.
- (c) Replace one or more records of data.

3.1.1.d.5

Organize data for easy reference by providing:

- (a) Automatic generation of record sequence numbers within files for data added to the library.
- (b) Regeneration of record sequence numbers as a result of changes to data.
- (c) The capability to add records to an existing file without regeneration of record sequence numbers.

3.1.1.d.6

Copy one or more records from one library data file to another library data file:

- (a) Within a single library.
- (b) Between libraries.

3.1.2 Full Source Data Maintenance Requirements

Full Source Data Maintenance requirements include all of the Basic Source Data Maintenance requirements and the following additional requirements.

3.1.2.a Full Data File Storage

Full Data File Storage - Provision must be made for storing data (except for object and load modules) in a compressed form to minimize storage space. This requirement involves:

1. The compression of data being stored to eliminate leading and trailing blank characters in each record, as well as intervening blanks when more than three appear contiguously. This will result in storage of significant characters only.
2. Restoration of data from its compressed library storage form to its original uncompressed form.

3.1.2.b Full Data Maintenance

Full Data Maintenance - A PSL must provide for the generation and updating of library files. In addition to the basic requirements there must also be the following capabilities:

- (1) Make the following changes to a file of source data as well as to a data file:
 - (a) Modify a record of data (i.e., change one or more characters in a record without requiring the user to input the entire line).
 - (b) Scan for and replace a specific string of data in a single record or in every record in a data file in which that string appears.
 - (c) Insert a string of data into specific positions of every record of a data file.
- (2) Make temporary changes to a data file (i.e., change a data file for the duration of a single job without making permanent changes to the stored data file).
- (3) Automatically generate program stubs for subroutines and coroutines which are referenced in the source code currently in development but which themselves have not yet been coded. A program stub is defined as an incomplete routine which minimally includes entry and exit points and statements causing printout of the name of the routine entered.
- (4) Merge two or more PSL data files from two or more different PSL libraries into a single file without destroying the integrity of the source files.

3.2 Output Processing

3.2.1 Basic Output Processing Requirements

3.2.1.a Library Control Listings

Library control listings - A basic PSL implementation must also include the means to generate printed reports of a library's status. Such reports should contain the following:

1. Information related to the physical storage of files. This will include file storage space allocated and used, directory space allocated and used, type and serial number of storage device, data identification, and similar information.
2. A directory listing (i.e., an alphabetical list of file names) of the contents of a single library.

3.2.1.b Source Data Listings

Source data listings - The computer installation must be able to provide printed listings of the data stored in a PSL. Such printout shall include:

1. Listings of selected data records.
2. Listings of all data in a single library data file.

3.2.1.c Control Data

Control data - All printed listings produced by the library must contain the internal library and file names and the date and time produced for use in maintaining the external libraries.

3.2.2 Full Output Processing Requirements

Full Requirements - Full Output requirements include all of the Basic Output requirements and the following additional requirements.

3.2.2.a Full Magnetic Tape Output

Full Magnetic Tape Output - The Full implementation of PSL output processing must include the capability to copy data onto magnetic tape for permanent storage or for distribution to other computer facilities. The user must be able to determine which data is to be copied, i.e., access said data, in at least one of the following two ways:

1. Copy one or more specific data records onto magnetic tape.
2. Copy one or more specific data files onto magnetic tape.

3.2.2.b Full Punched Card Output

Punched card output - The Full implementation of PSL output processing must include the capability to output data on punched cards. The user must be able to select data for output in terms of either records or files, i.e., he must be able to use at least one of the following methods:

1. Punch the contents of one or more data records.
2. Punch the contents of one or more data files.

3.2.2.c Full Management Control Listings

Full Management Control Listings - Additional printed output requirements are defined in the Management Data Collection and Reporting requirements.

3.2.2.d Full File Directory and Listings by Programmer

Full File directory and listings by programmer - A full PSL implementation must include the capability to generate a list of all files and/or listings thereof, that are identified as having been generated by a specific programmer.

3.2.2.e Full Automatic Indentation of Source Code Listings

Full Automatic indentation of source code listings - The Full implementation of a PSL must include the capability to automatically indent source code listings generated from PSL source data files. Indentation for JOVIAL, ANS FORTRAN, ANS COBOL, and TACPOL is in accordance with the structured programming standards defined in Volume I of the Structured Programming Series, "Programming Language Standards".

