ESD-TR-70-278

USER'S MANUAL
JOVIAL COMPILER VALIDATION SYSTEM

July 1970

DIRECTORATE OF SYSTEMS DESIGN & DEVELOPMENT
HQ ELECTRONIC SYSTEMS DIVISION (AFSC)
L.G. Hanscom Field, Bedford, Massachusetts 01730

LEGAL NOTICE

When U. S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

ESD-TR-70-278

USER'S MANUAL
JOVIAL COMPILER VALIDATION SYSTEM

July 1970

DIRECTORATE OF SYSTEMS DESIGN & DEVELOPMENT
HQ ELECTRONIC SYSTEMS DIVISION (AFSC)
L.G. Hanscom Field, Bedford, Massachusetts 01730

FOREWORD

The JOVIAL Compiler Validation System (JCVS) Users Manual is intended as the reference manual for on-site operations.

The system was developed as a part of Project 6917 under Contract F19628-68-C-0301 for the Electronic Systems Division (AFSC) by Data Dynamics, Inc., Los Angeles, California 90045. The project monitor was Captain Martin J. Richter, ESMDA. The work was performed during the period March 1968 through February 1969.

This technical report has been reviewed and is approved.

William F. Heisler

WILLIAM F. HEISLER, Colonel, USAF
Director, Systems Design & Development
Deputy for Command and Management Systems

ABSTRACT

This technical report consists of detailed specifications for the use of the JOVIAL Compiler Validation System (JCVS). The system is designed to measure the compliance of a specific JOVIAL J3 compiler against the language specifications in Air Force Manual 100-24, "Standard Computer Programming Language for Air Force Command and Control Systems". This report describes the card input formats, deck structures, tape requirements, test modules, and operator procedures required to use the system.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

SECTION I

INTRODUCTION

The purpose of this manual is to :

1. Introduce the JOVIAL Compiler Validation System
2. Describe how the system may be used to validate JOVIAL Compilers

This manual is organized into five sections. Following Section I, the Introduction, Section II describes the JOVIAL Compiler Validation System. Included in this Section is a brief discussion of the current JOVIAL J3 Standard, the AFM 100-24 document, some insight into the design criteria which guided the development of the system and a discussion of the functions performed by components of the system. Section III suggests how the JCVS may be used as a package to validate JOVIAL compilers. Section IV presents the details of each of the system components. Sufficient material will be included in this section to completely describe the uses of each component in the system. In addition, details of programs and their relationship to data are fully described.

Although the AFM 100-24 document defines specific input/output statements for the JOVIAL J3 language, discussions with implementors of this language have established that of the existing JOVIAL compilers none have adhered to these input/output specifications. Most current JOVIAL compilers use either the input/output capabilities provided by the operating system in which JOVIAL is embedded or an associated ancillary system within the software environment. There is currently little control over the form of the JOVIAL associated input/output statements. In addition only the GE-635 JOVIAL Users Manual is currently available. These two facts when taken together present considerable difficulty to those JOVIAL support statements that concern themselves with printing the results of the execution and comparison of JOVIAL test statements. Until a firming of the input/output specifications to the JOVIAL language has been established, this fact is a major obstacle to the successful usage of this system.

Section V will discuss the JOVIAL Compiler Validation System as it applies to the five computers upon which the system will reside. Because of the absence of information relating the JOVIAL compiler to its operating system, the requirements relating the two will be discussed in general terms only. This section will describe how the JCVS must be used by defining input deck structures and tape mountings, providing the required instructions to operate the system, and giving examples of the results obtained from the various modules comprising the JCVS.

SECTION II

SYSTEM DESCRIPTION

The JOVIAL Compiler Validation System (JCVS) is designed to evaluate the extent af campliance of any JOVIAL compiler with the current JOVIAL Standard Computer Programming Language for Air Force Command and Control Systems Manual, AFM 100-24.

Depending an the extensiveness and depth of testing, the user may either select a representative collection of test statements or the camplete test repertaire. If the user is interested in a particular capability as provided by the JOVIAL campiler, he may desire to execute test statements exclusively in the area of that particular capability.

Having decided upan the particular collection of test statements to be executed, the user specifies his intent ta the JCVS by means af test selector cards. These cards are interpreted by the JCVS and are used to select the desired test statements to be included in the generated test program. The resulting JOVIAL test program will be produced for compilation in card image form on magnetic tape or on cards.

2.1 The JOVIAL Standard

The JOVIAL J3 language is completely specified in AFM 100-24, Standard Camputer Pragramming Language far Air Force Command and Cantral Systems, 15 June 1967. The JOVIAL language has the basic elements required by most languages, namely, the ability to define simple data items and basic item structures and the capability to reference this data fram within procedural statements. The procedural statement repetoire is adequate, consisting af the following procedure types:

1. Data Transmission
2. Algebraic Expression Formulation
3. Logical Expression Farmulation
4. Transfers af Pragram Cantral
   - 4.1 Canditional
   - 4.2 Unconditional
   - 4.3 Switching
   - 4.4 Looping
5. Input/Output

There are other adds and ends in the language that are useful but computer dependent and serve ta confound the intent of this specification, namely, standardizatian.

Another section of this manual is devoted to establishing standards for the development of compilers of JOVIAL J3. Elements of this standard are, on occasion, ignored by the implementors of the language. This is particularly true in the case of the input/output specifications provided by the language. These specifications are rudimentary in character and are, generally, replaced by comprehensive (but non-standard) input/output procedures more closely associated to the operating system within which the compiler is embedded. Unless a more stringent attitude toward the development of JOVIAL compilers is maintained it is impossible to write JOVIAL input/output statements with the conviction that they will be compatable from one computer-compiler configuration to another.

For purposes of this system, the entire JOVIAL language will be treated as a single module. Because of the size and pointedness of the language, no submodularization will be required.

2.2 JCVS Testing Concepts

The following sections discuss briefly the scope of the JCVS and the tests selected for inclusion in the Population File.

2.2.1 JCVS Scope

For purposes of the JCVS, the JOVIAL system to be tested is assumed to consist of a processor that compiles standard JOVIAL source program statements called the JOVIAL compiler, and all programs and subroutines used by the JOVIAL object code generated from standard JOVIAL statements. The JCVS is designed to test both the compilation and execution of specific JOVIAL features.

2.2.2 Data Concepts

JOVIAL language organization has guided the identification of language features to be tested. In order to validate the JOVIAL compiler ideally, each of the specific language features must be validated. The validation of each feature of a language, however, is not always possible. For example, how can one determine that any value stored in a floating point item is truly stored as a floating point number; how can one determine that a fixed point constant has actually been converted to a fixed point binary point constant. Looking at information as it resides in the internal storage medium, we may observe a string of bits, however, the interpretation of this content is inconclusive. Consequently, some of the features provided by the JOVIAL language are not susceptible to validation independently. These features are generally the more basic notions in the language and will be used constantly in the Test Modules comprising the Population File. With repeated correct usage of these basic concepts, it is hoped that the credibility of their required implementation will be considerably improved.

With these thoughts in mind, the following aspects of the data definitional capabilities of the JOVIAL language will not be tested independently and will be assumed present in the language and correctly implemented:

1. The ability to specify any item type and have it retained according to its defining attributes.
2. The ability to formulate any constant type and have it retained according to its defining attributes.
3. The ability to specify any data structure type (table, array, etc.) and have it retained according to its defining attributes.

The JOVIAL language provides the user with a myriad of options to form constants, simple items, tables, and arrays. There are so many data defining attributes possible in JOVIAL that exercising each option in an independent test is quite impossible. As a compromise, the test repertoire will use a subset of data definitions that exercise, at least once, all of the data attributes available to define data items and structures. In addition, the repertoire will utilize every variation provided to formulate constants with the exception of the dual item definitions which will be exercised in part, only. It goes without saying that the formation of acceptable JOVIAL symbols (names, labels, etc.) will be exercised every time a symbol is formed.

2.2.3 Procedural Concepts

The JOVIAL language provides the user with the ability to process formulas and relations; it provides for program organization and it provides certain compiler directing features. Every variant of each of these features will be tested at least once. Further substantiation of the ability of a feature to perform its intended function will be supplied by its correct use as a support statement in other test modules.

With these thoughts in mind, the following aspects of the procedural capabilities of the JOVIAL language will be assumed to be present in the language and correctly implemented:

1. The ability to name a statement with a label.
2. The fact that normal procedural control passes from one JOVIAL statement to the next.

Comprehensiveness

The variants provided in the data base form a nucleus from which tests may be created. Selected data statement variants and all procedure statement variants will be included in the data base. Selected values for variant operands will also be a part of the data base. Since the collection of values comprising the complete range for each variant operand may be extremely large, only a representative number of values for each operand may be included. These factors, of course, indicate that individual variants may be tested only for a subset of their possible operand values.

This subset of operands will be large enough, however, to associate a large degree of confidence with the evaluation of each variant.

A JOVIAL compiler is said to be validated if each individual data base variant with its appropriate subset of operand values has been executed and results compared successfully. The collection of variants and operands on the data base necessary to validate the compiler will be referred to as the "nominal" data base. The JOVIAL source test program that may be used to validate the entire JOVIAL compiler is called the nominal "test case".

The design reflects the following:

a) A careful sampling of selected operands from possible combinations of operand types available to the statement.
b) No tests are made of erroneous statements.
c) All possible variants of procedural statements are performed.
d) Tests are not designed to indicate how a function is implemented. Thus, there is no attempt to distinguish between efficient and inefficient implementations.
e) No testing of non-standard extensions to JOVIAL is made. However, such tests and extensions can be added to the system by the user through the add, change and delete option cards in the Population File program.
f) No test of direct code is attempted.

Openendedness

Modification to the data base may become necessary as changes are made in the JOVIAL Standard. Variants and operand values may be added to the data base to test user-specific extentions to the JOVIAL language. Variants and operand values may be deleted or modified because of reinterpretations of existing JOVIAL language features. The JCVS will provide the means to add, change or delete any data base variants and operand values in the Population File.

Ease of Use

Complete and detailed input and test configurations facilitate ease of use. In Section 4 the input cards to each program are described in detail. Each input card is defined, card columns are specified, and all mandatory cards are so designated. In Section 5, the order of all the cards from each program needed for a JCVS run is graphically portrayed. The collection of test statements provided by the JCVS is shown in Appendix 6 together with their individual test serial numbers. The test serial number permits the user to select, eliminate or add specific tests.

Additional features that make the JCVS easy to use are:

a) A test can be specified by a user without detailed knowledge of JOVIAL.
b) Test Results which show discrepancies are output. An option exists for viewing an indication of the results of all tests (see Section 3.5).
c) Program modules are machine independent.

2.3 JCVS Computer Program System Capabilities

The JCVS consists of a collection of three major program modules and a data base that provides the user with a simple technique to generate a JOVIAL source program capable of testing some particular aspect of the compiler or the entire compiler itself. The data base, called the Population File, contains all of the test statements that are potential candidates for inclusion in subsequent generated source programs. A particular test may be created including specific functions and excluding those functions that are not provided, for one reason or another, by the particular compiler. A comprehensive test package may be developed by the user for each compiler.

The Population File is maintained by the Population File Maintenance Module. Population File test modules are added, deleted or replaced by means of this routine. The Selector Module extracts user-specified test modules from the Population File, distributes the necessary operating system control cards and support statements and generates a self contained JOVIAL source program for subsequent processing. The Source Program Maintenance Module may be used to update a generated JOVIAL source program.

2.3.1 The JCVS Data Base

The Population File contains the following types of information:

1. Environmental - Hardware/Software
2. Test Modules
3. Identification

This information is presented to the Population File on cards whose descriptions are given in Section 4.

2.3.1.1 Environmental - Hardware/Software

Environmental data, both hardware and software, for all computers of interest is carried in the Population File. Hardware specific information such as printer control codes, magnetic tape designations and memory size and software specific information such as operating system control card descriptions and computer-compiler specific JOVIAL control card descriptions offset by one column are carried in the first few records of the Population File.

2.3.1.2 Test Modules

A Test Module is a collection of JOVIAL statements that test a particular feature of the JOVIAL compiler. The feature may be a JOVIAL concept, a single JOVIAL statement or a collection of JOVIAL statements. Included in each Test Module are the:

1. Test identification field
2. Input test data fields
3. Test Results fields
4. Expected Result fields
5. Initialization procedures
6. Test statements comprising the test
7. Results analysis procedures
8. Output procedures

Test Modules are located on the Population File in order of their test serial number, the DDI-NO. With each test statement is associated a sequence number within the DDI-NO that specifies the ordering of the statements within the DDI-NO.

2.3.1.3 Identification

The first 80 characters of the Test Module are devoted to information describing several aspects of the test. These 80 characters, called the Test Module Header, contain the name of the test, its test serial number, any CED AFM 100-24 numbers associated with the test, and any required references to other test modules.

2.3.2 Population File Maintenance Module

This module operates on a Population File and permits the user to add, delete, replace, or change logical records on the Population File. This feature is the means by which the user updates the Population File with current information. Environmental, test, and identification information may be augmented by means of this module.

This module will be used to modify the contents of the Population File to incorporate new tests resulting from extensions to the JOVIAL compiler, to delete current tests when particular aspects of the compiler have not been implemented, or to include information describing the environment in which the JOVIAL tests will be conducted.

2.3.3 The Selector Module

The Selector Module performs the major task of assembling and organizing test and support statements for the JOVIAL test program.

1) Using the input specifications obtained from the user, appropriate variants and operand values may be selected.

2) The resulting test and support statements are placed in the order needed for compilation.
3) Operating system control cards are placed before and after the JOVIAL source test program.

2.3.4 Source Program Maintenance Module

This program is used to modify a JOVIAL source program either generated by the Selector Module or previously modified by the Source Program Maintenance Module. This module may be used if:

1) One or more tests did not compile correctly (therefore deletions of erroneous statements or changes to existing statements can be made).
2) The user wished to change a test in order to compare with a previous run using different user defined operand values (parametric study).
3) The user wishes to add non-standard tests to the JOVIAL source program.

2.3.5 The Test Program

The JOVIAL source program generated by the Selector Module is a self contained JOVIAL J3 program in compilable form. The test structure and content of the particular source program has been completely specified by the user. All statements supporting the test are provided automatically by the JCVS.

Each test within the source program exercises one or more of the features provided by the JOVIAL compiler by actually compiling and executing those JOVIAL statements that provide the feature. The results of this procedure stating the outcome of this execution may be displayed.

It was originally intended to display expected versus actual results. Lack of adequate capabilities of the input/output portions of the JOVIAL language, coupled with an inability to acquire input/output information about the JOVIAL implementations themselves, reduces the comparison printout to a message stating whether the test has passed or failed together with an identification of the associated DDI-NO.

Test results printed under these constraints do not fully reveal the causes of errors in tests devoted to the accuracy of arithmetic operations. The results of syntax-semantic testing, however, are not affected by this constraint.

SECTION III

SYSTEM USAGE

3.1 Hypothetical JCVS Operational Philosophy

DDI hypothesized that the JCVS may be utilized operationally in any of several ways:

1. The entire system, including the Population File, can be distributed to JOVIAL J3 implementors for use in validating their JOVIAL implementation.
2. JOVIAL source programs may be developed by a central agency and the source programs sent to JOVIAL implementors for compilation and execution. Results of these runs could be returned for processing by the same agency.
3. A team of personnel could accompany the JCVS to a specified computer upon which the JOVIAL compiler is to be exercised. The JCVS is then made operational on the computer system and the particular JOVIAL compiler is tested.

Any of the above operational philosophies could be followed, however, based upon the work statement description of the problem, the third philosophy appears to be the most probable approach.

If operational philosophy one or three is followed, all the program modules will be required to execute on the computer for which the JOVIAL compiler has been prepared.

If philosophy two is followed, the Population File Maintenance Module and the Selector Module will be processed on a single computer (possibly not one of the target computers in this contract) while the Source Program Maintenance Module will be processed, at a minimum, on the computer upon which the JOVIAL compiler has been implemented.

For the remainder of this document we shall assume that philosophy three is to be followed and all JCVS modules must be operational on each computer containing the JOVIAL compiler to be tested.

3.2 System Initiation

For each specified computer a Population File and three source decks will be provided. Each of the source decks must be compiled and a resultant binary deck of each program

module obtained. All JCVS program modules will have been written in a subset of COBOL to ensure that the program will, after changes to the input/output characteristics of each program module and appropriate control cards, compile into a useable program.

Once the Population File Maintenance Module has been established on one of the target computers, the Population File may be developed for this computer. Since certain aspects of the JOVIAL language may be specified by the implementor there may be idiosyncracies of the JOVIAL implementation that could necessitate modifications to the JOVIAL test statements or the JOVIAL test statement formats. It is impossible at this time to predict what form these idiosyncracies might take; consequently, the user must be aware of this situation and be capable of adjusting the test statements, if required, to conform to the specific compiler.

A notable example of this problem occurs because the reference format as specified by the AFM 100-24 document indicates that a JOVIAL source program statement may occupy any of the 80 columns on a card. Specific implementors, in general, do not permit this free field interpretation and specify margins within which a JOVIAL statement must be written.

Once the program modules have been compiled and the Population File has been created, the user may proceed to the next step, the generation of a test program.

3.3 Test Program Generation

The selection of tests necessary to validate a JOVIAL compiler may vary widely depending upon the testing philosophy.

More than likely, the particular compiler features to be tested depend entirely on the uses to which the compiler will be exposed and the environment in which the compiler will reside. The JCVS user, presumably knowing this, will have produced specifications to which the compiler must adhere. In order to ensure that the compiler does, indeed, adhere to these specifications, the user selects from the Population File those tests that exercise those features whose correct execution will result in a verification of the stated specifications.

A second approach to the validation of the compiler might consist of selecting for testing all of the features stated as standard by the AFM 100-24 document. Using this approach would give the user a "look see" at what features were implemented.

Having chosen the tests to be processed, the user submits this information to the Selector Module by means of Test Selector Cards. The Selector Module Program deck must be augmented by the operating system control cards for the particular computer upon which the Selector Module is being run.

The exact job deck structure for each computer required to achieve a Selector Module run is given in Appendix 1.

There are occasions when the generated JOVIAL source program will exceed the limitations of either the compiler or the hardware environment. In order to remedy compiler violations, consult the JOVIAL compiler users manual to establish the cause of the trouble.

In order to remedy excessive core storage requirements, segment the generated JOVIAL source program by selecting several smaller programs rather than one large program.

3.4 Test Program Execution

The JOVIAL test program resulting from the Selector Module run is then compiled. If the program compiles with error, these errors should be recorded by the user. By means of the cross referencing mechanism provided with each test, DDI-NO versus CED-NO's, all references to the test may be located in the AFM 100-24 document.

The Source Program Maintenance module may then be used to eliminate from the source program those elements causing the compilation errors. The compilation and element removal process is continued until an error-free compilation has been achieved.

Following a successful compilation, the object program is executed. If the execution terminates abnormally, a study of the partial results obtained by the run will be required to locate the offending test elements. If the execution terminates normally, a glance at the results of the test will provide information signifying individual feature compliance with AFM 100-24 standard.

3.5 Test Result Evaluation

The notion of what constitutes a validated JOVIAL compiler is a function of the requirements to be levied on the compiler. Consequently, the user, based upon the compilation and execution of one or more test programs, must formulate his decision with the information gathered as a result of these test runs.

Within each generated source program there may be tests of two types: Those that test the various syntax-semantics relationships present in the language and those that test the accuracy of arithmetic computations provided by the algebraic expression capabilities of the language.

The syntax-semantics tests are logical in character and can be answered by monitoring the semantic response the compiler provides for a syntactic type.

For example, a reasonable test for the GOTO statement could consist of: Does it go where it says it is going to go? The result is either yes or no. If yes is the case, an

appropriate message is printed out and if not is the case, another message results. As a general rule, the results of logical tests may be indicated by a yes or no decision only.

The tests for accuracy, on the other hand, require that computed results be compared with expected results; that both results, if possible, be converted and printed together with a decision stating that the feature either passed or failed to pass its accuracy requirements.

Accuracy tests, in general, depend upon the ability of the compiler-computer configuration to represent and process correctly numbers exercising the extreme capabilities of the hardware. Given that these operations have been performed correctly (in binary) the problem of converting these numbers to printable form (decimal) requires the application of some JOVIAL output procedures. Since no standard JOVIAL formatting conversion and output procedure exists machine language or other higher level language coding must be utilized in order to view the results. This foreign conversion process, however, can introduce non JOVIAL compiler computational errors into the computed results and render the accuracy considerations of the tests useless.

In the absence of input/output specifications for four of the five JOVIAL compilers in question, only the statement indicating that the feature has passed or failed its test will be printed. When the JOVIAL language provides proper formatting capabilities the ability to display computed and expected results may be added to the output sections of the test modules.

SECTION IV

FUNCTION DESCRIPTION

4.1 Population File Maintenance Module (POPFM)

4.1.1 Purpose and Uses

The POPFM module may be used to generate a new Population File either by initiating the file from cards or by updating an old Population File with current additions to the information contained in the old file. This information consists of Environmental Data or Test Statements. These information types are organized into 4000 character physical records for recording on magnetic tape. Each physical record consists of either a System Module or a Test Module.

The modules are stored in numerically ascending sequence by serial number, the DDI-NO, associated with each of the modules in order to facilitate the processing to be applied to the Population File. This processing permits addition, deletion, or replacement of user specified information to this file.

All physical records are treated identically and the updating functions provided by the JCVS regard only the items (DDI-NO, CARD-TYPE and SEQ-NO) to control the updating process.

Input to the POPFM consists of a current file of information called the Current File-PF, an optionally present old Population File and a control card requesting specific options provided by the module. The Current File-PF is a card file, while the old Population File and the new Population File are magnetic tape files.

Output from this routine consists of an updated Population File, an Audit File-PF consisting of diagnostics, trace messages with an optional listing of all of the card images on the Population File containing information, and an optional Punch File consisting of a card deck of all card images on the new Population File containing information.

4.1.2 Preparation of Inputs

4.1.2.1 System Module

Each computer-compiler configuration will contain environmental data describing specific aspects of the hardware environment in which the JCVS will reside.

Information identifying the hardware configuration, the facility, the user, etc., will be supplied to the Population File by means of System Header cards. Environmental hardware information such as printer codes, magnetic tape designations, etc., will be supplied to the Population File by means of Environmental Hardware cards.

The aforementioned information will be carried as descriptive material only and will not participate in the generation or will not become a part of any of the generated JOVIAL source programs.

On the other hand, the environmental-software information supplied to the Population File by means of Environmental Software cards will become an integral part of the generated JOVIAL program. This software information consists of operating system control cards and the JOVIAL START and TERM cards. Some of these cards precede and others follow the generated program.

4.1.2.1.1 System Header Card 1

System Header Card 1 occupies the first 80 character positions in the System Module (the first 4000 character physical record on the Population File).

Columns	Name	Description
1-12	Users Name	These 12 columns may be used to identify the agency or organization using the JCVS. The name may be positioned any place within the field, (Example: bbUSAF-ESDbb, or USAF-ESDbbbb.)
13-24	Facility	These 12 columns may be used to identify the facility at which the JCVS is being utilized. The name may be positioned any place within the field. (Example: bbHANSCOMbbb, or bHANSCOM AFB.)
25-34	Computer-Name	These 10 columns may be used to identify the computer manufacturer and machine serial number. The name may be positioned any place within the field. (Example: bCDC-6600b, or GE-635 bbbb.)

Columns	Name	Description
35-45	Data of Basic File Creation	These 11 columns may be used to identify the date on which the basic farm of the Population File has been created. The month, day, and year are specified YYYYbMM MbDD. (Example: MAYb12b1968, or SEPb13b1967).
46-47	Modification Number	These 2 columns may be used ta identify the number of times that the basic file has been modified. (Example: Secand modificatian 02, tenth modificatian 10).
48-58	Date of Creation of this File	These 11 columns may be used to identify the date that this file was created. The manth, day, and year are specified YYYYbMMM bDD. (Example: DECb17b1968, ar MAYb25b1968).
59-72	Not Used	These 14 columns are not used.
73-76	DDI-NO	These 4 columns contain the test serial number, the DDI-NO, 0001.
77	Card Type	This column contains the character A that indicates that this card is a nan-test statement card.
78-80	Sequence Number	These 3 columns contain 001 indicating that this card accupies the first 80 columns in the System Module.

4.1.2.1.2 System Header Card 2

System Header Card 2 occupies the second set of 80 character positians in the System Module and contains the following information:

Columns	Name	Description
1-35	Validation System Name	These 35 columns may be used to identify the particular modification af the validation system. (Example: bbbbbbbbbbbJCVSbMAYb 1968bbbbbbbbbbb, or JOVIALbCOMPILERbVALIDATIONb SYSTEMb5).

Columns	Name	Description
36-72	Operating System Name	These 37 columns are used to identify the operating system within which the JCVS is imbedded. (Example: bbbbIBM-360bDISKb OPERATINGbSYSTEMbbbb).
73-76	DDI-NO	These 4 columns contain the test serial number, the DDI-NO, 0001.
77	Card Type	This column contains the character A that indicates this card is a non-test statement card.
78-80	Sequence Number	These 3 columns contain 002 indicating that this card occupies the second 80 columns in the System Module.

See Appendix 2 for complete description of all of the System Header Card 2 card types used on the five computer-compiler configurations.

4.1.2.1.3 Environmental Hardware Card 1

Environmental Hardware Card 1 occupies the third set of 80 character positions in the System Module and contains the following information:

Columns	Name	Description
1-30	System Input	These 30 columns contain the acceptable hardware name for the system input unit.
31-60	System Output	These 30 columns contain the acceptable hardware name for the system output unit.
61	Space Code	This column contains the printer single space code.
62	Double Space Code	This column contains the printer double space code.
63	Page Eject	This column contains the printer page eject code.
64-70	Memory Size	These 7 columns contain the core memory size.
71-72	Not Used	These 2 columns are not used.
73-76	DDI-NO	These 4 columns contain a DDI-NO = 0001.

Columns	Name	Description
77	Card Type	This column contains the character A that indicates this card is a non-test statement card.
78-80	Sequence Number	These 3 columns contain a sequence number = 003.

4.1.2.1.4 Environmental Hardware Card 2

This card occupies the fourth set of 80 character positions in the System Module and contains the following information:

Columns	Name	Description
1-30	System Punch	These 30 columns contain the acceptable hardware name for the system punch unit.
31-60	Scratch 1	These 30 columns contain the acceptable name for a scratch unit 1.
61-72	Not Used	
73-76	DDI-NO	These 4 columns contain a DDI-NO = 0001.
77	Card Type	This column contains the character A that indicates that this card is a non-test statement card.
78-80	Sequence Number	These 3 columns contain a sequence number = 004.

4.1.2.1.5 Environmental Hardware Card 3

This card occupies the fifth set of 80 characters in the System Module and contains the following information:

Columns	Name	Description
1-30	Scratch 2	These 30 columns contain the acceptable hardware name for tape scratch unit 2.
31-60	Scratch 3	These 30 columns contain the acceptable hardware name for tape scratch unit 3.
61-72	Not Used	

Columns	Name	Description
73-76	DDI-NO	These 4 columns contain the DDI-NO = 0001.
77	Card Type	This column contains the character A that indicates this card is a non-test statement card.
78-80	Sequence Number	These 3 columns contain a sequence number = 005.

See Appendix 3 for a complete description of the Environmental Hardware Cards for the five computer configurations.

4.1.2.1.6 Environmental Software Cards

These cards provide specific operating system control cards that may be used to specify the functions to be performed by the operating system and the JOVIAL START and TERM cards that bracket the JOVIAL source program. These cards have SEQ-NO's greater than 005 and are stored in the System Module shifted right one column.

Columns	Name	Description
1	Not Used	
2-72	Environmental Software Statement	These 71 columns provided contain a request of the operating system to perform a specific task.
73-76	DDI-NO	These 4 columns contain the test serial number, the DDI-NO, 0001.
77	Card Type	This column contains either the character L that indicates this card precedes the JOVIAL source program or the character F that indicates this card follows the JOVIAL source program.
78-80	Sequence Number	These 3 columns may contain the digits 005 through 050 which serves to indicate the relative position of this card in the System Module.

A current list of the Environmental Software cards excepting the JOVIAL START and TERM cards (GE-635 only) used by the JCVS is given for each computer configuration in Appendix 4.

4.1.2.2 Test Modules

A test module is a collection of JOVIAL statements that test a particular feature of the JOVIAL compiler. The feature to be tested may be a JOVIAL concept, a single JOVIAL

statement or a collection of JOVIAL statements. Included in each test module are the:

1. Test identification field
2. Input test data fields
3. Test result fields
4. Expected result fields
5. Initialization procedures
6. Test statements comprising the test
7. Results analysis procedures
8. Output formatting procedures

The tests are carried in the Population File in order of ascending DDI-NO. Within each DDI-NO the test header and the JOVIAL test statement cards are carried in order by ascending Sequence Number. The DDI-NO identifies each test module to all of the JCVS program modules and the user. Population File test modules may be assigned a four digit DDI-NO between 0500 and 9997.

Each Test Module begins with a Test Header Card that contains the DDI-NO, the Sequence Number, the test name, one or more references to the associated paragraphs in the AFM 100-24, and, if required, a number called the Mandatory DDI-NO of a module called the Mandatory Module upon which the current module depends. Additional JOVIAL comment cards may be included anywhere in the Test Module. See Appendix 5 for samples of these cards in the Typical Test Module.

The Mandatory Module could contain data or support statements required by the dependent module and, hence, must be present in any JOVIAL source program including the dependent module; or the Mandatory Module could contain another feature test whose validity must be established before a successful execution of the dependent module feature test may be considered valid. See Appendix 5 for some typical Population File modules.

4.1.2.2.1 Test Header Card

The Test Header Card occupies the first 80 characters in a JOVIAL Test Module record in the Population File and contains information about the test statements that follow.

Columns	Name	Description
1-2	Open Quotes	These 2 columns contain quote marks.
3-22	Test Name	These 20 columns describe what feature the JOVIAL statements test. (Example: THREEbFACTORbFOR bbbb, or GOTObSTATEMENTbbbbbb).

Columns	Name	Description
23-27	CED-NO1	These 5 columns identify a reference in the AFM 100-24 to the feature being tested.
28	Not Used	
29-33	CED-NO2	These 5 columns identify a reference in the AFM 100-24 to the feature being tested.
34	Not Used	
35-39	CED-NO3	These 5 columns identify a reference in the AFM 100-24 to the feature being tested.
40	Not Used	
41-45	CED-NO4	These 5 columns identify a reference in the AFM 100-24 to the feature being tested.
46	Not Used	
47-51	CED-NO5	These 5 columns identify a reference in the AFM 100-24 to the feature being tested.
52	Not Used	
53-57	CED-NO6	These 5 columns identify a reference in the AFM 100-24 to the feature being tested.
58	Not Used	
59-62	Mandatory DDI-NO	These 4 columns identify the DDI-NO of a Mandatory Module upon which the current test module depends.
63-64	Close Quotes	These 2 columns contain quote marks.
65-72	Not Used	
73-76	DDI-NO	These 4 columns contain the test serial number, the DDI-NO. (Example: 4500, 7500, 1410).
77	Card Type	This column contains either the character A or B or C that indicates this card is a non-test statement car
78-80	Sequence Number	These 3 columns contain 001, indicating that this card occupies the first 80 columns of the Test Module on the Population File.

4.1.2.2.2 JOVIAL Statement Card

The JOVIAL Statement Card contains one or more JOVIAL statements to be used in a generated JOVIAL source program. Only the first seventy-two card columns may be used for the statement. Columns 73-80 will be used for card identification.

Columns	Name	Description
1-72	JOVIAL Statements	These 72 columns may contain one or more JOVIAL statements.
73-76	DDI-NO	These 4 columns contain the test serial number, the DDI-NO. (Example: 2100, 4500, 7600).
77	Card Type	This column contains the character J that indicates this card is a test statement card.
78-80	Sequence Number	These three columns contain a number, 002-050, specifying the position of the JOVIAL Test Statement Card within the Test Module. (Example: 015 indicates that this card occupied the 15th 80 column position in the Test Module.)

4.1.2.3 Packet Cards

4.1.2.3.1 Control Card - PF

The various options permitted by the Population File Maintenance Module may be requested by means of the following control card:

Columns	Name	Description
1	Control Card Indicator	This column must contain the character C denoting the card as a control card.
2-4	Control Card Indentifier	These 3 columns may be assigned any 3 digits by the user to identify the control card.
5	Mode Designator	This column is used to signify the run type C = CREATE run U = UPDATE run

Columns	Name	Description
6	Print Option	This column may be used to request the printing of the new Population File on the Audit File-PF non-space - Print space - Do not print
7	Punch Option	This column may be used to request the punching of the new Population File. non-space - Punch space - Do not punch
8-80	Not Used	

When submitting this card to the Population File Maintenance Module the Control Card - PF directly precedes the card deck comprising the Current File - PF.

4.1.2.3.2 Delete Card

The Delete card is used to signal the Population File Maintenance Module to eliminate a record or a specific card from the Population File.

The form of the Delete card follows:

Columns	Field Size	Description
1-72	72	Not Used
73-76	4	DDI-NO
77	1	Update Function = D
78-80	3	Sequence Number

When this card is used to delete a module from the Population File, it must be included in the Current File - PF with the DDI-NO equal to the DDI-NO of the record to be eliminated from the Population File and the Sequence Number equal to 000. When this card is used to delete a card image from a record in the Population File it must be included in the Current File - PF with the Sequence Number and DDI-NO equal to the corresponding Sequence Number and DDI-NO of the card image to be eliminated from the Population File.

4.1.2.4 Input Files

The Population File Maintenance Module operates upon two input files, an optionally present Population File and the Current File - PF.

4.1.2.4.1 Population File

The Population File is organized into equal size logical records. Each logical record is composed of 4000 characters and consequently can accomodate fifty 80-column cards. Each logical record is recorded on one physical record.

The first few records on the Population File are System Modules and each contains all of the environmental and indicative information pertinent to various hardware configuration operating systems and JOVIAL compilers. A System Module may be assigned any DDI-NO between 0001 and 0499.

The remainder of the records (excepting modules 9998 and 9999) contain the individual test modules. The first eighty characters af the madule are cal led the Test Module Header and contain information pertinent to the specific test module. Column 77 of the Test Madule Header contains either the characters A, B, or C. The character B present in a Test Module Header indicates the module is an extension of the previous module and the two physical test modules act as a collection of physical modules. The character A or C present in a Test Module Header indicates the module is the beginning of a new physical module or a collection of physical modules. The character C present in a Test Module Header indicates the physical module or collection of physical modules is a mandatory module that must be present in every generated source program. Figure 4-1 gives a physical layout of the Population File.

4.1.2.4.2 Current File-PF

The Current File-PF, which directs the Population File Maintenance Module to update the Population File consists of card packets containing environmental, test, indicative or functional information. Environmental information (e.g., hardware configuration descriptions, operating system control cards, etc.) is presented by means of the Environmental Packets, test information (e.g., JOVIAL test statements) by means of the Test Packets and functional information (the Population File Maintenance Module update command, delete) by the Delete Packets. Indicative information (e.g., DDI-NO, Sequence Number, etc.) is included where required in all packets. A test serial number, the DDI-NO, is assigned to each packet and each card within the packet contains this number in columns 73-76. In addition, ordering the cards within each packet is controlled by the Sequence Number in columns 78-80. The Environmental Packet cansists of the following cards in the arder specified:

	Number of Cards
1) System Header Card 1	1
2) System Header Card 2	1
3) Environmental Hardware Cards	3
4) Environmental Software Cards	M

The total number of Current File-PF cards in one packet acceptable to the Populatian File cannat exceed 50; consequently, M, the number of Environmental Software Cards, must be less than or equal 45.

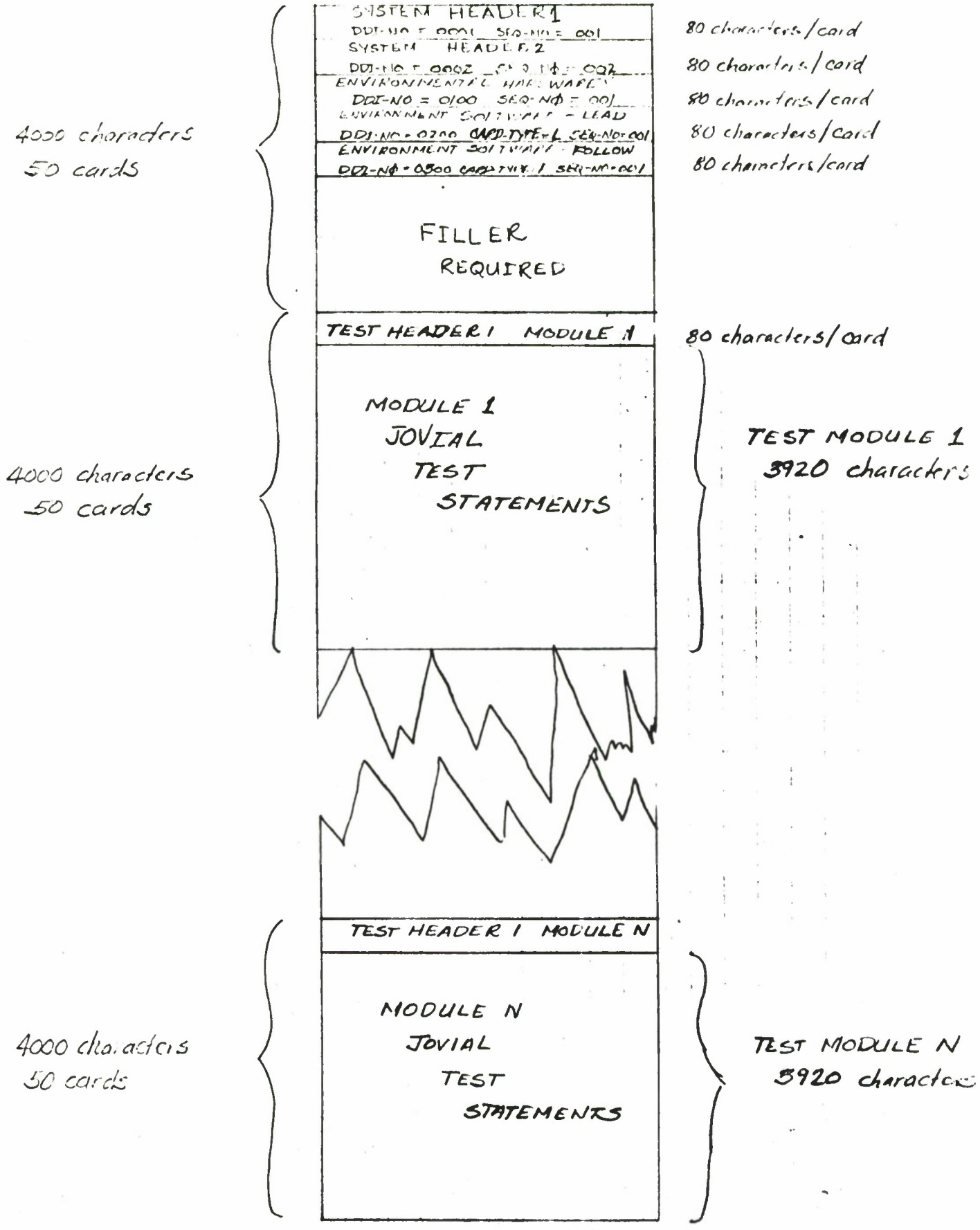


Figure 4-1 Population File

The Test Packet consists of the following cards:

		Number of Cards
1)	Test Header Card	1
2)	JOVIAL Statement Cards	N_1
3)	DELETE Cards	N_2

The total number of cards in one packet acceptable to the Population File cannot exceed 50; consequently, $N_1 + N_2$, the number of JOVIAL Statement Cards plus DELETE cards may not exceed 49.

The DELETE Packet consists of one card, the DELETE Card.

The Current File - PF consists of a collection of the above mentioned packets in order of Sequence Number within DDI-NO. Only those cards that are to effect elements in the old Population File need to be included in the Current File - PF.

4.1.3 Function Operation

The Population File Maintenance Module operates either to initiate a Population File completely from the Current File-PF or to update an existing Population File by means of information residing on the Current File - PF. In each case, the control card permits the user to specify options to print and/or to punch the resulting new Population File.

4.1.3.1 Create Population File

When the Population File Maintenance Module is used to initiate a Population File the Current File - PF may contain only information to be added to the file. Consequently, no DELETE cards are permitted in the Test Packets that comprise the Current File-PF.

The packets are placed in order by DDI-NO to form the Current File - PF. The Mode Designator in the control card is set to C and the appropriate print/punch options are selected. The control card precedes the Current File - PF when submitted to the Population File Maintenance Module.

Since no old Population File is required for this run, all the test modules in the Current File - PF utilize the update function ADD and are ADDed to form the Population File.

4.1.3.2 Update Population File

When the Population File Mai ntenance Module is used to update an existing Population File, DELETE cards may be present in the packets comprising Current File-PF. Each Current File - PF packet is composed of a collection of cards, each card invoking an update function which performs one of the following operations:

1) Add a card to a new or an existing test module
2) Replace a card on an existing test module
3) Delete one or more existing test modules
4) Delete a card from an existing test module

The update functions are controlled on the basis of two items included in every card in the Population File:

1) DDI-NO (columns 73-76)
2) Sequence Number (columns 78-80)

The packets in the Current File-PF may contain no more than 50 cards and must be in order within the packet by Sequence Number within DDI-NO. The Sequence Numbers, however, need not be consecutive.

In order to reduce the card preparation requirements of the system, the ADD feature and REPLACE feature are invoked automatically. Specifically the update functions adhere to the following rules:

1. ADD

   If an ADD (a card to be ADDed to the Population File) card is included in a packet on the Current File-PF and no card with the same DDI-NO (columns 73-76) and Sequence Number (columns 78-80) is present in the old Population File, the card in the Current File-PF is automatically added to the Population File in its proper sequence.

2. REPLACE

   If a REPLACE card (a card intended to REPLACE another card on the Population File) is included in a packet on the Current File-PF and a card with the same DDI-NO (columns 73-76) and Sequence Number (columns 78-80) is present in the old Population File, the card in the Current File-PF automatically replaces the corresponding card on the new Population File.

3. DELETE

   The DELETE option is invoked by means of a DELETE packet included in the Current File-PF. This packet may instruct that either an entire record or a card within a record not be recorded on the new Population File. If the Sequence Number on the DELETE packet is 000 and the DDI-NO matches a DDI-NO in the old Population File the entire record and any suceeding records with B in column 77 of the Test Module Header are not recorded on the new Population File.

   If the Sequence Number on the DELETE packet is a number between 001 and 050 and the DDI-NO and Sequence Number match a DDI-NO and Sequence Number

in the old Population File, the matched card is not recorded on the new Population File. If a match is not effected, a diagnostic is printed.