3.2.2.f Full Data Scanning Capability

Full Data Scanning Capability - A Full implementation of a PSL must include the capability to scan for a specific string of data in every record of every file in a PSL and to list the names of the files in which it appears. This capability allows a user to determine where a specific string of data is used and if it can be replaced in every file in which it is used.

3.2.2.g Full Directory Availability Printout

Full Directory Availability Printout - A Full implementation of a PSL must include the capability to generate a printed report of

- a) the initial space allocation made for a directory on a directory oriented device,
- b) the date of such allocation,
- c) the number of directory records currently utilized and/or the present space utilization within the directory,
- d) the date and time of the report,
- e) the date of the last directory compressions.

3.3 Programming Language Support

This functional area involves the validation and compilation of program source code stored in the PSL. The basic requirements involve the compilation of source code. The full requirements involve the validation of source code based on structured programming standards.

3.3.1 Basic Programming Language Support Requirements

3.3.1.a Compiler Interface

Compiler Interface - The Basic implementation of a PSL must include a method for invoking precompilers, if available, and compilers in order to process source statements stored in the library. Such source statement processing must include:

1. Syntax checking of source code (i.e., precompiler and compiler checking of source code for proper use of the language syntax and for identification of all errors).
2. Compilation of source code and storage of the resulting object module in the library.

3.3.1.b Load Module Generation

Load Module Generation - The Basic implementation of a PSL must provide the interface required to execute those programs necessary for the conversion of object modules into executable code. The PSL must also provide a means of storing the resulting machine code in the library or of loading the program(s) into storage for subsequent execution.

3.3.2 Full Programming Language Support Requirements

Full Programming Language Support requirements include all of the Basic Programming Language Support requirements and the following additional requirements.

3.3.2.a Full Top Down Structured Programming Support

Full Top Down Structured Programming Support - The PSL must include facilities which provide a capability to perform, at user option, automatic exception checking of certain top down structured programming standards which are more efficiently checked during library update than during compilation. Such exception checking must include but is not limited to:

1. Flagging (i.e., identifying on a listing) of all explicit branches such as GO TO statements.
2. Flagging program language source code files or programs blocks that exceed a maximum size to be defined by the user.
3. Flagging any lines of source code that contain more than one source statement (a line of code is defined as one source file record). This requirement is based on coding standard 4 presented in Section 3.2 of Vol. I of the Structured Programming Series which states that "only one statement per line of code is permitted". The reasons for this restriction are increased program readability and suitability for pre-compiler input.

3.4 Library System Maintenance

This functional area involves the generation and maintenance of the library system and related data storage and indices. The basic requirements include a facility to install, maintain and terminate the PSL. The full requirements expand this facility to provide a system generation capability.

3.4.1 Basic Library System Maintenance Requirements

3.4.1.a Library Installation Support

Library Installation Support - A procedure and the related system support facilities must be provided to install the PSL at a user location.

3.4.1.b Library Maintenance Support

Library maintenance support - The following support of data storage maintenance must be provided.

1. Allocation of new library data storage and directory space.
2. Reallocation of existing data storage and directory space.
3. Compression of data in existing files to eliminate deleted or superceded data and recover unused space.

3.4.1.c Library Termination Support

Library termination support - The system facilities required to terminate the use of a PSL must be provided. Such termination is defined to include the deletion of all library data storage and related indices.

3.4.2 Full Library System Maintenance Requirements

Full Library System Maintenance requirements include all of the Basic Library System Maintenance requirements and the following additional requirements.

3.4.2.a Full Library Installation Support

Full Library Installation Support - The PSL installation procedures and related system support facilities must provide for the following two installation requirements.

1. Definition of the hardware configuration at installation time to meet the user's requirements (e.g., no specific hardware configuration beyond general minimum requirements).
2. A system generation facility that allows the user to generate a PSL that contains only the major functional capabilities (e.g., management data collection and reporting, etc.) required at that installation. This capability provides the flexibility to generate a PSL to meet requirements that fall in between the Basic and Full requirements as defined in this report.

3.5 Data Security

This functional area involves control over the integrity and security of the data stored in the library. The PSL must provide control over different versions of programs/systems under development, control over the inadvertent destruction of data, and the protection of data from unauthorized access. The requirements in the area of data protection provide data privacy and support the protection of classified data. However, these are minimum capabilities and are not intended to provide a completely secure system. Protection of classified data is left primarily to physical security measures.

3.5.1 Basic Data Security Requirements

The ability to recover from inadvertent loss or destruction of data through the use of a backup capability is covered in Vol. II, Paragraph 3.1.1.c.