Consequently, a packet in the Current File - PF may contain ADD, REPLACE, and DELETE functions applicable to a specific record on the old Population File. When card images on the old Population File are to be altered, only the cards that are to provide the changes need be included in the Current File - PF packets.

On the other hand, an entire record may be deleted by the inclusion in the Current File - PF of the appropriate DELETE packet.

The Population File Maintenance Module only changes those card images on records in the existing Population File that have been specified by the user.

The packets are placed in order by Sequence Number within DDI-NO to form the Current File - PF. The Mode Designator in the control card is set to U and the appropriate print/punch options are selected. The control card precedes the Current File - PF when submitted to the Population File Maintenance Module.

4.1.4 Description of Expected Results

4.1.4.1 Output Card Formats

The output card formats correspond to the formats for cards as described in Section 4.1.2.

4.1.4.2 Output Files

The Population File Maintenance Module produces three output files, the Population File, the Audit File - PF, and the Punch File - PF.

4.1.4.2.1 Population File

The results of either a CREATE or an UPDATE run will always produce a new Population File which is completely described in Section 4.1.2.

4.1.4.2.2 Audit File - PF

The Audit File - PF contains a listing of all diagnostics and trace messages originating from this module. As an optional feature, the user may request to print on the Audit File - PF a working listing of the card images on the new Population File by selecting the print option on the Control Card - PF. Since the Audit File - PF is only a working listing, diagnostic and

tracing information will be interspersed with the Population File card images on the Audit File - PF.

Following is a list of the diagnostic messages to be printed in the Audit File-PF together with their explanations:

Diagnostic Message	Explanation
NO UPDATE FUNCTION CARD	There is no control card preceding the Current File-PF.
RECORD TO BE DELETED NOT ON OLD MASTER FILE	The Current File-PF contains a DELETE packet referencing a DDI-NO not on the old Population File.
CURRENT FILE CARDS ARE OUT OF SEQUENCE	The cards in the Current File-PF are not in sequence by DDI-NO.
INITIAL RUN CARD NOT PRESENT	The control card preceding the Current File-PF contains an incorrect Mode Designator.
OVERFLOW MASTER RECORD BUFFER	The Current File-PF contains a card whose sequence number is greater than 50.

Following is a list of the trace messages to be printed on the Audit File-PF.

The following messages are all paragraph names printed from within each named paragraph:

1) IUC
2) UPDATE CONTROL
3) OLD MASTER FILE READOUT
4) END OF CURRENT FILE
5) END OF OLD MASTER FILE
6) END OF OLD MASTER FILE 4

The following typical trace message is printed whenever the WRITE-ERROR paragraph is entered:

LAST CARD KEY	0002A005
LAST CURRENT FILE KEY	0002A003
LAST OLD MASTER FILE KEY	0005A004

The information opposite the LAST CARD KEY represents the control field (columns 73-80) of the last Current File-PF card read.

The information opposite the LAST CURRENT FILE KEY represents the control field of the next to lost Current File-PF cord reod.

The information opposite LAST OLD MASTER FILE KEY represents the control field of the first card image in the lost physical record reod from the old Populotion File.

This troce information is printed on one line in the Audit File-PF.

4.1.4.2.3 Punch File - PF

Yet another option, the punch option, may be selected by the user to obtoin o cord deck of all cord images on the Population File containing information.

4.2 Selector Module (SJCVS)

4.2.1 Purposes and Uses

The Selector Module performs the major tosk of assembling ond organizing test and support structures for the JOVIAL test progrom.

1. Using the input specificotions obtained from the user, oppropriote test and support structures may be selected.
2. The resulting test and support structures are ploced in the order needed for compilation.
3. Environmentol Softwore cords ore placed before and after the JOVIAL source test program.

Input to the Selector Module consists of the Populotion File, the Test Selection File, (o collection of user specified cards which control the identity of the tests selected from the Populotion Fi le) ond o control cord requesting the specific options provided by the module.

Output of the Selector Module includes a Source Program File consisting of the generated JOVIAL Source program, the Audit File-S consisting of o diagnostics, trace message with on optional listing of the Source Progrom File and an optional Punch File-S consisting of o card deck of the Source Program File.

4.2.2 Preparation of Inputs

4.2.2.1 Input Card Formots

Following is a description of the cord types and formats input to the Selector Module.

4.2.2.1.1 Test Selector Card

The Test Selector Card permits the user to specify the selection of one or more test modules from the Population File. The user specifies the DDI-NO identifying the first test module to be selected, the increment to be added to the DDI-NO identifying the first test module, and the DDI-NO identifying the last test module to be selected. If only one test module is to be selected at a time, the increment may be set to 0000 or left blank. The following describes the format of the Test Selector Card:

Columns	Name	Description
1-4	Control Word	These 4 columns must contain the control word TEST.
5-10	Not Used	
11-14	Starting DDI-NO	These 4 columns contain the DDI-NO identifying the first Population File Test Module to be selected by this Test Selector Card.
15-20	Not Used	
21-24	Increment	These 4 columns contain the value to be added to the starting DDI-NO and succeeding DDI-NO's until the final DDI-NO has been selected.
25-30	Not Used	
31-34	Final DDI-NO	These 4 columns contain the DDI-NO identifying the last Population File Test Module to be selected by this Test Selector card.
35-80	Not Used	

4.2.2.1.2 Control Card-S

The various options permitted by the **Selector** Module may be requested by means of the following control card:

Columns	Name	Description
1	Control Card Indicator	This column must contain the character C denoting the card as a control card.

Columns	Name	Description
2-4	Control Card Identifier	These 3 columns may be assigned any 3 digits by the user to identify the control card.
5-6	Margin A	These 2 columns are used to designate the column number of Margin A on the Source Program File card images.
7-8	Margin B	These 2 columns are used to designate the column number of Margin B on the Source Program File card images.
9	Print Option	This column may be used to request the printing of the Source Program File on the Audit File-PF. non-space - Print space - Do not print
10	Punch Option	This column may be used to request the punching of the Source Program File. non-space - Punch space - Do not punch
11-14	System Module DDI-NO	The DDI-NO of the appropriate System Module to be selected from the Population File.
15-80	Not Used	

4.2.2.2 Input Files

The Selector Module operates upon two input files: the Population File and the Test Selection File.

4.2.2.2.1 Population File

The Population File has been thoroughly described in Section 4.1.2.

4.2.2.2.2 Test Selection File

The Test Selection File consists of a collection of Test Selector cards that direct the generation of a JOVIAL source program. One or more tests may be selected by means of a Test Selector card. The collection of Test Selector cards may be submitted to the Selector Module in any order.

4.2.3 Function Operation

The Selector Module, under the direction of the Test Selection File, operates on the Population File to produce a single JOVIAL source program consisting of 80 column card images from one or more JOVIAL test modules residing on the Population File.

The Test Selection File controls the identity of the Population File test modules that are recorded on the Source Program File. For example, suppose the Test Selection File consisted of the following Test Selector card information with no Mandatory DDI-NO's involved.

Card Number	Starting DDI-NO	Increment	Final DDI-NO
1	4100	0010	4200
2	3000	0005	3010
3	6000	0000	
4	8100	0001	8105

The Source Program File would consist of the following sequence of selected test modules as identified by their associated DDI-NO's:

3000, 3005, 3010, 4100, 4110, 4120, 4130, 4140, 4150, 4160, 4170, 4180, 4190, 4200, 6000, 8100, 8101, 8102, 8103, 8104, 8105

Notice that the test modules selected as indicated by the list of DDI-NO's are not in the same order as they appear on the Test Selection File, but are in ascending order by DDI-NO, the same order that they appear on the Population File. All mandatory and environmental software cards supporting the generated test, and modules 9998 and 9999, are automatically selected or generated by the Selector Module.

In the following example, suppose the Test Selection File consisted of the following Test Selector Card information:

Card Number	Starting DDI-NO	Increment	Final DDI-NO
1	4100	0010	4120
2	3000	0005	3010
3	6000	0000	

Suppose further that the Mandatory DDI-NO's associated with each of the above DDI-NO's are given in the following list:

DDI-NO	Mandatory DDI-NO
4100	2500
4110	----
4120	1200
3000	2215
3005	2210
3010	2210
6000	4000

Suppose also that the Test Module headers for modules 2000 and 8000 have C's in column 77 and that the Test Module headers for modules 2216, 4101, 4102, and 8001 have B's in column 77. Assuming this, when the Test Selection File is submitted to the Selector Module, the following test modules will be selected and placed on the Source Program File in the following order:

Test Module	DDI-NO	Test Module	DDI-NO
1	1200	10	4000
2	2000	11	4100
3	2210	12	4101
4	2215	13	4102
5	2216	14	4110
6	2500	15	4120
7	3000	16	6000
8	3005	17	8000
9	3010	18	8001

Mandatory test modules will be supplied only once in the output of the Selector Module. Notice that again the test modules are placed on the Source Program File in order of ascending DDI-NO. In addition the mandatory modules supporting the generated test, modules 0001, 9998 and 9999 are selected or generated by the Selector Module. All modules with a C in column 77 are automatically selected by the Selector Module. Modules with a B in column 77 should not be selected by the user.

4.2.4 Description of Expected Results

4.2.4.1 Output Card Formats

Following is a description of the card types and formats output by the Selector Module.

4.2.4.1.1 Environmental Software Card

These cards provide communication between the generated JOVIAL source program and the operating system of the particular computer. These cards both precede and follow the JOVIAL source program and are operating system specific. For a description of the operating system cards for the five computers used by the JCVS see Appendix 4. For a complete description of this card see Section 4.1.2.1.6.

4.2.4.1.2 Test Header Card

These cards are placed in the JOVIAL source program as comment cards. They serve to identify the test and provide cross referencing information between the DDI-NO and associated AFM 100-24 references, the CED-NO's. A complete description of this card is given in Section 4.1.2.2.1.

4.2.4.1.3 JOVIAL Source Program Card

The JOVIAL Source Program Card contains one or more JOVIAL statements to be used in a generated JOVIAL source program. As with most cards associated with the JCVS columns 73-80 will be used for card identification.

Columns 1-72, however, will be subdivided into a maximum of three sections as indicated in the diagram.

Margins A and B specify card columns selected by the user between which is contained as much of the content of a JOVIAL Statement Card as permitted by the margin specifications. Card column 1 from the JOVIAL Statement Card is transferred to the card column specified by Margin A in the JOVIAL Source Program Card; Column 2 is transferred to column Margin A+1, etc. If column k is transferred to Margin B, columns k+1 through 72 of the JOVIAL Statement Card are not transferred and, hence, lost. These margin specification features are provided to the user because of the lack of standardization of JOVIAL J3 reference formats.

The two margins must adhere to the following inequality:

$$\text{column } 1 \leq \text{Margin A} < \text{Margin B} \leq \text{column } 72$$

If no Margins are specified, Margin A will nominally be set to 1 and Margin B to 72. Notice that the character string signifying the JOVIAL statement must be short enough to fit between the margin. Specifically the character string must adhere to the following inequality:

$$\text{Length of character string} \quad \text{Margin B} - \text{Margin A} + 1$$

The form of the JOVIAL Source Program Card follows.

Columns	Name	Description
1-Margin A	Not Used	
Margin A - Margin B	JOVIAL Statement	These (Margin B - Margin A) columns contai n one or more JOVIAL statements.
Margin B-72	Not Used	
73-76	DDI-NO	These 4 columns contain either the DDI-NO (e.g., 2100, 4500, 7610) or the number 9999.
77	Card Type	This column contains the character J that indicates this card is a test statement card.
78-80	Sequence Number	These 3 columns contain a number, 002-051 specifying the position of the JOVIAL Source Program Card within the card images from the selected Test Module.

4.2.4.2 Output Files

The Selector Module produces three output files: The Source Program File, the Audit File-S, and a Punch File-S.

4.2.4.2.1 Source Program File

The Source Program File contains the JOVIAL source program. The generated source program consists of, in part, JOVIAL statement card images from Test Modules in the Population File. Preceding and following the source program are operating system cards that form the linkage between the JOVIAL source program and the operating system. In addition, every test present in the Source Program File may be identified by the Test Header card preceding the JOVIAL test statements comprising the test.

The Source Program File is recorded one output card image per physical record. Since the Source Program File is in the same order as the Population File, by Sequence Number within DDI-NO, the DDI-NO and Sequence Number act as the control items for this file.

Since the environmental software cards that follow the generated JOVIAL source program originate from the System Module; these cards would normally have a DDI-NO equal to 0001 in the Source Program File. As a result, these cards would be out of order in a generated JOVIAL source program. In order to alleviate this situation, all trailing environmental software cards are automatically assigned a DDI-NO = 9999. Sequence numbers in these cards, however, remain unchanged.

The START card will contain the DDI-NO of the selected System Module and the same Sequence Number it possessed in the System Module. The TERM card is assigned the DDI-NO = 9999 but contains the same Sequence Number it possessed in the System Module.

Figure 4-2 gives a physical layout of the Source Program File.

4.2.4.2.2 Audit File-S

The Audit File-S contains a listing of all diagnostics and trace messages emanating from this module. As an optional feature, the user may request to print on the Audit File-S, a working listing of the card images on the new Source Program File by selecting the print option on the Selector control card. Since the Audit File-S is only a working listing, diagnostic and tracing information will be interspersed with Source Program File card images on this file.

Following is a list of the diagnostic messages to be printed on the Audit File.

Diagnostic	Explanation
EXCEEDED DDI-NO TABLE	There exists on the Population File a DDI-NO greater than 9998. Check the Population File for cause of error.
DDI-NO AND INDEX NOT SYNCHRONIZED	In processing the Population File the DDI-NO on the current Population File record is less than the DDI-TABLE index. Probable cause: Machine malfunction.
UNEXPECTED EOF INFILE	An unexpected end of file has been triggered on INFILE. Check the Control Card-S and the Test Selection File for cards that could cause the end of file and restart the progr
UNEXPECTED EOF POP-FILE	An unexpected end of file has been encountered on the Population File Check to see if the Population File has been rewound properly and restart program. This diagnostic is probably triggered by a machine error.
NO CONTROL CARD	There is no Control Card-S or an incorrect Control Card-S present in the INFILE. Supply the correct Control Card-S and restart.

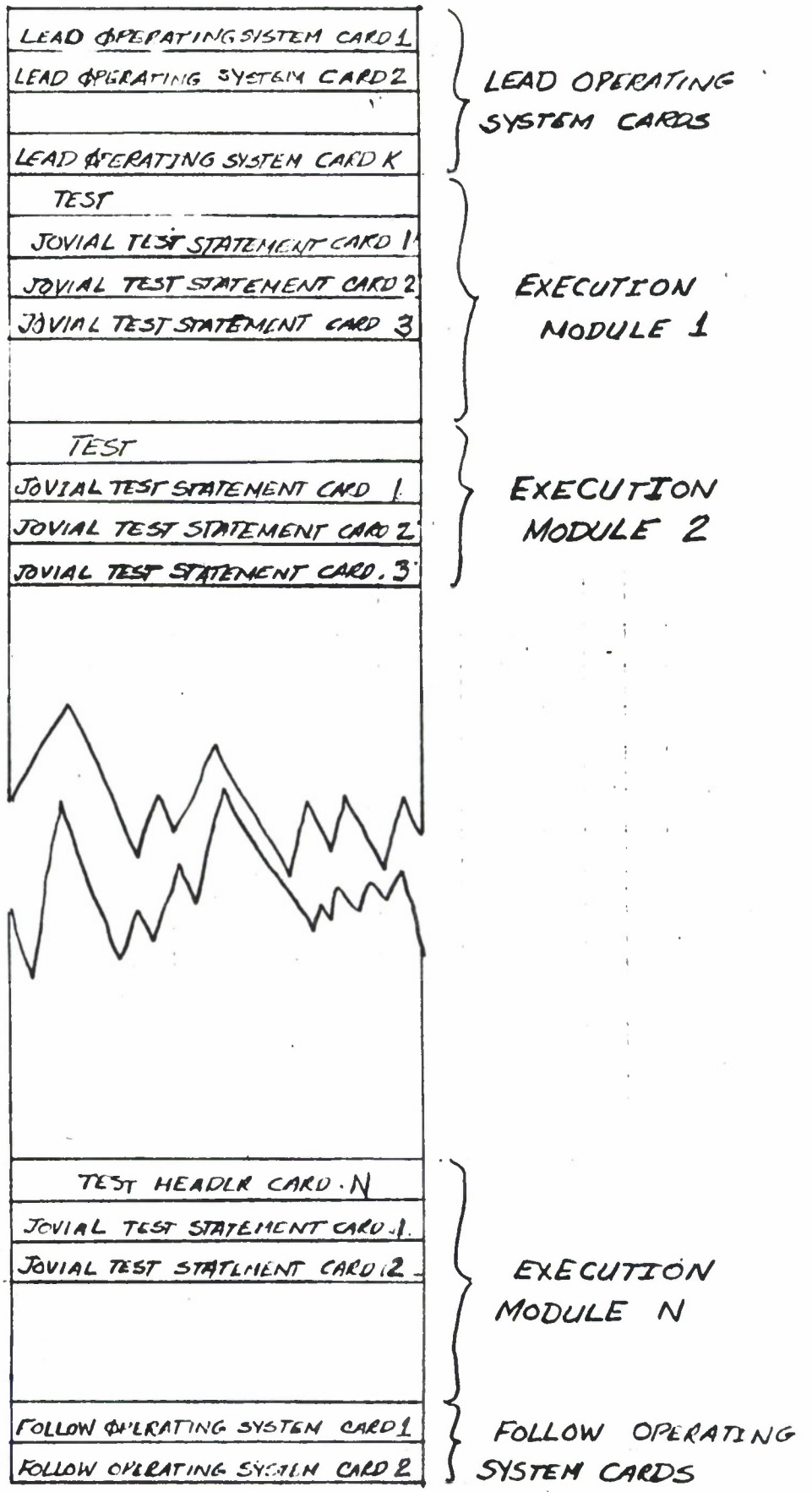


Figure 4-2. Source Program File

Diagnostic	Explanation
INCORRECT TEST SELECTOR CARD	There is an incorrect Test Selector Card in the INFILE. Correct the card and restart the program.
INCORRECT CONTROL CARD	The Control Card-S margin specifications are incorrect. Correct specifications and restart program.

Following is a list of trace messages to be printed on the Audit File-S.

The following trace messages are all paragraph names printed from within the named paragraphs:

1. BDT1
2. BUILD-SPF

The following trace messages are values that monitor the contents of key items together with the paragraph names printed from within the named paragraphs.

Message	Originating Paragraph
1. Contents of item DDI-NUMBER	BDT2
2. BMT1, Contents of item DUMP	BMT1
3. Contents of record CARD	ERR-PROC-6

4.2.4.2.3 Punch File-S

Yet another option, the punch option, may be selected by the user to obtain a card deck of all card images on the Source Program File.

4.3 Source Program Maintenance Module (SOPMM)

The Source Program Maintenance Module is used to modify either the JOVIAL source program generated by the Selector Module or a JOVIAL source program previously modified by SOPMM. Modifications may be necessary because:

1) One or more tests did not compile correctly; therefore, deletions of erroneous statements or changes to existing statements can be made.
2) The user wishes to change a test in order to compare with a previous run.
3) The user may wish to add self contained non standard tests.
4) Certain areas of the JOVIAL compiler have not been debugged completely.
5) The user may wish to eliminate partially implemented features.

Input to the Source Program Maintenance Module consists of a Source Program File, the Current File-SP, and a control card requesting specific options provided by this module.

Output from the Source Program Maintenance Module includes an updated Source Program File consisting of the modified JOVIAL source program, the Audit File-SP consisting of diagnostics, trace messages with an optional listing of the Source Program File and an optional Punch File-SP consisting of a card deck of the updated Source Program File.

4.3.1 Preparation of Inputs

4.3.1.1 Card Inputs

4.3.1.1.1 Control Card-SP

The various options permitted by the Source Program Maintenance Module may be requested by means of the following control card:

Columns	Name	Description
1	Control Card Indicator	This column must contain the character C denoting the card as a control card.
2-4	Control Card Identifier	These 3 columns may be assigned any 3 digits by the user to identify the control card.
5	Mode Designator	This column is only referenced descriptively and does not influence the run type which is always an UPDATE run. It should be set to U for documentary purposes.
6	Print Option	This column may be used to request the printing of the new Source Program File non-space - Print space - Do not print
7	Punch Option	This column may be used to request the punching of the new Source Program File. non-space - Punch space - Do not punch

Columns	Name	Description
8	Trace Option	This column may be used to request printing on the Audit File-SP of all the trace messages originating in this module. non-space - Print messages space - Do not print messages
9-80	Not Used	

When submitting this card to the Source Program Maintenance Module the Control Card-SP directly precedes the card deck comprising the Current File-SP.

4.3.1.1.2 Other Card Inputs

A complete description of all other card forms contained in either the Source Program File or the Current File-SP is given in Sections 4.1.2.3.2 and 4.2.4.

4.3.1.2 Input Files

4.3.1.2.1 Source Program File

The Source Program File has been completely described in Section 4.2.4.2.1.

4.3.1.2.2 Current File-SP

The Current File-SP which directs the Source Program Maintenance Program to update the Source Program File is composed of individual cards that provide the capability to add, delete, and replace information on the Source Program File.

The following card types may appear in the Current File-SP:

1. Environmental Software Card
2. Test Header Card
3. JOVIAL Source Program Card
4. DELETE Card

The information content of the aforementioned cards has been completely specified in Sections 4.1.2 and 4.2.4.1.3.

A serial number, the DDI-NO, is present in columns 73-76 of each card in this file, and a Sequence Number in columns 78-80. The cards in this file are placed in order by Sequence Number within DDI-NO.

4.3.2 Function Operation

The Source Program Maintenance Module operates on an existing Source Program File directed by a Current File-SP to update and generate a new Source Program File. The control card associated with this program permits the user to specify options to print and/or punch the resulting new Source Program File.

The Source Program Maintenance Module provides the user with the ability to add information to the Source Program File, delete information from the Source Program File or replace information on the Source Program File on a card image by card image basis. The Current File-SP consists of individual cards ordered by DDI-NO and Sequence Number. Each card invokes an update function implicitly or explicitly. The cards within the Current File-SP permit the user to change any card in the Source Program File. The control card precedes the Current File-SP when submitted to the Source Program Maintenance Module.

Each card in the Current File-SP specifies an update function which performs one of the following operations:

1. ADD a card to the Source Program File
2. REPLACE a card on the Source Program File
3. DELETE one or more test modules from the Source Program File
4. DELETE an entire test module from the Source Program File

The update functions are controlled on the basis of two items included in every card in the Source Program File.

1. DDI-NO (columns 73-76)
2. Sequence Number (columns 78-80)

In order to reduce the card preparation requirements of the system, the ADD feature and REPLACE feature are invoked automatically. Specifically, the update functions adhere to the following rules.

ADD

If an ADD card (a card to be ADDed to the Source Program File) is included in the Current File-SP and no card with the same DDI-NO (columns 73-76) and a Sequence Number (columns 78-80) is present in the old Source Program File, the card on the Current File-SP is automatically added to the Source Program File in its proper sequence.

REPLACE

If a REPLACE card (a card intended to REPLACE another card in the Source Program File) is included in a packet on the Current File-SP and a card with the same DDI-NO (columns 73-76) and Sequence Number (columns 78-80) is present on the old Source Program File, the card on the Current File-SP automatically replaces the corresponding card on the new Source Program File.

DELETE

The DELETE option is invoked by means of a DELETE card included in the Current File-SP. This card causes a card with the same DDI-NO and Sequence Number not to be recorded on the new Source Program File. If the Sequence Number equals 000, the entire module specified by the DDI-NO and any directly succeeding modules with a B in column 77 in the Test Module Header are deleted.

4.3.3 Description of Expected Results

4.3.3.1 Output Card Formats

A complete description of all of the card forms contained in either the Source Program File or the Punch File-S is given in Section 4.1.2 and 4.2.4.1.3.

4.3.3.2 Output Files

The Source Program Maintenance Module produces three output files: The Source Program File, the Audit File-SP, and a Punch File-SP.

4.3.3.2.1 Source Program File

The Source Program File has been completely described in Section 4.2.4.2.1.

4.3.3.2.2 Audit File-SP

The Audit File-SP as an optional feature may contain a listing of all diagnostics and trace messages originating from this module. A second optional feature the user may request is to print on the Audit File-SP a working listing of the card images on the new Source Program File by selecting the print option on the Control Card-SP. Since the Audit File-SP is only a working listing, diagnostic and tracing information will be interspersed with the Source Program File card images on the Audit File-SP.

Following is a list of the diagnostic messages to be printed in the Audit File-SP together with their explanations:

Diagnostic Message	Explanation
NO UPDATE FUNCTION CARD	There is no control card preceding the Current File-SP.
RECORD TO BE DELETED NOT ON OLD MASTER FILE	The Current File-SP contains a DELETE card referencing a DDI-NO not on the old Source Program File.
CURRENT FILE CARDS ARE OUT OF SEQUENCE	The cards in the Current File-SP are not in sequence by DDI-NO.

Following is a list of the trace messages to be printed on the Audit File-SP.

The following messages are all paragraph names and are printed upon entering the paragraph:

1. IUC
2. UPDATE CONTROL
3. OLD MASTER FILE READOUT
4. END OF CURRENT FILE
5. END OF OLD MASTER FILE
6. END OF OLD MASTER FILE 4

The following typical trace message is printed whenever the WRITE-ERROR paragraph is entered:

LAST CARD KEY	0002A005
LAST CURRENT FILE KEY	0002A003
LAST OLD MASTER FILE KEY	0005A004

The information opposite the LAST CARD KEY represents the control field (columns 73-80) of the Current File-SP card read. The information opposite the LAST CURRENT FILE KEY represents the control field of the next to last Current File-SP card read. The information opposite LAST OLD MASTER FILE KEY represents the control field of the card image in the last physical record read from the old Source Program File. This trace information is printed on one line of the Audit File-SP.

4.3.3.2.3 Punch File-SP

Another option, the punch option, may be selected by the user to obtain a card deck of all card images on the Source Program File containing information.

4.4 Initiate Population File Module (INIPOP)

4.4.1 Purposes and Uses

This module may be used to assign new test serial numbers, DDI-NO's on the Population File. Renumbering the Population File might be required if the Test Modules were to be reorganized and placed in a different sequence or if within the current organizational structure of the Test Modules a new Test Module may not be assigned a convenient number relating it to its associated Test Modules.

Whatever the reason, INIPOP eliminates the necessity for re-keypunching the Population File card deck by automatically reassigning new DDI-NO's. The user is permitted to select the first new number to be assigned and an increment which will be added to successive assigned numbers to form new numbers for assignment. All DDI-NO

references on the Test Header card (including all mandatory DDI-NO's) and JOVIAL statement cards (including all mandatory DDI-NO references) will be updated to reflect the new number assignments.

Input to INIPOP consists of a Population File, in the form of a card deck or a magnetic tape, and a control card requesting specific options provided by the module.

Output from INIPOP includes a renumbered Population File, the Audit-File-IP, that contains diagnostic messages, an optional working listing on the Audit-File-IP consisting of those Population File card images containing information, and an optional Punch File-IP consisting of a card deck of all card images on the Population File containing information.

Population File modules recorded on cards may be renumbered in groups if desired. This feature of INIPOP is invoked on the card deck modules by placing control cards in the card deck before each independent group of modules to be renumbered. Each of the card deck modules following the control card will be renumbered according to the values given on the control card.

When invoking this feature on a Population File residing on tape, control cards designating the various renumbering conventions must also contain the DDI-NO of the last test module to which the current control card applies.

When a portion of the Population File is to be renumbered, the entire Population File should be submitted to INIPOP in order to ensure a correct resequencing of all embedded DDI-NO references. For those modules of the Population File not requiring renumbering, the control card must include the information that the following modules are to be included in the renumbering process but are not to be themselves renumbered.

4 4.2 Preparation of Inputs

4 4.2.1 Card Inputs

4 4.2.1.1 Control Card-IP

The various options provided by INIPOP may be requested by means of the following control card:

Columns	Name	Description
1	Control Card Indicator	This column must contain the character C denoting the card as a control card.
2-4	Control Card Identifier	These 3 columns may be assigned any 3 digits by the user to identify the control card.

Columns	Name	Description
5	Renumber Option*	The column is used to indicate to INIPOP that a renumbering of the Population File is required. non-space - Renumber space - Do not renumber
6-8	Card/Record	These 3 columns are used to designate the number of card images present in each record of the Population File.
9-12	Initial Number	These 4 columns are used to designate the first four digit DDI-NO to be assigned.
13-16	Increment	These 4 columns are used to designate the increment representing the difference between two successively assigned DDI-NO's.
17	Print Option	This column may be used to request the printing of the generated Population File non-space - Print space - Do not print
18	Punch Option	This column may be used to request the punching of the new Population File non-space - Punch space - Do not punch
19	Old Population File Option	This column may be used to signify that an old Population File residing on magnetic tape will be used as input. non-space - Magnetic tape input space - Card input
20-23	Record Maximum	These 4 columns are used to designate the DDI-NO of the last record to be incremented using the current initial number and increment.
24-80	Not Used	

*If renumbering is not selected, INIPOP may be used to initiate a Population File from a card deck or to copy an old Population File from one magnetic tape to the other. Print and punch options still apply.

When submitting this card to INIPOP, it precedes the Current File-PF if the Population File is to be generated from o card deck or it reploces the Current File-PF if the Populotion File is to be generated from on old Population File.

4.4.2.1.2 Other Cord Inputs

A complete description of all other card forms contained in either the Populotion File or the Current File-PF is given in Section 4.1.2.

4.4.2.2 Input Files

The Initiate Population File Module operotes on either of two input files, the Population File or the Current File-PF.

4.4.2.2.1 Population File

The Populotion File has been completely described in Section 4.1.2.4.1.

4.4.2.2.2 Current File-PF

The Current File-PF has been completely described in Section 4.1.2.4.2.

4.4.3 Function Operation

The Initiate Populotion File Module operates to initiote ond, ot the users option, renumber o Populotion File from either o Current File-PF or from an existing Population File. Additional features selectable from the control cord include the options to print o working listing of the generoted Populotion File ond/or to punch the resulting new Populotion File.

The Populotion File is renumbered by assigning to the first DDI-NO the value as stated on the Control Card-IP for the Initial Number; to the second DDI-NO, the Initial Value + Increment as stoted on the Control Card-IP; the third DDI-NO, the Initial Value + 2 * Increment. For example, if the Initiol Volue was specified as 5 ond the Increment was specified os 10, then the volues ossigned to the DDI-NO for eoch Test Module would be 5, 15, 25, etc., until the Test Modules hod been exhousted.

4.4.4 Description of Expected Results

4.4.4.1 Output Card Formats

The output card formats correspond to the formats for cords described in Section 4.1.2.

4.4.4.2 Output Files

The Initiate Population File Module produces three files: The Population File, the Audit File-IP, and the Punch File-IP.

4.4.4.2.1 Population File

The Population File is completely described in Section 4.1.2.4.1.

4.4.4.2.2 Audit File-IP

The Audit File-IP contains a listing of all diagnostics originating from the module. As an optional feature, the user may request to print on the Audit File-IP, a working listing of the card images on the new Population File by selecting the print option on the Control Card-IP. Since the Audit File-IP is only a working listing, diagnostic information will be interspersed with the Population File card images on the Audit File-IP. If no diagnostics occur, however, the Audit File-IP will consist entirely of a listing of the Population File.

Following is a list of the diagnostic messages to be printed in the Audit File-IP together with their explanations:

Diagnostic Message	Explanation
UNEXPECTED EOF INFILE	There is an unexpected end of file encountered on the unit containing the control card and Current File-PF.
DDI-NO LARGER THAN 9997	Successive incrementing of the originally assigned Initial Number have generated a number greater than 9997. There are too many Test Modules being renumbered given the particular assigned values for Initial Value and/or Increment. Reduce either value or both and try again.
NO CONTROL CARD	The control card has not been submitted to INIPOP.

4.4.4.2.3 Punch File-IP

Another option, the punch option, may be selected by the user to obtain a card deck of all card images on the Population File containing information.

4.5 JCVS Report Writer Module (JCVSRP)

4.5.1 Purposes and Uses

This module may be used to produce a finished listing of a Population File and/or a listing of the Test Header Cards in a Population File.

Input to this module consists of a Population File and a control card specifying the options available to the user.

Output from JCVSRP may include a listing of either the Population File or the collection of Test Header Cards on the Population File or both. These reports are printed on the Audit File-RP together with any diagnostics and trace messages originating from this module.

4.5.2 Preparation of Inputs

4.5.2.1 Card Inputs

4.5.2.1.1 Control Card-RP

The various options provided by JCVSRP may be requested by the following control card:

Columns	Name	Description
1	Control Card Indicator	This column must contain the character C denoting the card as a control card.
2-3	Report Selection	These 2 columns are used to select the two reports generated by this module. Column 2: non-space - Population File Listing space - No Population File Listing Column 3: non-space - Cross Referencing Listing space - No Cross Referencing Listing
4-13	Not Used	
14-19	Date	These six columns specify the date as follows: 14-15 Month 16-17 Day 18-19 Year (Example: 040968)

Columns	Name	Description
20-61	Test Identification	These 42 columns are used to specify the computer name. The name may be positioned any place in the field.
62-63	Control Tape Size	These two columns are used to specify the number of lines per printer page that are available to be printed on.
64-65	Line/Record	These two columns are used to specify the number of cards per record on the Population File. In this case of JCVS this value is 50.
66-80	Not Used	

4.5.2.2 Input Files

The JCVS Report Writer Module operates on one input file, the Population File.

4.5.2.2.1 Population File

The Population File has been completely described in Section 4.1.2.4.1.

4.5.3 Function Operation

The JCVS Report Writer Module operates on a Population File to produce two reports, a listing of the Test Modules on the Population File and/or a listing of the Test Header Cards on the Population File. The JCVSRP is directed by means of user options selected on the Control Card-RP.

4.5.4 Description of Expected Results

The JCVSRP produces one file, the Audit File-RP.

4.5.4.1 Audit File-RP

The Audit File-RP may contain either a listing of all the Test Modules on a Population File or a listing of all of the Test Header Cards on a Population File or both. This is a formal listing in that no diagnostics or trace messages are interspersed. A trace message does, however, precede the writing of each report on a separate page.

Following is the diagnostic message to be printed in the Audit File-RP together with its explanation:

Diagnostic Message	Explanation
UNEXPECTED EOF INFILE	This problem results from attempting to read the Control Card-RP and getting an end of file condition. Check input to make sure the control card is present and is not preceded by any extra end of file cards.

Following is the trace message that is printed out on a separate page at the beginning of the writing of each report:

REPORT WRITER.

SECTION V

USAGE INSTRUCTION

Since the JCVS will operate on several different computers it would be advisable if the user availed himself of the following documents:

1. Implementors COBOL Manual
2. Implementors Operating System Manual
3. Implementors JOVIAL J3 Manual

5.1 JCVS Operating Philosophy

Although the JCVS is to operate on various computers, the functions that will be performed on each computer to utilize the JCVS will be identical. Each of the JCVS program modules is processible by either of the following two methods:

1. Compile Source Program and Go

   Using this technique, the appropriate control cards, source program and data are submitted to the computer system. The system then compiles the source program and writes the resulting object program on the operating system's Load and Go unit. This object program is then loaded from the Load and Go unit and program executing follows.

2. Load Binary Deck and Go

   Using this technique, the appropriate control cards, object program binary deck, and data is submitted to the computer system. The system then loads the object program from the object program binary deck and program execution follows.

5.2 JCVS Function

There are seven functions that are available to the user of the JCVS. They are given in the following list:

1.	Create a new Population File	POPFM1
2.	Update an old Population File	POPFM2
3.	Generate a JOVIAL source program	SELECT
4.	Update a Source Program File	SOPMM
5.	Initiate a Population File from a Population File card deck	INIPOP1

6.	Initiate a new Population File from an old Population File on magnetic tape	INIPOP2
7.	Write reports from Population File	JCVSRP

5.3 Preparation of JCVS Input

5.3.1 Current File-PF

The Current File-PF which is used to update the Population File has been described in Section 4.1.2.4.2. An example of this file is given in Figures 5-1a, 5-1b, and 5-1c.

Notice all packets are in order by DDI-NO and that there are no DELETE packets.

5.3.2 Current File-SP

The Current File-SP which is used to update the Source Program File has been described in Section 4.3.1.2.2. An example of this file is given in Figure 5-2.

Notice that all of the cards in this file are in order by sequence number, columns 78-80 within DDI-NO, columns 73-76.

5.3.3 Test Selection File

The Test Selection File which directs the selection of the appropriate test modules has been described in Section 4.2.2.2.2. An example of this file is given in Figure 5-3.

This particular set of Test Selector Cards select the following test modules. In this example, it is assumed that no Mandatory DDI-NO's are involved.

5.4 Functional Processing

Diagrams will be proficed describing the status of the computer system at input time and again at output time for each function performed by the JCVS modules applying each operating philosophy and on each computer.

A complete list of these diagrams is given in Appendix 1.

5.5 Results of Operations

The JCVS modules generate magnetic tape output, printer listings and punched decks. The files associated with this output have already been completely described previously in this document. Actual samples of computer generated output will now be presented.

5.5.1 Printed Output

```
    TEST MODULE 0002 JOVIAL STATEMENT 003        ADD A CARD                0002J003
    TEST MODULE 0009 JOVIAL STATEMENT 050        ADD A CARD                0009J050
EST MODULE NAME 10 2427  2428  2429              M                 ''       0010A001
    TEST MODULE 0010 JOVIAL STATEMENT 001                                   0010J002
    TEST MODULE 0010 JOVIAL STATEMENT 002                                   0010J003
    TEST MODULE 0010 JOVIAL STATEMENT 003                                   0010J004
    TEST MODULE 0010 JOVIAL STATEMENT 004                                   0010J005
    TEST MODULE 0010 JOVIAL STATEMENT 005                                   0010J006
    TEST MODULE 0010 JOVIAL STATEMENT 006                                   0010J007
    TEST MODULE 0010 JOVIAL STATEMENT 007                                   0010J008
    TEST MODULE 0010 JOVIAL STATEMENT 008                                   0010J009
    TEST MODULE 0010 JOVIAL STATEMENT 009                                   0010J010
    TEST MODULE 0010 JOVIAL STATEMENT 010                                   0010J011
    TEST MODULE 0010 JOVIAL STATEMENT 011                                   0010J012
    TEST MODULE 0010 JOVIAL STATEMENT 012                                   0010J013
    TEST MODULE 0010 JOVIAL STATEMENT 013                                   0010J014
    TEST MODULE 0010 JOVIAL STATEMENT 014                                   0010J015
    TEST MODULE 0010 JOVIAL STATEMENT 015                                   0010J016
    TEST MODULE 0010 JOVIAL STATEMENT 016                                   0010J017
    TEST MODULE 0010 JOVIAL STATEMENT 017                                   0010J018
    TEST MODULE 0010 JOVIAL STATEMENT 018                                   0010J019
    TEST MODULE 0010 JOVIAL STATEMENT 019                                   0010J020
    TEST MODULE 0010 JOVIAL STATEMENT 020                                   0010J021
    TEST MODULE 0010 JOVIAL STATEMENT 021                                   0010J022
    TEST MODULE 0010 JOVIAL STATEMENT 022                                   0010J023
    TEST MODULE 0010 JOVIAL STATEMENT 023                                   0010J024
    TEST MODULE 0010 JOVIAL STATEMENT 024                                   0010J025
    TEST MODULE 0010 JOVIAL STATEMENT 025                                   0010J026
    TEST MODULE 0010 JOVIAL STATEMENT 026                                   0010J027
    TEST MODULE 0010 JOVIAL STATEMENT 027                                   0010J028
    TEST MODULE 0010 JOVIAL STATEMENT 028                                   0010J029
    TEST MODULE 0010 JOVIAL STATEMENT 029                                   0010J030
    TEST MODULE 0010 JOVIAL STATEMENT 030                                   0010J031
    TEST MODULE 0010 JOVIAL STATEMENT 031                                   0010J032
    TEST MODULE 0010 JOVIAL STATEMENT 032                                   0010J033
    TEST MODULE 0010 JOVIAL STATEMENT 033                                   0010J034
    TEST MODULE 0010 JOVIAL STATEMENT 034                                   0010J035
    TEST MODULE 0010 JOVIAL STATEMENT 035                                   0010J036
    TEST MODULE 0010 JOVIAL STATEMENT 036                                   0010J037
    TEST MODULE 0010 JOVIAL STATEMENT 037                                   0010J038
    TEST MODULE 0010 JOVIAL STATEMENT 038                                   0010J039
    TEST MODULE 0010 JOVIAL STATEMENT 039                                   0010J040
    TEST MODULE 0010 JOVIAL STATEMENT 040                                   0010J041
    TEST MODULE 0010 JOVIAL STATEMENT 041                                   0010J042
    TEST MODULE 0010 JOVIAL STATEMENT 042                                   0010J043
    TEST MODULE 0010 JOVIAL STATEMENT 043                                   0010J044
    TEST MODULE 0010 JOVIAL STATEMENT 044                                   0010J045
    TEST MODULE 0010 JOVIAL STATEMENT 045                                   0010J046
    TEST MODULE 0010 JOVIAL STATEMENT 046                                   0010J047
    TEST MODULE 0010 JOVIAL STATEMENT 047                                   0010J048
    TEST MODULE 0010 JOVIAL STATEMENT 048                                   0010J049
    TEST MODULE 0010 JOVIAL STATEMENT 049                                   0010J050
    TEST MODULE 0011 JOVIAL STATEMENT 015        REPLACE A CARD            0011J015
    TEST MODULE 0012 JOVIAL STATEMENT 004        REPLACE A CARD            0012J004
    TEST MODULE 0013 JOVIAL STATEMENT            DELETE MODULE 13          0013D000
ST MODULE NAME 14 2437                                    ''               0014A001
```