In addition to such backup capability, the PSL must also provide, as a Basic requirement, protection against the inadvertent destruction of data as a result of an updating operation. The PSL must prevent the addition or copying of files onto a PSL library if units with the same names already exist within that library.

3.5.2 Full Data Security Requirements

In addition to the Basic Data Security requirements a PSL must support the following requirements in order to provide control over the integrity and security of the data stored in the library.

3.5.2.a Full Data Integrity

A method for maintaining control over multiple versions of a program or program system as well as a method for maintaining control over various change levels of source files within a specific version of a program system must be provided. The objective of such control is to allow a development and/or maintenance group to manipulate multiple versions of the same program system (which would be stored in separate subdivisions) while still ensuring that source code is not duplicated unnecessarily. Such control also allows data to be merged from several subdivisions in order to obtain the desired combination for the purpose of performing tests (see also Paragraph 3.9.2.a).

It is to be noted that a version is defined as a complete program or program system as it exists at a specific point in time, usually related to a certain phase of development. Any given program system might have an operational version, a development version and a maintenance version, among others. A change level within a version refers to the number of major changes that have been made to the individual source files within the version.

3.5.2.b Full Data Protection

Full Data Protection - A Full implementation of a PSL must include the protection of data from unauthorized access and unauthorized update. The data protection requirements are:

1. The PSL must provide a means to restrict access to the subdivisions within a project and the data types within a subdivision.
2. The PSL must provide a means to filter access to files of a given data type stored within a subdivision. Specifically, it must be possible to:
 - (a) Restrict access to data files for either output or update to only authorized individuals.
 - (b) Allow free access to data files for output but restrict update to only authorized individuals.
 - (c) Allow free access to data files for either output or update.
3. The PSL must provide a means of printing classified titles (i.e., Confidential, Secret, etc.) on any printed reports produced by the system.

3.6 Management Data Collection and Reporting

This function involves the collection and storage of data related to program development and maintenance and the generation of management reports containing the data and/or summaries of the data. For further details on data items collected, report classes, etc., see the Management Data Collection and Reporting report (Structured Programming Series, Volume IX).

3.6.1 Basic Management Data Collection and Reporting Requirements

The collection and reporting of management data is not a Basic requirement of a PSL.

3.6.2 Full Management Data Collection and Reporting Requirements

The functional requirements are subdivided into full collecting, full updating, full accumulating, full archiving and full reporting requirements. All management data collected by the PSL will enter the PSL through either the collecting or updating facilities.

3.6.2.a Full Collecting

Full Collecting - The data collection facility is an automatic function of the PSL. This facility provides

- (a) for the gathering of Actual data, i.e., data describing the results of programming actions within the scope of a project. Such data is gathered when programming related data as defined in Vol. II, para. 3.1.1.a is entered into the PSL.
- (b) for the storage of Actual data in a Management Statistical Data Storage area.

This facility must have the following capabilities:

1. Counting input source code lines, i.e., records, and storing or modifying already existing Actual data stored in the PSL. Such data may include version/modification level, net lines of source code per file and total lines of source code input per file.
2. Gathering and storing in a Management Statistical Data Base all pertinent source file data (e.g., start date and end date) which is available to the PSL.

3. Counting and storing Actual data concerning certain PSL functions. Typical data would be the number of compilations, number of assemblies, number of lines of code from another source, etc.
4. Allowing the user to add routines to the PSL which may be used for gathering of Actual data needed to satisfy unique user requirements.

3.6.2.b Full Updating

Full Updating - Provide a data update facility to add, delete, or replace Plan or Actual data stored in a Management Statistical Data Base. This facility must provide the following capabilities.

1. Allocating Management Statistical Data Storage area(s) on direct access devices and initializing the PSL for the data collection and reporting functions for a new structured programming project. These storage area(s) are reserved for data at both the hierarchical level (file, program, subsystem and system) and computer job level.
2. Editing Management Statistical Data and notifying the user of format or data errors.
3. Adding Management Statistical data, supplied by the user, to a Management Statistical Data Base in the PSL. This data will include such items as source file name, user identification, programming language, specification omission errors, implementation misinterpretation errors, and man-months or hours of programming time.
4. Deleting Management Statistical Data from a Management Statistical Data Base. Such deletion will be based on user-supplied information.
5. Replacing Management Statistical Data existing in a Management Statistical Data Base. Such replacement will be based on user-supplied information such as deliverable and indicator, number of lines of source code from another source, and man-months or hours of programmer time.
6. Storing user-supplied data relating to computer job turnaround time. This data will be used to compute the average, maximum and minimum computer turnaround time for a job.