Figure 5-1a Current File-PF

```
    TEST MODULE 0014 JOVIAL STATEMENT 001                                  0014J002
    TEST MODULE 0014 JOVIAL STATEMENT 003                                  0014J004
    TEST MODULE 0014 JOVIAL STATEMENT 005                                  0014J006
    TEST MODULE 0014 JOVIAL STATEMENT 007                                  0014J008
    TEST MODULE 0014 JOVIAL STATEMENT 009                                  0014J010
    TEST MODULE 0014 JOVIAL STATEMENT 011                                  0014J012
    TEST MODULE 0014 JOVIAL STATEMENT 013                                  0014J014
    TEST MODULE 0014 JOVIAL STATEMENT 015                                  0014J016
    TEST MODULE 0014 JOVIAL STATEMENT 017                                  0014J018
    TEST MODULE 0014 JOVIAL STATEMENT 019                                  0014J020
    TEST MODULE 0014 JOVIAL STATEMENT 021                                  0014J022
    TEST MODULE 0014 JOVIAL STATEMENT 023                                  0014J024
    TEST MODULE 0014 JOVIAL STATEMENT 025                                  0014J026
    TEST MODULE 0014 JOVIAL STATEMENT 027                                  0014J028
    TEST MODULE 0014 JOVIAL STATEMENT 029                                  0014J030
    TEST MODULE 0014 JOVIAL STATEMENT 031                                  0014J032
    TEST MODULE 0014 JOVIAL STATEMENT 033                                  0014J034
    TEST MODULE 0014 JOVIAL STATEMENT 035                                  0014J036
    TEST MODULE 0014 JOVIAL STATEMENT 037                                  0014J038
    TEST MODULE 0014 JOVIAL STATEMENT 039                                  0014J040
    TEST MODULE 0014 JOVIAL STATEMENT 041                                  0014J042
    TEST MODULE 0014 JOVIAL STATEMENT 043                                  0014J044
    TEST MODULE 0014 JOVIAL STATEMENT 045                                  0014J046
    TEST MODULE 0014 JOVIAL STATEMENT 047                                  0014J048
    TEST MODULE 0014 JOVIAL STATEMENT 049                                  0014J050
    TEST MODULE 0015 JOVIAL STATEMENT 003      DELETE A CARD               0015D003
    TEST MODULE 0016 JOVIAL STATEMENT 020      REPLACE A CARD              0016J020
    TEST MODULE 0017 JOVIAL STATEMENT 006      ADD A CARD                  0017J006
    TEST MODULE 0017 JOVIAL STATEMENT 021      REPLACE A CARD              0017J021
    TEST MODULE 0018 JOVIAL STATEMENT 007      ADD A CARD                  0018J007
EST MODULE NAME 19 2445                              0016''                0019A001
    TEST MODULE 0019 JOVIAL STATEMENT 002                                  0019J003
    TEST MODULE 0019 JOVIAL STATEMENT 004                                  0019J005
    TEST MODULE 0019 JOVIAL STATEMENT 006                                  0019J007
    TEST MODULE 0019 JOVIAL STATEMENT 008                                  0019J009
    TEST MODULE 0019 JOVIAL STATEMENT 010                                  0019J011
    TEST MODULE 0019 JOVIAL STATEMENT 012                                  0019J013
    TEST MODULE 0019 JOVIAL STATEMENT 014                                  0019J015
    TEST MODULE 0019 JOVIAL STATEMENT 016                                  0019J017
    TEST MODULE 0019 JOVIAL STATEMENT 018                                  0019J019
    TEST MODULE 0019 JOVIAL STATEMENT 020                                  0019J021
    TEST MODULE 0019 JOVIAL STATEMENT 022                                  0019J023
    TEST MODULE 0019 JOVIAL STATEMENT 024                                  0019J025
    TEST MODULE 0019 JOVIAL STATEMENT 026                                  0019J027
    TEST MODULE 0019 JOVIAL STATEMENT 028                                  0019J029
    TEST MODULE 0019 JOVIAL STATEMENT 030                                  0019J031
    TEST MODULE 0019 JOVIAL STATEMENT 032                                  0019J033
    TEST MODULE 0019 JOVIAL STATEMENT 034                                  0019J035
    TEST MODULE 0019 JOVIAL STATEMENT 036                                  0019J037
    TEST MODULE 0019 JOVIAL STATEMENT 038                                  0019J039
    TEST MODULE 0019 JOVIAL STATEMENT 040                                  0019J041
    TEST MODULE 0019 JOVIAL STATEMENT 042                                  0019J043
    TEST MODULE 0019 JOVIAL STATEMENT 044                                  0019J045
    TEST MODULE 0019 JOVIAL STATEMENT 046                                  0019J047
    TEST MODULE 0019 JOVIAL STATEMENT 048                                  0019J049
    TEST MODULE 0021 JOVIAL STATEMENT          REPLACE MODULE 21           0021D000
```

Figure 5-1b Current File-PF

```
TEST MODULE 0021 JOVIAL STATEMENT 001        REPLACE MODULE 21      0021A001
TEST MODULE 0021 JOVIAL STATEMENT 002        REPLACE MODULE 21      0021J002
TEST MODULE 0021 JOVIAL STATEMENT 003        REPLACE MODULE 21      0021J003
TEST MODULE 0021 JOVIAL STATEMENT 004        REPLACE MODULE 21      0021J004
TEST MODULE 0021 JOVIAL STATEMENT 005        REPLACE MODULE 21      0021J005
TEST MODULE 0021 JOVIAL STATEMENT 010        REPLACE MODULE 21      0021J010
TEST MODULE 0021 JOVIAL STATEMENT 025        REPLACE MODULE 21      0021J025
TEST MODULE 0021 JOVIAL STATEMENT 050        REPLACE MODULE 21      0021J050
TEST MODULE 0022 JOVIAL STATEMENT 032        DELETE A CARD          0022D032
TEST MODULE 0023 JOVIAL STATEMENT 007        DELETE A CARD          0023D007
TEST MODULE 0024 JOVIAL STATEMENT 026        DELETE A CARD          0024D026
TEST MODULE 0025 JOVIAL STATEMENT            DELETE MODULE 25       0025D000
```

Figure 5-1c Current File-PF

```
TEST MODULE 0003 JOVIAL STATEMENT 004          ADD A CARD              0003J004
TEST MODULE 0003 JOVIAL STATEMENT 005          DELETE A CARD           0003D005
TEST MODULE 0008 JOVIAL STATEMENT 024          DELETE A CARD           0008D024
TEST MODULE 0008 JOVIAL STATEMENT 036          REPLACE A CARD          0008J036
TEST MODULE 0009 JOVIAL STATEMENT 020          ADD A CARD              0009J020
TEST MODULE 0011 JOVIAL STATEMENT 017          REPLACE A CARD          0011J017
TEST MODULE 0012 JOVIAL STATEMENT 031          ADD A CARD              0012J031
TEST MODULE 0013 JOVIAL STATEMENT 021          REPLACE A CARD          0013J021
TEST MODULE 0024 JOVIAL STATEMENT 020          DELETE A CARD           0024D020
TEST MODULE 0024 JOVIAL STATEMENT 025          ADD A CARD              0024J025
TEST MODULE 0024 JOVIAL STATEMENT 050          REPLACE A CARD          0024J050
TEST MODULE 0025 JOVIAL STATEMENT 017          DELETE A CARD           0025D017
```

Figure 5-2 Current File-SP

```
T       0002
T       0003        0003        0012
T       0025        0000
T       0024
T       0005
T       0009        0002        0013
```

Figure 5-3 Test Selection File

5.5.1.1 Population File

Figure 5-4 shows portions of a tape dump of the Population File from the GE-635. Exact positions of the test statements within the block should be noted. Since this is test information, the content of the various cards in the record are not actual JOVIAL statements but indications as to where Population File information would replace the checkout statements.

5.5.1.2 Audit File-PF

Figure 5-5 presents a portion of the listing of the Audit File-PF generated by POPFM on the GE-635. Notice diagnostic and trace messages interspersed with the list of the new Population File.

5.5.1.3 Audit File-S

Figures 5-6A and 5-6B present a portion of the Audit File-S generated by SJCVS on the GE-635.

5.5.1.4 Audit File-SP

Figure 5-7 presents a portion of the Audit File-SP generated by SOPMM on the GE-635. Notice that no trace messages appear in the listing giving the user a "clean" listing of the new Source Program File.

5.5.1.5 Audit File-IP

Figure 5-8 presents a portion of the Audit File-IP generated by INIPOP on the GE-635. The diagnostic messages 'MANDATORY MODULE NOT ON POPULATION FILE' are printed but processing is permitted to continue. Notice that no trace messages appear on the listing giving the user a "clean" listing (except for diagnostics) of the new Population File.

5.5.1.6 Audit File-RP

Figures 5-9A and 5-9B present a portion of a listing of the Audit File-RP generated by JCVSRP on the GE-635. The trace message appears on a separate page thereby giving the user a "clean" listing of the two reports: The POPULATION FILE and the CROSS REFERENCE TABLE.

5.5.2 Punched Output

```
18499 02  12-16-68                    UTILITY REPORT-050268                    PAGE      1
$      UTILITY
$      TAPE    F1,X3R
$      FFILE   F1,PHYREC
$      FUTIL   F1;,RWD/F1/;DUMP/1F/
FILE #     1 -- RECORD #     1 -- FILE CODE - F1 -- REPORT CODE XX  ---MODE----BCD-  --DENSITY---HIGH-

     1    202064622126   522562242020   203021456223   464420212622   202027255206       USAF-ESD   HANSCOM AFB  GE-6
     6    030520204121   452003002001   110611000141   214520011120   011106112020   35  JAN 30 196901JAN 19 1969
    11    202020202020   202020202020   000000012100   000141466531   214320234644                0001A001JOVIAL COM
    16    473143255120   652143312421   633146452062   706263254420   012020202020   PILER VALIDATION SYSTEM 1
    21    202020202020   202020202020   272523466220   202020202020   202020202020                GECOS
    26    202000000001   210000022101   202020202020   202020202020   202020202020     0001A002A1
    31    202020202020   202020202102   202646512043   31626331 4527  202020202020               A2 FOR LISTING
    36    202020202020   202020202020   200605422020   202020200000   000121000003                 65K       0001A003
    41    210520264651   202321512462   202020202020   202020202020   202020202020   A5 FOR CARDS
    46    210320202020   202020202020   202020202020   202020202020   202020202020   A3
    51    202020202020   202020202020   000000012100   000421042020   202020202020                0001A004A4
    56    202020202020   202020202020   202020202020   202021062020   202020202020                      A6
    61    202020202020   202020202020   202020202020   202020202020   202020202020
    66    202000000001   210000052020   202020202020   202020202020   202020202020     0001A005
    71    202020202020   202020202020   202020202020   202020202020   202020202020
   121*   202020202020   204325212431   452720464725   512163314527   206270626325            LEADING OPERATING SYSTE
   126    442023464563   514643202321   512420200000   012020202020   202020202020   M CONTROL CARD  001
   131    202020202020   202020202020   000000014300   010020202020   202020432521                0001L010       LEA
   136    243145272046   472551216331   452720627062   632544202346   456351464320   DING OPERATING SYSTEM CONTROL
   141    232151242020   000002202020   202020202020   202020202020   202020202020   CARD  002
   146    202000000001   430001012020   202020202043   252124314527   204647255121     0001L011       LEADING OPERA
   151    633145272062   706263254420   234645635146   432023215124   202000000320   TING SYSTEM CONTROL CARD  003
   156    202020202020   202020202020   202020202020   202020200000   000143000102                              0001L012
   161    202020202020   202631452143   202020464725   512163314527   206270626325            FINAL   OPERATING SYSTE
   166    442023464563   514643202321   512420200000   012020202020   202020202020   M CONTROL CARD  001
   171    202020202020   202020202020   000000012600   010320202020   202020432521                0001F013       LEA
   176    243145272046   472551216331   452720627062   632544202346   456351464320   DING OPERATING SYSTEM CONTROL
   181    232151242020   000004202020   202020202020   202020202020   202020202020   CARD  004
   186    202000000001   430001042020   202020202020   202020202020   202020202020     0001L014
   191    202020202020   202020202020   202020202020   202020202020   202020202020
   266*   202020202020   202020202020   202020202026   314521432020   204647255121                        FINAL   OPERA
   271    633145272062   706263254420   234645635146   432023215124   202000000220   TING SYSTEM CONTROL CARD  002
   276    202020202020   202020202020   202020202020   202020200000   000126000801                              0001F021
   281    202020202020   202020202020   202020202020   202020202020   202020202020
   321*   202020202020   202631452143   202020464725   512163314527   206270626325            FINAL   OPERATING SYSTE
   326    442023464563   514643202321   512420200000   032020202020   202020202020   M CONTROL CARD  003
   331    202020202020   202020202020   000000012600   020520202020   202020202020                0001F025
   336    202020202020   202020202020   202020202020   202020202020   202020202020
   666*   202020202020   202020200000                                                                    00

FILE #     1 -- RECORD #     2 -- FILE CODE - F1 -- REPORT CODE XX  ---MODE----BCD-  --DENSITY---HIGH-
```


Figure 5-4 Population File Tape Dump, GE-635

```
      TEST MODULE 0015 JOVIAL STATEMENT 013                          0015J014
      TEST MODULE 0015 JOVIAL STATEMENT 014                          0015J015
      TEST MODULE 0015 JOVIAL STATEMENT 015                          0015J016
      TEST MODULE 0015 JOVIAL STATEMENT 016                          0015J017
      TEST MODULE 0015 JOVIAL STATEMENT 017                          0015J018
      TEST MODULE 0015 JOVIAL STATEMENT 018                          0015J019
      TEST MODULE 0015 JOVIAL STATEMENT 019                          0015J020
      TEST MODULE 0015 JOVIAL STATEMENT 020                          0015J021
      TEST MODULE 0015 JOVIAL STATEMENT 021                          0015J022
      TEST MODULE 0015 JOVIAL STATEMENT 022                          0015J023
      TEST MODULE 0015 JOVIAL STATEMENT 023                          0015J024
      TEST MODULE 0015 JOVIAL STATEMENT 024                          0015J025
      TEST MODULE 0015 JOVIAL STATEMENT 025                          0015J026
      TEST MODULE 0015 JOVIAL STATEMENT 026                          0015J027
      TEST MODULE 0015 JOVIAL STATEMENT 027                          0015J028
      TEST MODULE 0015 JOVIAL STATEMENT 028                          0015J029
      TEST MODULE 0015 JOVIAL STATEMENT 029                          0015J030
      TEST MODULE 0015 JOVIAL STATEMENT 030                          0015J031
      TEST MODULE 0015 JOVIAL STATEMENT 031                          0015J032
      TEST MODULE 0015 JOVIAL STATEMENT 032                          0015J033
      TEST MODULE 0015 JOVIAL STATEMENT 033                          0015J034
      TEST MODULE 0015 JOVIAL STATEMENT 034                          0015J035
      TEST MODULE 0015 JOVIAL STATEMENT 035                          0015J036
      TEST MODULE 0015 JOVIAL STATEMENT 036                          0015J037
      TEST MODULE 0015 JOVIAL STATEMENT 037                          0015J038
      TEST MODULE 0015 JOVIAL STATEMENT 038                          0015J039
      TEST MODULE 0015 JOVIAL STATEMENT 039                          0015J040
      TEST MODULE 0015 JOVIAL STATEMENT 040                          0015J041
      TEST MODULE 0015 JOVIAL STATEMENT 041                          0015J042
      TEST MODULE 0015 JOVIAL STATEMENT 042                          0015J043
      TEST MODULE 0015 JOVIAL STATEMENT 043                          0015J044
      TEST MODULE 0015 JOVIAL STATEMENT 044                          0015J045
      TEST MODULE 0015 JOVIAL STATEMENT 045                          0015J046
      TEST MODULE 0015 JOVIAL STATEMENT 046                          0015J047
      TEST MODULE 0015 JOVIAL STATEMENT 047                          0015J048
      TEST MODULE 0015 JOVIAL STATEMENT 048                          0015J049
      TEST MODULE 0015 JOVIAL STATEMENT 049                          0015J050
UPDATE CONTROL
LAST CARD KEY  0016J020  LAST CURRENT FILE KEY 0015D003  LAST OLD MASTER FILE KEY 0016A001

**TEST MODULE NAME 16 2441  2442                          **         0016A001
      TEST MODULE 0016 JOVIAL STATEMENT 001                          0016J002
      TEST MODULE 0016 JOVIAL STATEMENT 003                          0016J004
      TEST MODULE 0016 JOVIAL STATEMENT 005                          0016J006
      TEST MODULE 0016 JOVIAL STATEMENT 007                          0016J008
      TEST MODULE 0016 JOVIAL STATEMENT 009                          0016J010
      TEST MODULE 0016 JOVIAL STATEMENT 011                          0016J012
      TEST MODULE 0016 JOVIAL STATEMENT 013                          0016J014
      TEST MODULE 0016 JOVIAL STATEMENT 015                          0016J016
      TEST MODULE 0016 JOVIAL STATEMENT 017                          0016J018
      TEST MODULE 0016 JOVIAL STATEMENT 020      REPLACE A CARD      0016J020
      TEST MODULE 0016 JOVIAL STATEMENT 021                          0016J022
      TEST MODULE 0016 JOVIAL STATEMENT 023                          0016J024
      TEST MODULE 0016 JOVIAL STATEMENT 025                          0016J026
      TEST MODULE 0016 JOVIAL STATEMENT 027                          0016J028
      TEST MODULE 0016 JOVIAL STATEMENT 029                          0016J030
      TEST MODULE 0016 JOVIAL STATEMENT 031                          0016J032
      TEST MODULE 0016 JOVIAL STATEMENT 033                          0016J034
      TEST MODULE 0016 JOVIAL STATEMENT 035                          0016J036
```


Figure 5-5 Audit File-PF, GE-635

```
BDT1
0002
BDT1
0003
0006
0009
0012
BDT1
0025
BDT1
0024
BDT1
0005
BDT1
0009
0011
0013
BDT1
BMT1
BMT1
BMT1                    0003
BMT1
BMT1                    0003
BMT1                    0008
BMT1                    0006
BMT1                    0012
BMT1
BMT1
BUILD-SPF
```


Figure 5-6a Audit File-S, GE-635

```
      LEADING OPERATING SYSTEM CONTROL CARD  001                0001L010
      LEADING OPERATING SYSTEM CONTROL CARD  002                0001L011
      LEADING OPERATING SYSTEM CONTROL CARD  003                0001L012
      LEADING OPERATING SYSTEM CONTROL CARD  004                0001L014
START                                                           0001L051

$$TEST MODULE NAME 02 2407  2408  2409                          0002A001
      TEST MODULE 0002 JOVIAL STATEMENT 001                     0002J002
      TEST MODULE 0002 JOVIAL STATEMENT 003                     0002J004
      TEST MODULE 0002 JOVIAL STATEMENT 005                     0002J006
      TEST MODULE 0002 JOVIAL STATEMENT 007                     0002J008
      TEST MODULE 0002 JOVIAL STATEMENT 009                     0002J010
      TEST MODULE 0002 JOVIAL STATEMENT 011                     0002J012
      TEST MODULE 0002 JOVIAL STATEMENT 013                     0002J014
      TEST MODULE 0002 JOVIAL STATEMENT 015                     0002J016
      TEST MODULE 0002 JOVIAL STATEMENT 017                     0002J018
      TEST MODULE 0002 JOVIAL STATEMENT 019                     0002J020
      TEST MODULE 0002 JOVIAL STATEMENT 021                     0002J022
      TEST MODULE 0002 JOVIAL STATEMENT 023                     0002J024
      TEST MODULE 0002 JOVIAL STATEMENT 025                     0002J026
      TEST MODULE 0002 JOVIAL STATEMENT 027                     0002J028
      TEST MODULE 0002 JOVIAL STATEMENT 029                     0002J030
      TEST MODULE 0002 JOVIAL STATEMENT 031                     0002J032
      TEST MODULE 0002 JOVIAL STATEMENT 033                     0002J034
      TEST MODULE 0002 JOVIAL STATEMENT 035                     0002J036
      TEST MODULE 0002 JOVIAL STATEMENT 037                     0002J038
      TEST MODULE 0002 JOVIAL STATEMENT 039                     0002J040
      TEST MODULE 0002 JOVIAL STATEMENT 041                     0002J042
      TEST MODULE 0002 JOVIAL STATEMENT 043                     0002J044
      TEST MODULE 0002 JOVIAL STATEMENT 045                     0002J046
      TEST MODULE 0002 JOVIAL STATEMENT 047                     0002J048
      TEST MODULE 0002 JOVIAL STATEMENT 049                     0002J050

$$TEST MODULE NAME 03 2410  2411  2412  2413                    0003A001
      TEST MODULE 0003 JOVIAL STATEMENT 002                     0003J003
      TEST MODULE 0003 JOVIAL STATEMENT 004                     0003J005
      TEST MODULE 0003 JOVIAL STATEMENT 006                     0003J007
      TEST MODULE 0003 JOVIAL STATEMENT 008                     0003J009
      TEST MODULE 0003 JOVIAL STATEMENT 010                     0003J011
      TEST MODULE 0003 JOVIAL STATEMENT 012                     0003J013
      TEST MODULE 0003 JOVIAL STATEMENT 014                     0003J015
      TEST MODULE 0003 JOVIAL STATEMENT 016                     0003J017
      TEST MODULE 0003 JOVIAL STATEMENT 018                     0003J019
      TEST MODULE 0003 JOVIAL STATEMENT 020                     0003J021
      TEST MODULE 0003 JOVIAL STATEMENT 022                     0003J023
      TEST MODULE 0003 JOVIAL STATEMENT 024                     0003J025
      TEST MODULE 0003 JOVIAL STATEMENT 026                     0003J027
      TEST MODULE 0003 JOVIAL STATEMENT 028                     0003J029
      TEST MODULE 0003 JOVIAL STATEMENT 030                     0003J031
      TEST MODULE 0003 JOVIAL STATEMENT 032                     0003J033
      TEST MODULE 0003 JOVIAL STATEMENT 034                     0003J035
      TEST MODULE 0003 JOVIAL STATEMENT 036                     0003J037
      TEST MODULE 0003 JOVIAL STATEMENT 038                     0003J039
      TEST MODULE 0003 JOVIAL STATEMENT 040                     0003J041
      TEST MODULE 0003 JOVIAL STATEMENT 042                     0003J043
      TEST MODULE 0003 JOVIAL STATEMENT 044                     0003J045
      TEST MODULE 0003 JOVIAL STATEMENT 046                     0003J047
      TEST MODULE 0003 JOVIAL STATEMENT 048                     0003J049

$$TEST MODULE NAME 05 2415  2416  2417  2418  2419  2420  00   0005A001
```

Figure 5-6b Audit File-S, GE-635

```
TEST MODULE 0013 JOVIAL STATEMENT 028                    0013J029
TEST MODULE 0013 JOVIAL STATEMENT 030                    0013J031
TEST MODULE 0013 JOVIAL STATEMENT 032                    0013J033
TEST MODULE 0013 JOVIAL STATEMENT 034                    0013J035
TEST MODULE 0013 JOVIAL STATEMENT 036                    0013J037
TEST MODULE 0013 JOVIAL STATEMENT 038                    0013J039
TEST MODULE 0013 JOVIAL STATEMENT 040                    0013J041
TEST MODULE 0013 JOVIAL STATEMENT 042                    0013J043
TEST MODULE 0013 JOVIAL STATEMENT 044                    0013J045
TEST MODULE 0013 JOVIAL STATEMENT 046                    0013J047
TEST MODULE 0013 JOVIAL STATEMENT 048                    0013J049
  @@TEST MODULE NAME 24 2455  2456                       0024A001
TEST MODULE 0024 JOVIAL STATEMENT 001                    0024J002
TEST MODULE 0024 JOVIAL STATEMENT 003                    0024J004
TEST MODULE 0024 JOVIAL STATEMENT 005                    0024J006
TEST MODULE 0024 JOVIAL STATEMENT 007                    0024J008
TEST MODULE 0024 JOVIAL STATEMENT 009                    0024J010
TEST MODULE 0024 JOVIAL STATEMENT 011                    0024J012
TEST MODULE 0024 JOVIAL STATEMENT 013                    0024J014
TEST MODULE 0024 JOVIAL STATEMENT 015                    0024J016
TEST MODULE 0024 JOVIAL STATEMENT 017                    0024J018
TEST MODULE 0024 JOVIAL STATEMENT 021                    0024J022
TEST MODULE 0024 JOVIAL STATEMENT 023                    0024J024
TEST MODULE 0024 JOVIAL STATEMENT 025        ADD A CARD  0024J025
TEST MODULE 0024 JOVIAL STATEMENT 025                    0024J026
TEST MODULE 0024 JOVIAL STATEMENT 027                    0024J028
TEST MODULE 0024 JOVIAL STATEMENT 029                    0024J030
TEST MODULE 0024 JOVIAL STATEMENT 031                    0024J032
TEST MODULE 0024 JOVIAL STATEMENT 033                    0024J034
TEST MODULE 0024 JOVIAL STATEMENT 035                    0024J036
TEST MODULE 0024 JOVIAL STATEMENT 037                    0024J038
TEST MODULE 0024 JOVIAL STATEMENT 039                    0024J040
TEST MODULE 0024 JOVIAL STATEMENT 041                    0024J042
TEST MODULE 0024 JOVIAL STATEMENT 043                    0024J044
TEST MODULE 0024 JOVIAL STATEMENT 045                    0024J046
TEST MODULE 0024 JOVIAL STATEMENT 047                    0024J048
TEST MODULE 0024 JOVIAL STATEMENT 050     REPLACE A CARD 0024J050
  @@TEST MODULE NAME 25 2457  2458  2459                 0025A001
TEST MODULE 0025 JOVIAL STATEMENT 001                    0025J002
TEST MODULE 0025 JOVIAL STATEMENT 002                    0025J003
TEST MODULE 0025 JOVIAL STATEMENT 003                    0025J004
TEST MODULE 0025 JOVIAL STATEMENT 004                    0025J005
TEST MODULE 0025 JOVIAL STATEMENT 005                    0025J006
TEST MODULE 0025 JOVIAL STATEMENT 006                    0025J007
TEST MODULE 0025 JOVIAL STATEMENT 007                    0025J008
TEST MODULE 0025 JOVIAL STATEMENT 008                    0025J009
TEST MODULE 0025 JOVIAL STATEMENT 009                    0025J010
TEST MODULE 0025 JOVIAL STATEMENT 010                    0025J011
TEST MODULE 0025 JOVIAL STATEMENT 011                    0025J012
TEST MODULE 0025 JOVIAL STATEMENT 012                    0025J013
```


Figure 5-7 Audit File-SP, GE-635

```
                                MANDATORY MODULE NOT ON POPULATION FILE
**TEST MODULE NAME 15 2438  2439  2440                    002500          0005A001
      TEST MODULE 0015 JOVIAL STATEMENT 001                               0005J002
      TEST MODULE 0015 JOVIAL STATEMENT 002                               0005J003
      TEST MODULE 0015 JOVIAL STATEMENT 003                               0005J004
      TEST MODULE 0015 JOVIAL STATEMENT 004                               0005J005
      TEST MODULE 0015 JOVIAL STATEMENT 005                               0005J006
      TEST MODULE 0015 JOVIAL STATEMENT 006                               0005J007
      TEST MODULE 0015 JOVIAL STATEMENT 007                               0005J008
      TEST MODULE 0015 JOVIAL STATEMENT 008                               0005J009
      TEST MODULE 0015 JOVIAL STATEMENT 009                               0005J010
      TEST MODULE 0015 JOVIAL STATEMENT 010                               0005J011
      TEST MODULE 0015 JOVIAL STATEMENT 011                               0005J012
      TEST MODULE 0015 JOVIAL STATEMENT 012                               0005J013
      TEST MODULE 0015 JOVIAL STATEMENT 013                               0005J014
      TEST MODULE 0015 JOVIAL STATEMENT 014                               0005J015
      TEST MODULE 0015 JOVIAL STATEMENT 015                               0005J016
      TEST MODULE 0015 JOVIAL STATEMENT 016                               0005J017
      TEST MODULE 0015 JOVIAL STATEMENT 017                               0005J018
      TEST MODULE 0015 JOVIAL STATEMENT 018                               0005J019
      TEST MODULE 0015 JOVIAL STATEMENT 019                               0005J020
      TEST MODULE 0015 JOVIAL STATEMENT 020                               0005J021
      TEST MODULE 0015 JOVIAL STATEMENT 021                               0005J022
      TEST MODULE 0015 JOVIAL STATEMENT 022                               0005J023
      TEST MODULE 0015 JOVIAL STATEMENT 023                               0005J024
      TEST MODULE 0015 JOVIAL STATEMENT 024                               0005J025
      TEST MODULE 0015 JOVIAL STATEMENT 025                               0005J026
      TEST MODULE 0015 JOVIAL STATEMENT 026                               0005J027
      TEST MODULE 0015 JOVIAL STATEMENT 027                               0005J028
      TEST MODULE 0015 JOVIAL STATEMENT 028                               0005J029
      TEST MODULE 0015 JOVIAL STATEMENT 029                               0005J030
      TEST MODULE 0015 JOVIAL STATEMENT 030                               0005J031
      TEST MODULE 0015 JOVIAL STATEMENT 031                               0005J032
      TEST MODULE 0015 JOVIAL STATEMENT 032                               0005J033
      TEST MODULE 0015 JOVIAL STATEMENT 033                               0005J034
      TEST MODULE 0015 JOVIAL STATEMENT 034                               0005J035
      TEST MODULE 0015 JOVIAL STATEMENT 035                               0005J036
      TEST MODULE 0015 JOVIAL STATEMENT 036                               0005J037
      TEST MODULE 0015 JOVIAL STATEMENT 037                               0005J038
      TEST MODULE 0015 JOVIAL STATEMENT 038                               0005J039
      TEST MODULE 0015 JOVIAL STATEMENT 039                               0005J040
      TEST MODULE 0015 JOVIAL STATEMENT 040                               0005J041
      TEST MODULE 0015 JOVIAL STATEMENT 041                               0005J042
      TEST MODULE 0015 JOVIAL STATEMENT 042                               0005J043
      TEST MODULE 0015 JOVIAL STATEMENT 043                               0005J044
      TEST MODULE 0015 JOVIAL STATEMENT 044                               0005J045
      TEST MODULE 0015 JOVIAL STATEMENT 045                               0005J046
      TEST MODULE 0015 JOVIAL STATEMENT 046                               0005J047
      TEST MODULE 0015 JOVIAL STATEMENT 047                               0005J048
      TEST MODULE 0015 JOVIAL STATEMENT 048                               0005J049
      TEST MODULE 0015 JOVIAL STATEMENT 049                               0005J050

                                MANDATORY MODULE NOT ON POPULATION FILE
**TEST MODULE NAME 09 2425  2426                          **              0015A001
      TEST MODULE 0009 JOVIAL STATEMENT 002                               0015J003
      TEST MODULE 0009 JOVIAL STATEMENT 004                               0015J005
      TEST MODULE 0009 JOVIAL STATEMENT 006                               0015J007
      TEST MODULE 0009 JOVIAL STATEMENT 008                               0015J009
      TEST MODULE 0009 JOVIAL STATEMENT 010                               0015J011
      TEST MODULE 0009 JOVIAL STATEMENT 012                               0015J013
```

Figure 5-8 Audit File-IP, GE-635

POPULATION FILE
GENERAL ELECTRIC 635
NOV 30 1968

```
**TEST MODULE NAME 15 2438  2439  2440                    0014     **  0015A001
      TEST MODULE 0015 JOVIAL STATEMENT 001                            0015J002
      TEST MODULE 0015 JOVIAL STATEMENT 002                            0015J003
      TEST MODULE 0015 JOVIAL STATEMENT 003                            0015J004
      TEST MODULE 0015 JOVIAL STATEMENT 004                            0015J005
      TEST MODULE 0015 JOVIAL STATEMENT 005                            0015J006
      TEST MODULE 0015 JOVIAL STATEMENT 006                            0015J007
      TEST MODULE 0015 JOVIAL STATEMENT 007                            0015J008
      TEST MODULE 0015 JOVIAL STATEMENT 008                            0015J009
      TEST MODULE 0015 JOVIAL STATEMENT 009                            0015J010
      TEST MODULE 0015 JOVIAL STATEMENT 010                            0015J011
      TEST MODULE 0015 JOVIAL STATEMENT 011                            0015J012
      TEST MODULE 0015 JOVIAL STATEMENT 012                            0015J013
      TEST MODULE 0015 JOVIAL STATEMENT 013                            0015J014
      TEST MODULE 0015 JOVIAL STATEMENT 014                            0015J015
      TEST MODULE 0015 JOVIAL STATEMENT 015                            0015J016
      TEST MODULE 0015 JOVIAL STATEMENT 016                            0015J017
      TEST MODULE 0015 JOVIAL STATEMENT 017                            0015J018
      TEST MODULE 0015 JOVIAL STATEMENT 018                            0015J019
      TEST MODULE 0015 JOVIAL STATEMENT 019                            0015J020
      TEST MODULE 0015 JOVIAL STATEMENT 020                            0015J021
      TEST MODULE 0015 JOVIAL STATEMENT 021                            0015J022
      TEST MODULE 0015 JOVIAL STATEMENT 022                            0015J023
      TEST MODULE 0015 JOVIAL STATEMENT 023                            0015J024
      TEST MODULE 0015 JOVIAL STATEMENT 024                            0015J025
      TEST MODULE 0015 JOVIAL STATEMENT 025                            0015J026
      TEST MODULE 0015 JOVIAL STATEMENT 026                            0015J027
      TEST MODULE 0015 JOVIAL STATEMENT 027                            0015J028
      TEST MODULE 0015 JOVIAL STATEMENT 028                            0015J029
      TEST MODULE 0015 JOVIAL STATEMENT 029                            0015J030
      TEST MODULE 0015 JOVIAL STATEMENT 030                            0015J031
      TEST MODULE 0015 JOVIAL STATEMENT 031                            0015J032
      TEST MODULE 0015 JOVIAL STATEMENT 032                            0015J033
      TEST MODULE 0015 JOVIAL STATEMENT 033                            0015J034
      TEST MODULE 0015 JOVIAL STATEMENT 034                            0015J035
      TEST MODULE 0015 JOVIAL STATEMENT 035                            0015J036
      TEST MODULE 0015 JOVIAL STATEMENT 036                            0015J037
      TEST MODULE 0015 JOVIAL STATEMENT 037                            0015J038
      TEST MODULE 0015 JOVIAL STATEMENT 038                            0015J039
      TEST MODULE 0015 JOVIAL STATEMENT 039                            0015J040
      TEST MODULE 0015 JOVIAL STATEMENT 040                            0015J041
      TEST MODULE 0015 JOVIAL STATEMENT 041                            0015J042
      TEST MODULE 0015 JOVIAL STATEMENT 042                            0015J043
      TEST MODULE 0015 JOVIAL STATEMENT 043                            0015J044
      TEST MODULE 0015 JOVIAL STATEMENT 044                            0015J045
      TEST MODULE 0015 JOVIAL STATEMENT 045                            0015J046
      TEST MODULE 0015 JOVIAL STATEMENT 046                            0015J047
      TEST MODULE 0015 JOVIAL STATEMENT 047                            0015J048
      TEST MODULE 0015 JOVIAL STATEMENT 048                            0015J049
      TEST MODULE 0015 JOVIAL STATEMENT 049                            0015J050
```

(23)

Figure 5-9a Audit File-RP

CROSS REFERENCE TABLE
GENERAL ELECTRIC 635
NOV 30 1968

DDI NUMBER	TEST NAME	CED-NO	CED-NO	CED-NO	CED-NO	CED-NO	CED-NO	REQ DDI-NO
0002	TEST MODULE NAME 02	2407	2408	2409				
0003	TEST MODULE NAME 03	2410	2411	2412	2413			
0004	TEST MODULE NAME 04	2414						
0005	TEST MODULE NAME 05	2415	2416	2417	2418	2419	2420	0003
0006	TEST MODULE NAME 06	2421	2422					
0007	TEST MODULE NAME 07	2423						
0008	TEST MODULE NAME 08	2424						0002
0009	TEST MODULE NAME 09	2425	2426					0003
0011	TEST MODULE NAME 11	2430	2431	2432	2433			0008
0012	TEST MODULE NAME 12	2434	2435					0006
0013	TEST MODULE NAME 13	2436						0012
0015	TEST MODULE NAME 15	2438	2439	2440				0014
0016	TEST MODULE NAME 16	2441	2442					
0017	TEST MODULE NAME 17	2443						
0018	TEST MODULE NAME 18	2444						0002
0020	TEST MODULE NAME 20	2446	2447	2448	2449	2450	2451	0002
0021	TEST MODULE NAME 21	2452						0003
0022	TEST MODULE NAME 22	2453						0019
0023	TEST MODULE NAME 23	2454						
0024	TEST MODULE NAME 24	2455	2456					
0025	TEST MODULE NAME 25	2457	2458	2459				

(01)

Figure 5-9b Audit File-RP

5.5.2.1 Punch File-PF

The Punch File-PF, a card deck which contains the created or updated Population File, is identical in appearance to the Audit File-PF with the exception that there are no trace or diagnostic messages or blank cards. Only cards with information content are punched by POPFM.

5.5.2.2 Punch File-S

The Punch File-S, a card deck which contains the generated JOVIAL source program, is identical in appearance to the Audit File-S with the exception that there are no trace or diagnostic messages or blank cards. Only cards with information content are punched by SJCVS.

5.5.2.3 Punch File-SP

The Punch File-SP, a card deck which contains the updated JOVIAL source program, is identical in appearance to the Audit File-SP with the exception that there are no trace or diagnostic messages or blank cards. Only cards with information content are punched by SOPMM.

5.5.2.4 Punch File-IP

The Punch File-IP, a card deck which contains the resequenced Population File, is identical in appearance to the Audit File-IP with the exception that there are no trace or diagnostic messages or blank cards. Only cards with information content are punched by INIPOP.

5.5.3 Magnetic Tape Output

5.5.3.1 Population File

A Population File is always generated by either of two programming modules, INIPOP and POPFM. The Population File is recorded on magnetic tape for subsequent processing.

5.5.3.2 Source Program File

A Source Program File is always generated by SJCVS. This file contains the generated JOVIAL test program and is submitted directly to the operating system for compilation and execution.

TABLE OF CONTENTS

for

APPENDIX 1

Table of Contents for Appendix 1 (Continued)

Table of Contents for Appendix 1 (Continued)

APPENDIX 1

USAGE INSTRUCTIONS

Appendix 1 describes on the following pages usage instructions for each function on each computer. Usage instructions depict the status of the hardware configuration before the run (INPUT) and after the run (OUTPUT). All input/output considerations are fully described for both the INPUT stage and the OUTPUT stage. In addition, the exact form of an input card deck necessary to invoke the function is provided.

Each JCVS Usage Form contains the JCVS function to be performed, the computer, the operating philosophy and the program stage. All input/output functions and devices are specified over the six boxes on each form. On the top of each of these boxes is the logical system name associated with the input/output device.

For example, on the 6400 the logical tape designations are TAPE1, TAPE2, and TAPE3; the logical card input designation is INPUT, etc.

For those input/output units that are to be active for the current function, some indication of their participation is indicated. For those tape units that are to contain a switch tape for the subsequent processing, the word SCRATCH is placed at the bottom of the appropriate box; for those tape units that are to contain a JCVS input or output file, the file-name is placed in the bottom of the box; and for those tape units whose participation is not required, a N/A (not applicable) is placed at the bottom of the box.

In all cases, a job deck will be submitted through the card input unit which should be empty at the termination of the run. The printed output unit will always contain a standard form and standard carriage control tape and will contain the various audit files at the termination of a run. The card output unit will contain any punched output originating from any of the runs.

A complete description of the job deck structure required to process the function is given on each INPUT stage usage form. The (1) below the words JOB DECK STRUCTURE indicates column 1 of each card.

Logical Unit Names

The logical unit names for each computer will now be stated:

Units \ Configuration	CDC-6400	UNI-1108	GE-635	IBM 360-50
Card Input	INPUT	Card Reader Eighty	A1	SYS001
Card Output	PUNCH	Card Punch Eighty	A5	SYS003
Printed Output	OUTPUT	Printer	A2	SYS002
Tape Number 1	TAPE1	UNISERVO A	A3	SYS004
Tape Number 2	TAPE2	UNISERVO B	A4	SYS005
Tape Number 3	TAPE3	UNISERVO C	A6	SYS007

Special Cards

Certain configurations contain one or two special cards that act as end of record or end of file cards. The following table gives a list of these cards together with the characters that signify the EOR or EOF functions.

End \ Configuration	CDC-6400	UNI-1108	GE-635	IBM 360-50
EOR	7,8,9 punch Column 1	No Entry	$bbbbbbENDJOB	No Entry
EOF	6,7,8,9 punch Column 1	@bFIN Column 1-5	***EOF	1*

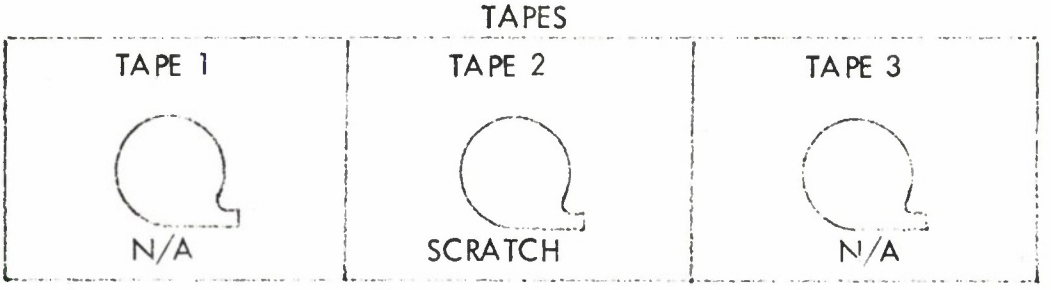
JCVS USAGE FORM

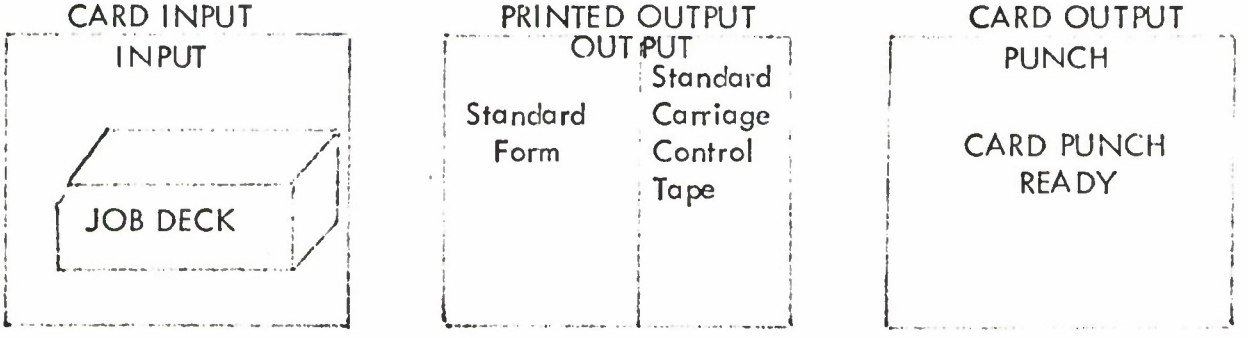
Function: POPFM 1

Computer: CDC - 6400

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA

JOB, 93007, 10, 10, 35000. POPFM 1 CG

REQUEST, TAPE 2, HI. (ASSIGN/RING)

REWIND (TAPE 2)

COBOL (LXRM).