7. Terminating the collection of data for specific data types (e.g., Computer Utilization). For clarification of data types see Section 3.5.1 of Volume IX of the Structured Programming Series.
8. Terminating the data collection and archival functions performed by the PSL for a programming project.

3.6.2.c Full Accumulating

Accumulating - Provide a data accumulation facility to extract and summarize or process Management Statistical Data stored in the PSL. This facility must provide the following capabilities:

1. Summarizing Management Statistical Data for the hierarchical levels of program, subsystem and system, and for all related source files existing in the PSL. The summarized data will include items such as total program source lines and total program files.
2. Processing Management Statistical Data which exists in the PSL. Such processing consists of determination of maxima, minima, and averages for certain items and may include the calculation of average source unit size, average number of source lines input per file, and maximum number of updates allowed for a file.
3. Retaining the summarized and processed data for the purposes of reporting and archiving.

3.6.2.d Full Archiving

Full Archiving - Provide an historical information storing facility to record historical information on structured programming projects. This facility must provide the following capabilities:

1. Storing on a secondary storage device all data items which have been collected or accumulated at the system level. This data should be retained for a user-determined duration in order to satisfy historical reporting requirements.
2. Backing up of Management Statistical Data. Backup capabilities were described in Vol. II, Subsection 3.1.1.

3.6.2.e Full Reporting

Full Reporting - Provide a report generation facility to output information in a convenient and meaningful manner. This facility must provide the following capabilities:

1. Producing Program Module Statistics reports for the various hierarchical levels including file, program, subsystem and system. These reports will contain such information as net lines of source code, total program source files, total update runs performed in a program and total lines of source code input for a program during a given reporting cycle.
2. Producing detailed and summary Computer Utilization reports. The detailed reports will contain information such as the average computer turnaround time for each programmer. The summary report will contain information such as the average computer turnaround time for all programmers on a project.
3. Producing Program Maintenance Statistics reports for the various hierarchical levels including file, program, subsystem and system. These reports will contain information such as specification omission errors, implementation misinterpretation errors, and number of update runs.
4. Producing Program Structure reports in both detailed and summary form. The detailed reports will list the total tree structure of the program starting at the file specified by the user and including all "included" and "called" source files. The detailed reports will also contain a cross reference listing of all included and called files. The summary reports will contain a partial tree structure showing only the top user-specified number of levels. They will also contain a cross-reference listing.
5. Producing Historical reports from the archival projects data. The content of these reports will be user-indicated, and may include such data as project name, date range, and range of project size expressed in number of program source files. The reports will also contain cost data and key historical data items from the Program Module Statistics, Computer Utilization, Program Maintenance, Statistics and Program Structure reports for both active and inactive software projects.

6. Producing Combination reports for the various hierarchical levels including program subsystem and system. These reports will combine cost data with information from more than one report class (e.g., Program Module Statistics and Program Maintenance Statistics).
7. Producing Cyclic versions of the reports described in Items 1 through 6 above. These reports will include data such as total lines of source code for a program for a given program cycle. They will also flag discrepancies, e.g., when travel costs expended to date exceed the travel costs budgeted to date.
8. Allowing the user to add output routines to the PSL. These routines will be used to output the actual data collected by the user routines which were added in order to satisfy unique user requirements.
9. Producing reports on an exception basis, without user request, when significant variances exist between Actual and Plan data.

3.7 Documentation Support

This functional area involves the storage, update and output of program documentation. The requirements defined in this subsection are based on the use of a program design language, structured code and text as the principal sources of program documentation. Support of the generation, maintenance and output of graphics (e.g., block diagrams, HIPO charts, etc.) are not included in these requirements. The intent is to use the PSL storage and maintenance capabilities along with some specialized output capabilities to provide a basic documentation capability. It is not the intent of these requirements to use the PSL as a sophisticated document generation and publication system.

3.7.1 Basic Documentation Support Requirements

The storage and update of program design language, program source code and textual data are described in Subsection 3.1 of this report. The requirements defined in Subsection 3.1 are sufficient to cover all of the requirements for storing and updating program documentation. Additional documentation support is not included in the PSL basic requirements.

3.7.2 Full Documentation Support Requirements

3.7.2.a

Full Documentation Support implementation includes the capability of printing in a user specified sequence one or more PSL files. This output facility must be able to merge Program Design Language files, source code files and textual data files into a single output listing.

3.7.2.b

Full requirements include the capability to format an output listing with the following user-specified information.