LGO.

(End of Record Card)
(CCBOL Source Porgram Deck POPFM
(End of Record Card)
(Control Card - PF)
(Current File - PF Deck)
(End of File Card)

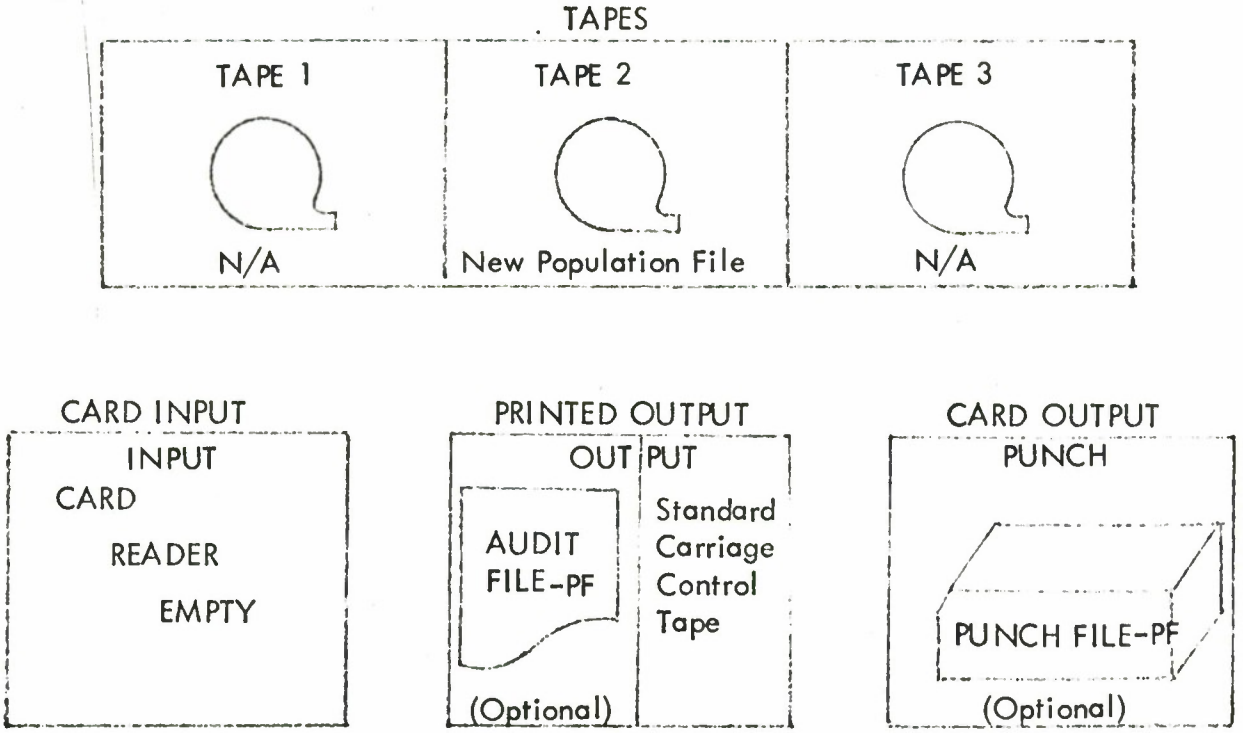
JCVS USAGE FORM

Function: POPFM 1

Computer: CDC - 6400

Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT



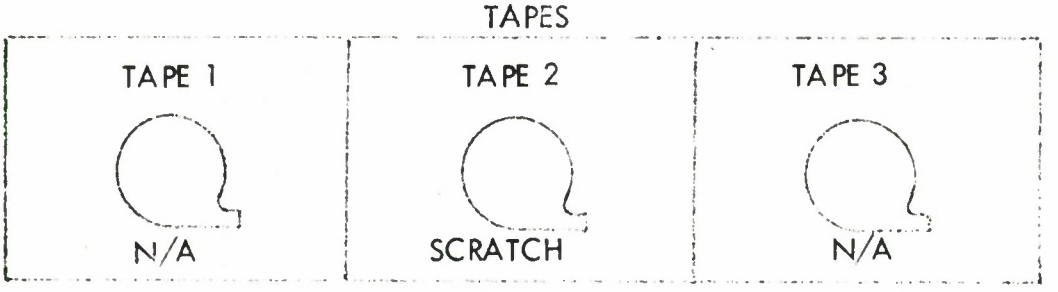
JCVS USAGE FORM

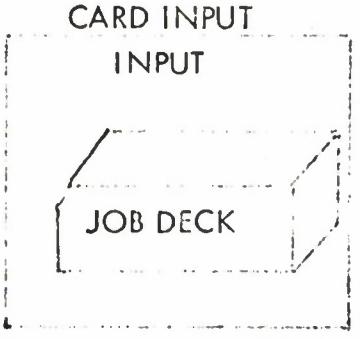
Function: POPFM 1

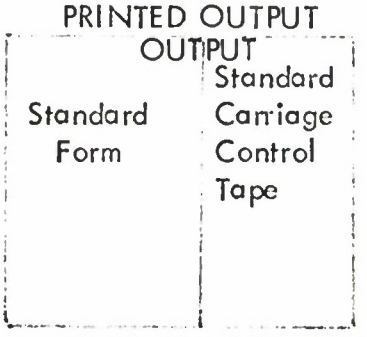
Computer: CDC - 6400

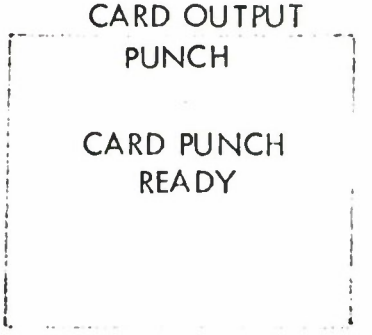
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT









JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA

JOB, 93007, 10, 10, 35000. POPFM 1 LG

REQUEST, TAPE 2, HI. (ASSIGN/RING)

REWIND (TAPE 2)

LOAD (INPUT)

EXECUTE (POPFM)

(End of Record Card)
(Binary Program Deck - POPFM)
(End of Record Card)
(Control Card - PF)
(Current File - PF Deck)
(End of File Card)

JCVS USAGE FORM

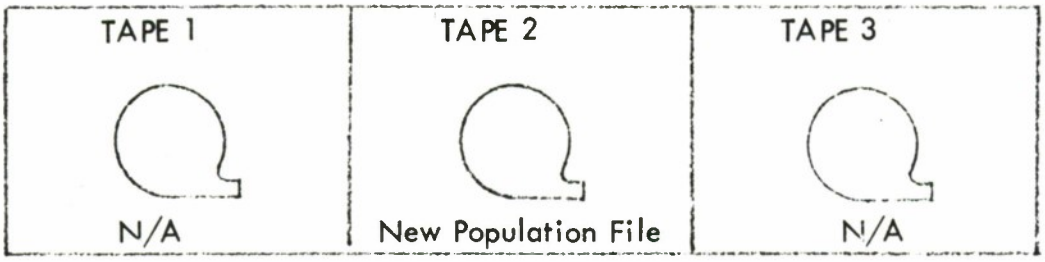
Function: POPFM 1

Computer: CDC - 6400

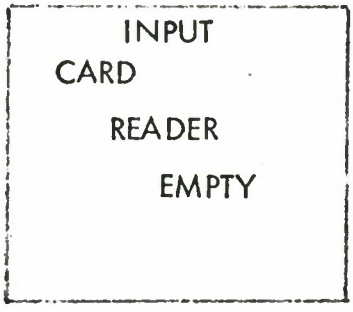
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

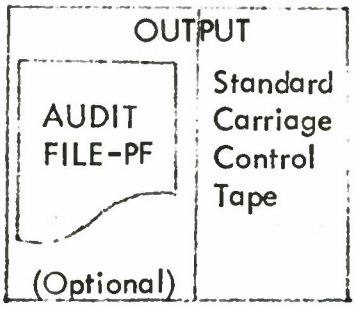
TAPES



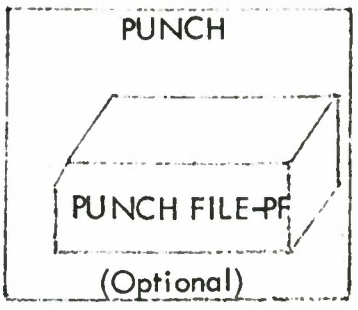
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

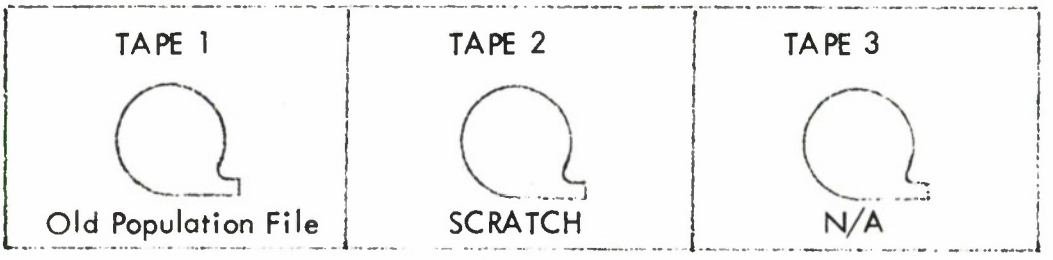
Function: POPFM 2

Computer: CDC - 6400

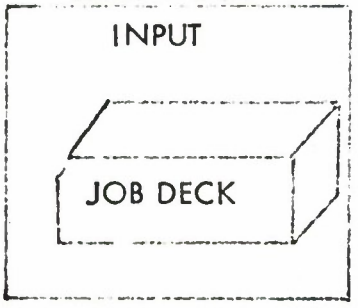
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

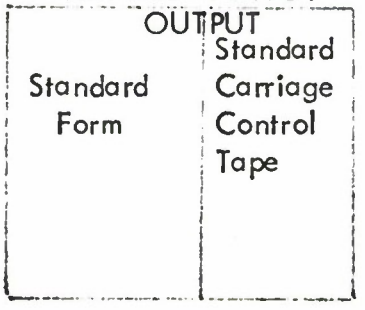
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA

JOB, 93007, 10, 10, 35000. POPFM 2 CG

REQUEST, TAPE 1, HI. (REEL/NO RING)

REQUEST, TAPE 2, HI. (ASSIGN/RING)

REWIND (TAPE 1).

REWIND (TAPE 2).

COBOL (LXRN).

LGO

(End of Record Card)
(COBOL Source Program Deck - POPFM)
(End of Record Card)
(Control Card - PF)
(Current File - PF Deck)
(End of File Card)

JCVS USAGE FORM

Function: POPFM 2

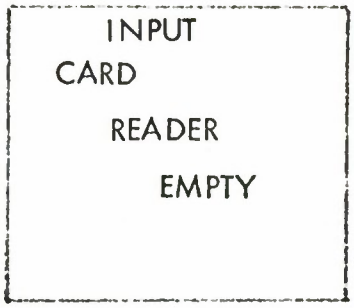
Computer: CDC - 6400

Operating Philosophy: Compile Source Program and Go

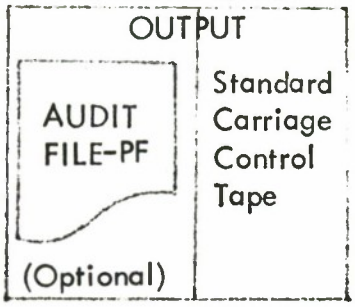
Stage: OUTPUT

TAPES

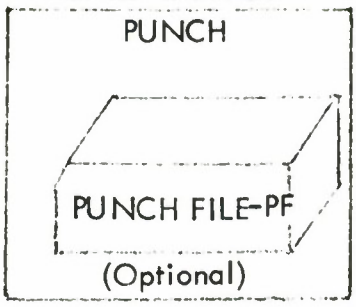
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



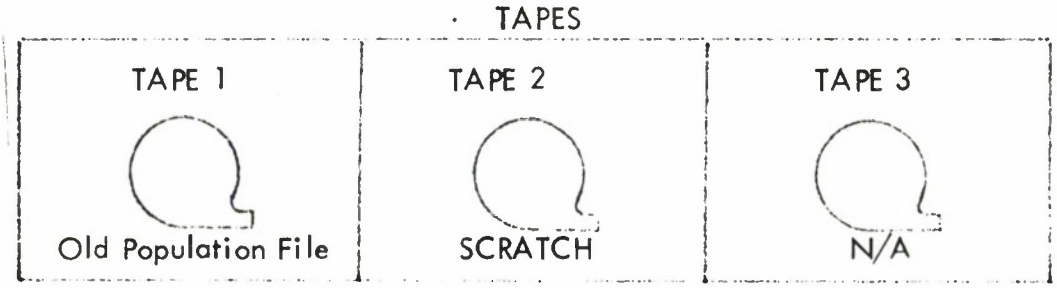
JCVS USAGE FORM

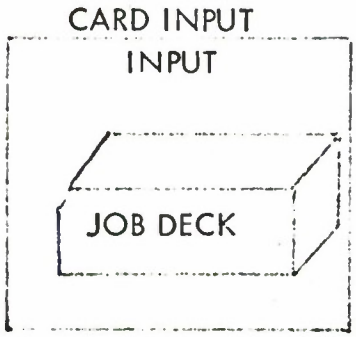
Function: POPFM 2

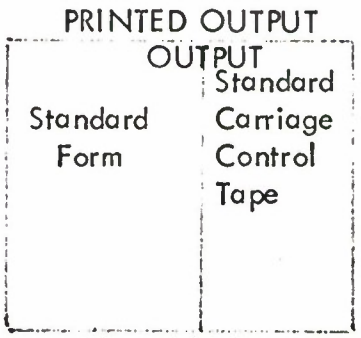
Computer: CDC - 6400

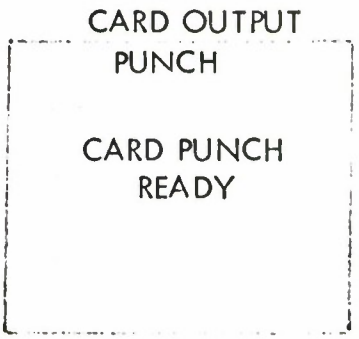
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT









JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA

JOB, 93007, 10, 10, 35000. POPFM 2 LG

REQUEST, TAPE 1, HI. (REEL/NO RING)

REQUEST, TAPE 2, HI. (ASSIGN/RING)

REWIND (TAPE 2).

LOAD (INPUT)

EXECUTE (POPFM)

(End of Record Card)
(Binary Program Deck - POPFM)
(End of Record Card)
(Control Card - PF)
(Current File - PF Deck)
(End of File Card)

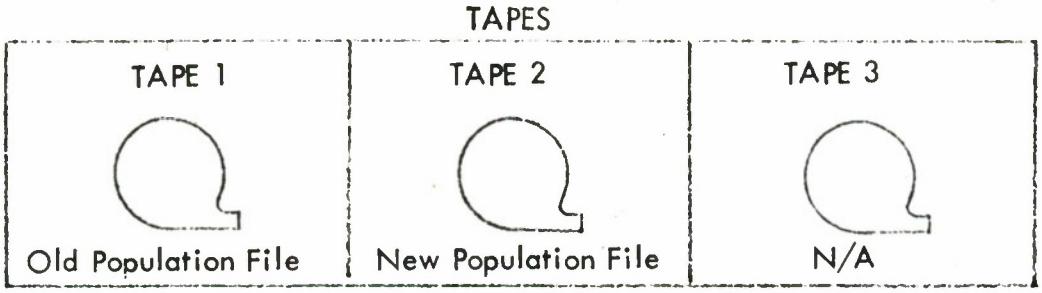
JCVS USAGE FORM

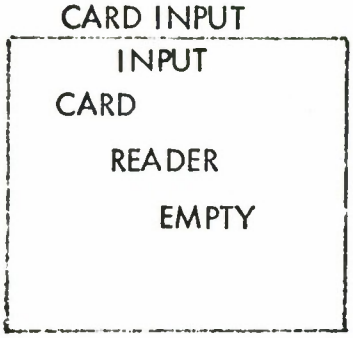
Function: POPFM 2

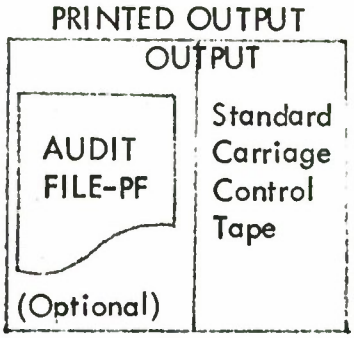
Computer: CDC - 6400

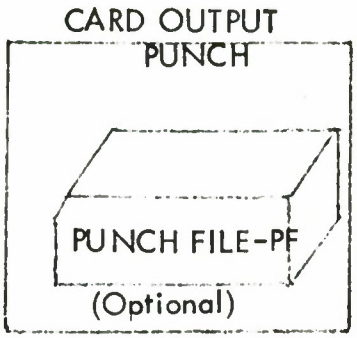
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT









JCVS USAGE FORM

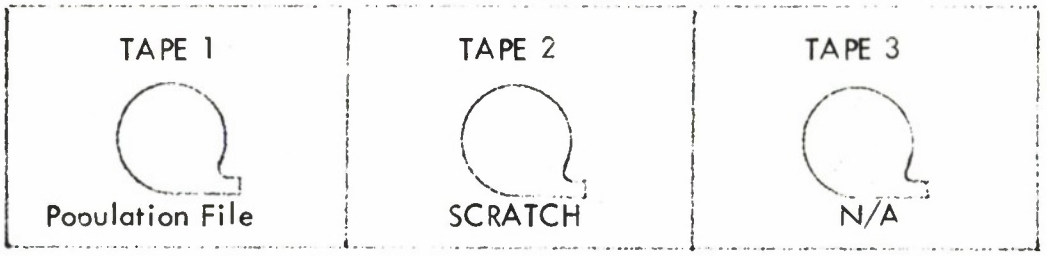
Function: SELECT

Computer: CDC - 6400

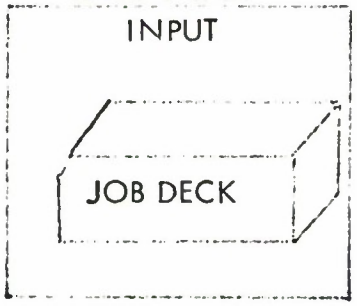
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

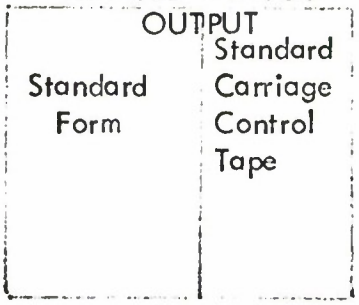
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA

JOB, 93007, 10, 10, 35000. SELECT

REQUEST, TAPE 1, HI. (REEL/NO RING)

REQUEST, TAPE2, HI. (ASSIGN/RING)

REWIND (TAPE 2).

COBOL (LXRM).

LGO.

(End of Record Card)
(COBOL Source Program Deck - SJCVS)
(End of Record Card)
(Control Card - S)

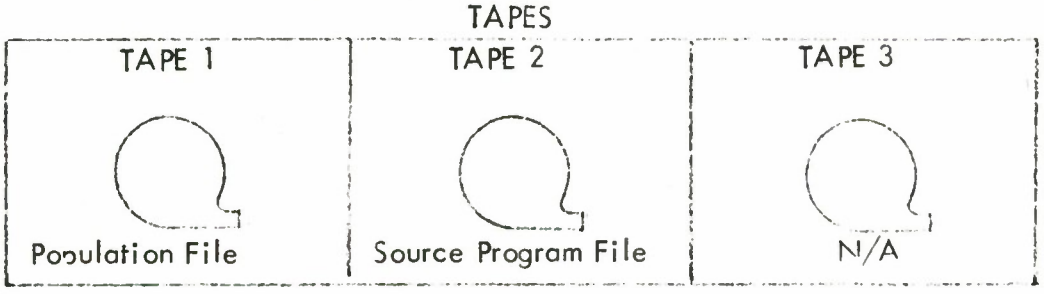
(Test Selection. File Deck)
(End of File Card)

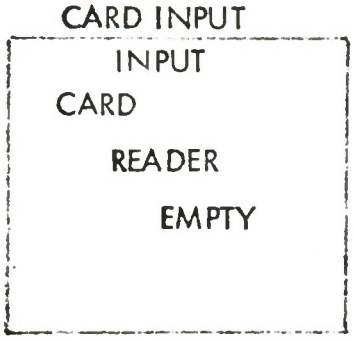
JCVS USAGE FORM

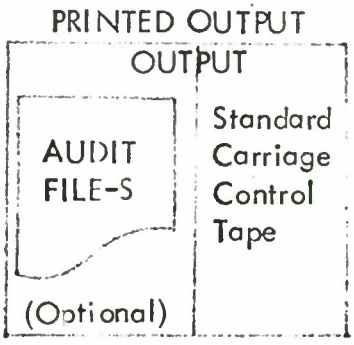
Function: SELECT
Computer: CDC - 6400

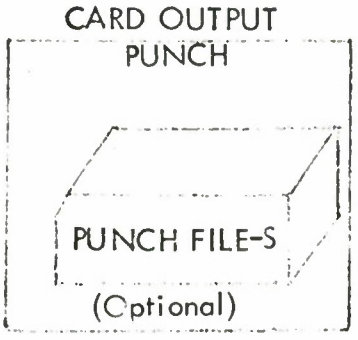
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT









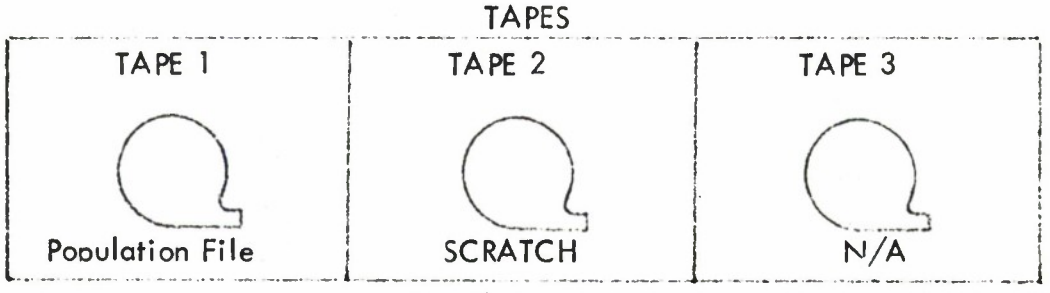
JCVS USAGE FORM

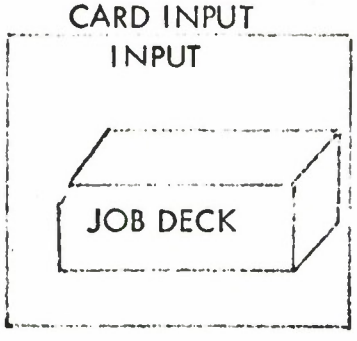
Function: SELECT

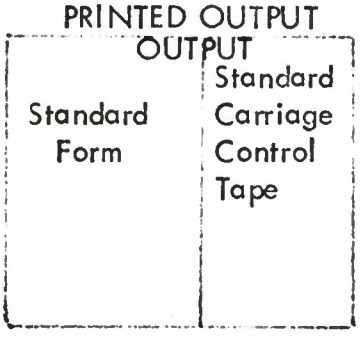
Computer: CDC - 6400

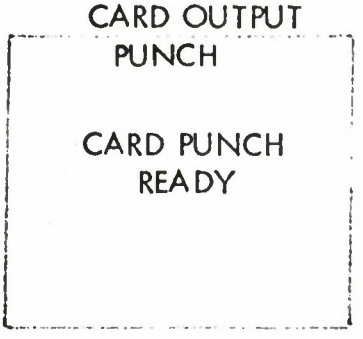
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT









JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA

JOB, 93007, 10, 10, 35000. SELECT

REQUEST, TAPE 1, HI. (REEL/NO RING)

REQUEST, TAPE 2, HI. (ASSIGN/RING)

REWIND, (TAPE 2).