1. Page beginning indicators.
2. Header information to be printed at the top of each page.
3. Spacing between files being printed.
4. Spacing between lines of output.
5. Number of lines to print on a page.

3.7.2.c

Complete formatting facilities must include automatic page numbering beginning with a user-supplied page number or with page 1.

3.7.2.d

Full Documentation Support requires the capability to print a title page which contains as a minimum the following user-specified information:

1. Document Title.
2. Document Date.
3. Name(s) of Author(s)
4. Organization and Address.

3.7.2.e

Full Documentation Support includes the capability to generate a magnetic tape in print image form. This allows the distribution of documentation in machine readable form, thus facilitating both storage and reproduction of documentation.

3.8 On-Line Terminal Implementation Requirements

Up to this point the descriptions of the PSL functional requirements have been independent of any implementation considerations. A system to satisfy these requirements could be implemented in either a batch or an on-line environment. However, the use of on-line terminals for the dynamic execution of normal library functions (e.g., source data maintenance, program compilation, etc.) introduces some additional requirements which apply only to an on-line implementation of a PSL. These requirements, which cover general on-line terminal capabilities, are stated in this subsection.

An on-line implementation of a PSL must provide the following capabilities to support the functional requirements defined in the other subsections of this report.

3.8.1 Basic On-Line Terminal Implementation Requirements

3.8.1.a

On-line implementation of a PSL must include support of terminals which are connected to the computer either via communication links or directly via cables and control units (if required).

3.8.1.b

Interactive PSL capabilities must include support of both display (Cathode Ray Tube) terminals and type-writer terminals.

3.8.1.c

A remote job entry capability must be provided so that PSL jobs initiated at an on-line terminal may be run in a batch environment.

3.8.1.d

On-line implementation of a PSL requires a terminal-to-terminal communication capability to allow the transfer of data between interactive ports within the same PSL system.

3.8.1.e

A facility to recover from system failure must be included in the PSL so that data which has been entered from a terminal prior to a hardware or software failure is not lost (i.e., a user may reinitialize a terminal session at the point of failure so that only the data displayed on the terminal but not entered into the system is lost).

3.8.1.f

An on-line PSL implementation requires the capability to scan through data files displayed at the terminal. This may be termed a paging capability for those terminals which are page-oriented, such as CRTs, but even those terminals which are not page-oriented should be able to display the first, last, and specified intermediate lines of a file. The total file scanning capability includes (where applicable):

1. Displaying the next page of data. A page of data is defined as the number of lines which may be displayed on the CRT at a single time; this number is dependent upon the specific make of CRT.
2. Displaying the previous page of data.
3. Displaying the first page of data retrieved.
4. Displaying the last page of data retrieved.
5. Rolling a display forward one or more lines.
6. Rolling a display backward one or more lines.

3.8.2 Full On-Line Terminal Implementation Requirements

3.8.2.a

Full interactive PSL implementations requires the capability to collect and output statistics related to each on-line terminal. Such statistics include time in use (real and CPU), types and number of requests, etc.

3.8.2.b

Full on-line implementation requires the capability to restrict the access to specific PSL data files to designated on-line terminals. I.e., access to certain files is allowed only from specific on-line terminals which may be located in physically secure or restricted areas.

3.8.2.c

Some form of user identification and authentication is required for access to a PSL from any on-line terminal.

3.8.2.d

Full on-line implementation requires the existence of a user support facility which allows a PSL user to receive at his terminal a summary of PSL functions and commands. The HELP file allows the on-line user to learn proper utilization of the PSL.

3.9 General Requirements

General support provided by a PSL includes several capabilities which are not directly associated with one of the other major functional areas.

3.9.1 Basic General Requirements

There are no general support requirements included in the PSL basic requirements.

3.9.2 Full General Requirements

A Full PSL must support the following general requirements.

3.9.2.a

A Full PSL implementation requires that the user be able to specify more than one file in all jobs requiring access to a data file. Said multiple files must be concatenated in user-specified order and treated as though they were one contiguous file. This capability allows the user to combine different versions of files (e.g., development and operational) for the purpose of testing new capabilities, error corrections, etc.

3.9.2.b Full Subroutine Support for User Programs

Subroutine support for user programs - Full PSL implementation requires that programs and/or subroutines used to perform certain PSL functions be made available to the user. Such programs include index searches, file listings, and other basic tasks which may be needed for integration into software written by the user for his own special needs.

*MISSION
of
Rome Air Development Center*

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C³I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.