LOAD (INPUT)

EXECUTE (SJCVS)

(End of Record Card)
(Binary Program Deck - SJCVS)
(End of Record Card)
(Control Card - S)

(Test Selection File Deck)
(End of File Card)

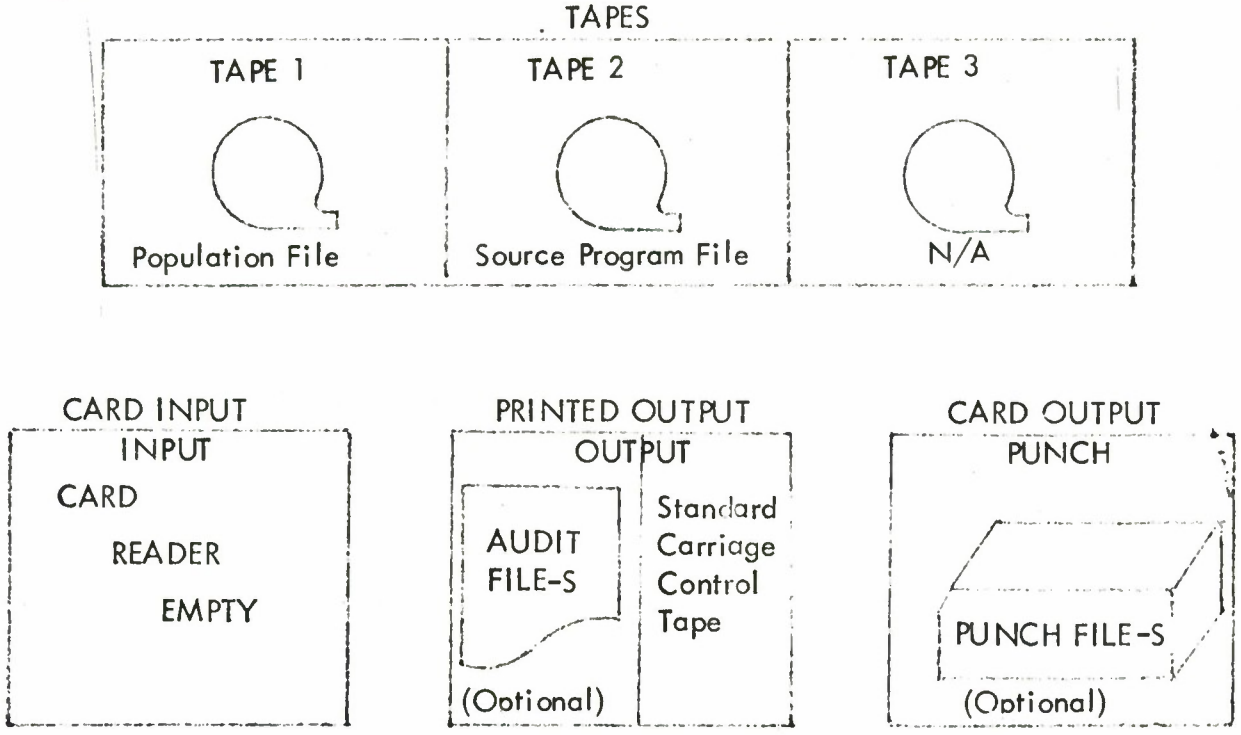
JCVS USAGE FORM

Function: SELECT

Computer: CDC - 6400

Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT



JCVS USAGE FORM

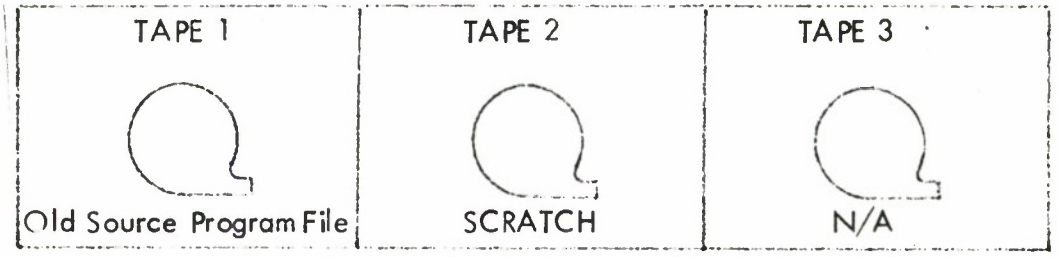
Function: SOPMM

Computer: CDC - 6400

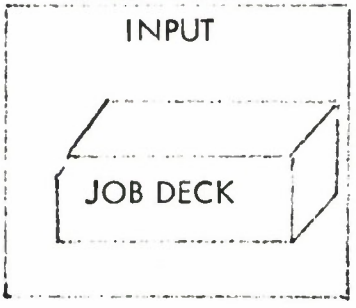
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

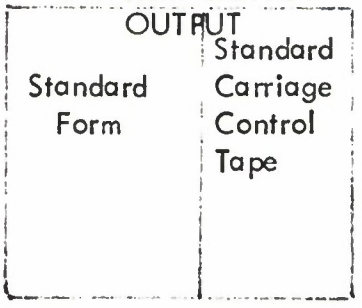
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA.

JOB, 93007, 10, 10, 35000. SOPMM

REQUEST, TAPE 1, HI. (REEL/NO RING)

REQUEST, TAPE 2, HI. (ASSIGN/RING)

REWIND (TAPE 2).

COBOL (LXRM).

LGO.

(End of Record Card)
(COBOL Source Program Deck - SOPMM)
(End of Record Card)
(Control Card - SP)
(Current File - SP Deck)
(End of File Card)

JCVS USAGE FORM

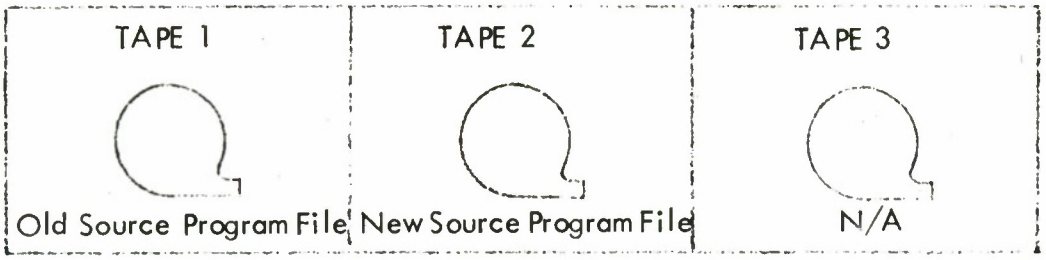
Function: SOPMM

Computer: CDC - 6400

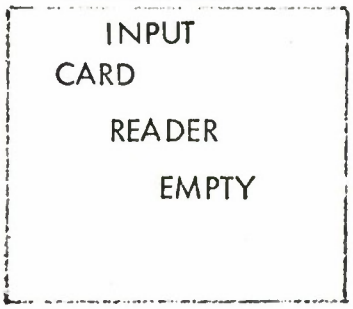
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

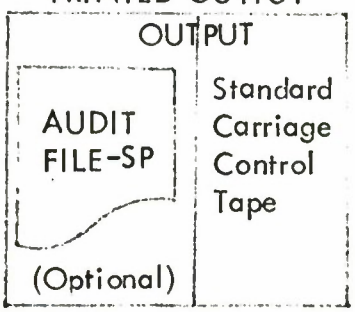
TAPES



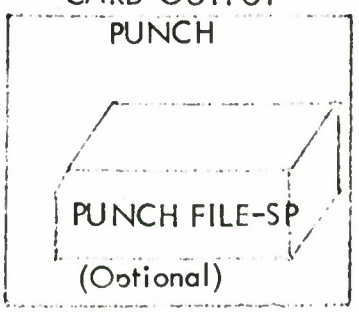
CARD INPUT



PRINTED OUTPUT

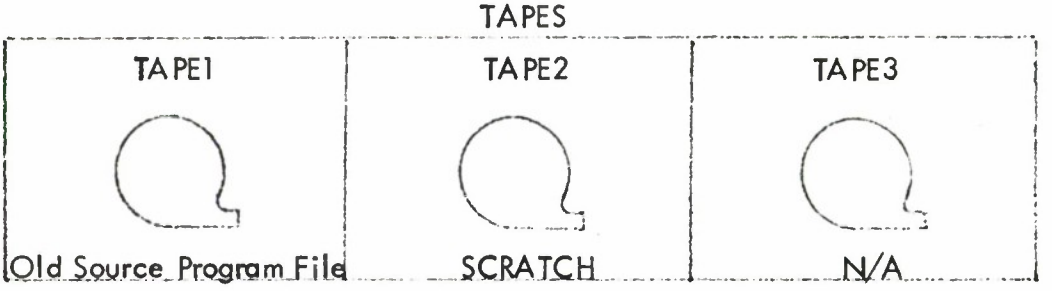


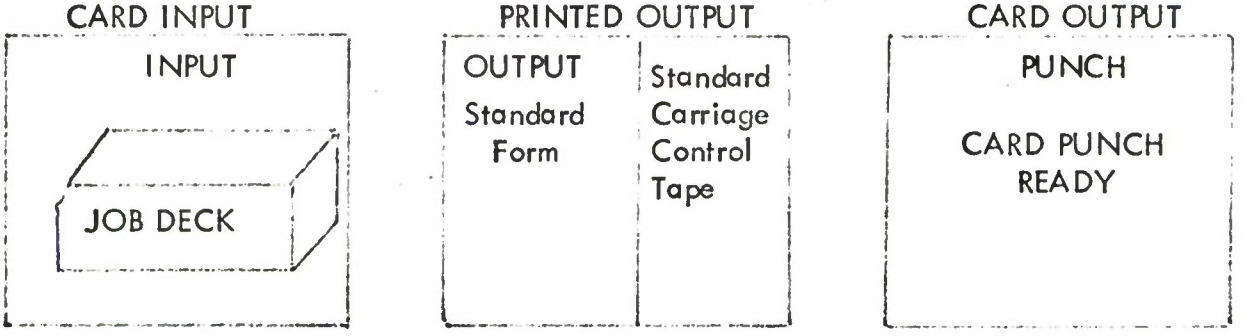
CARD OUTPUT



JCVS USAGE FORM

Function: SOPMM
Computer: CDC-6400
Operating Philosophy: Load Binary Deck and Go
Stage: INPUT





JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA.
JOB,93007,10,10,35000. SOPMM
REQUEST, TAPE1, HI. (REEL/NORING)
REQUEST, TAPE2, HI. (ASSIGN/RING)
REWIND (TAPE2)

LOAD (INPUT)
EXECUTE (SOPMM)

(End of Record Card)
(Binary Program Deck-SOPMM)
(End of Record Card)
(Control Card-SP)
(Current File-SP Deck)
(End of File Card)

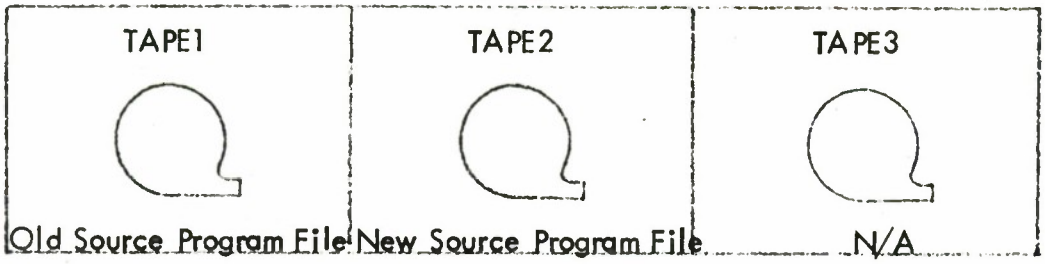
JCVS USAGE FORM

Function: SOPMM
Computer: CDC-6400

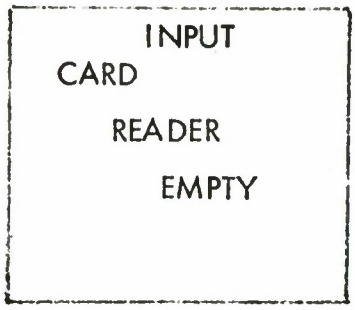
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

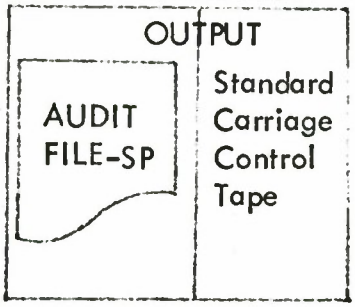
TAPES



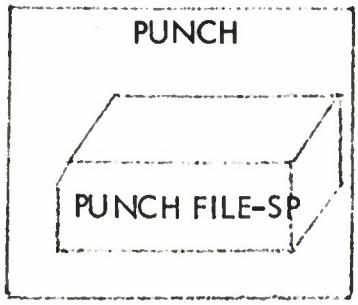
CARD INPUT



PRINTED OUTPUT

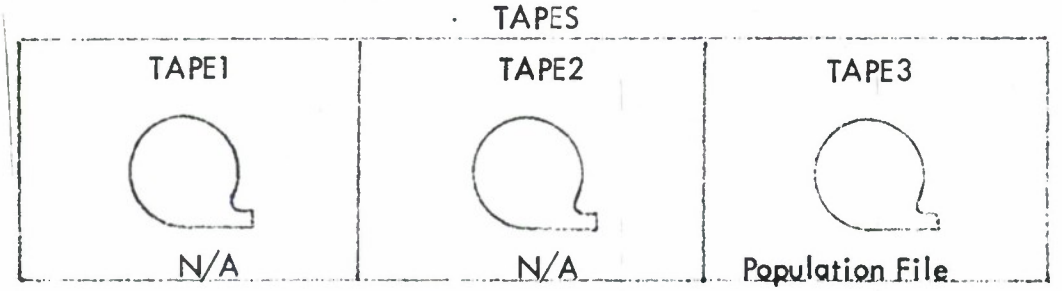


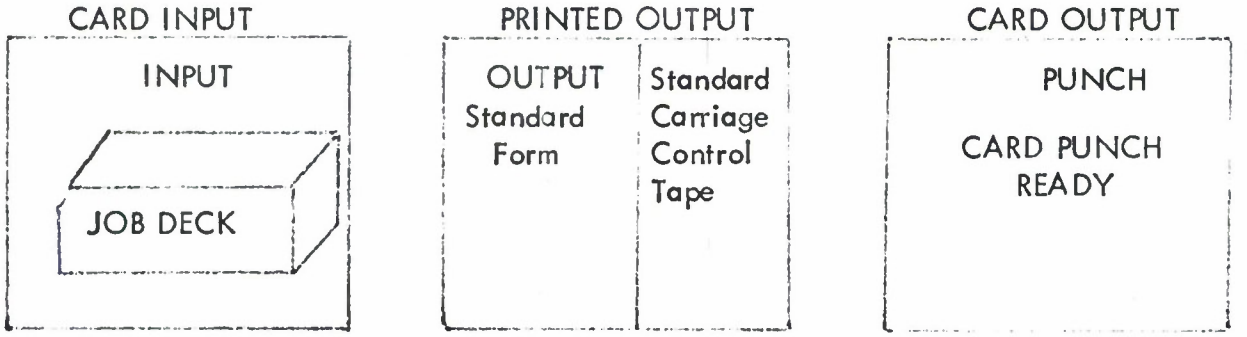
CARD OUTPUT



JCVS USAGE FORM

Function: JCVSRP
Computer: CDC-6400
Operating Philosophy: Compile Source Program and Go
Stage: INPUT





JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA.
JOB, 93007, 10, 10, 35000. JCVSRP.
REQUEST, TAPE3, HI. (XXXX/NORING)

XXXX = Population File Reel Number

COBOL (LXRM).
LGO.

(End of Record Card)
(COBOL Source Program Deck - JCVSRP)
(End of Record Card)
(Control Card-RP)
(End of File Card)

JCVS USAGE FORM

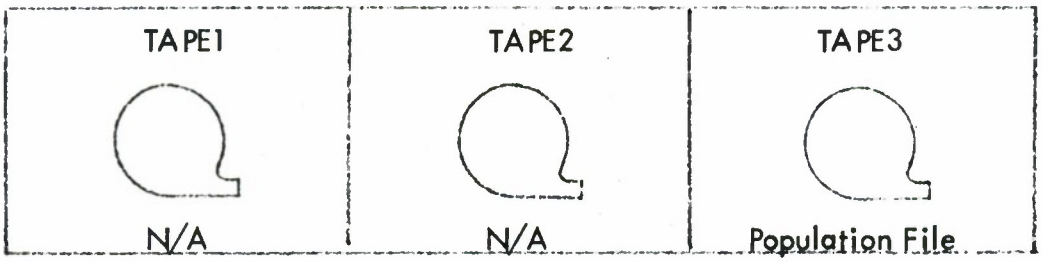
Function: JCVSRP

Computer: CDC-6400

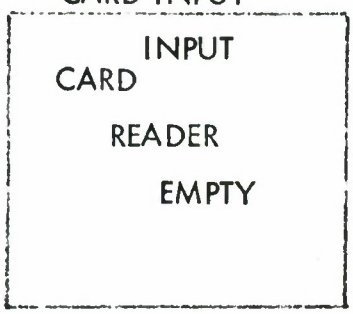
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

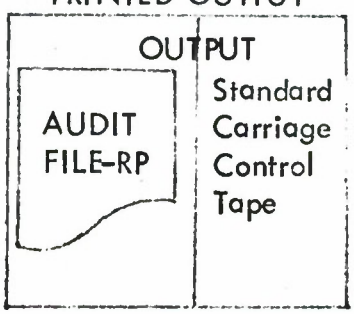
TAPES



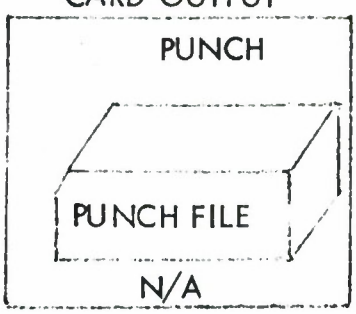
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



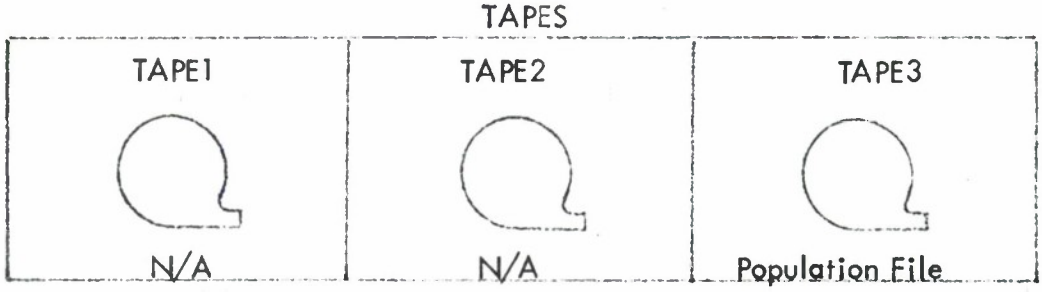
JCVS USAGE FORM

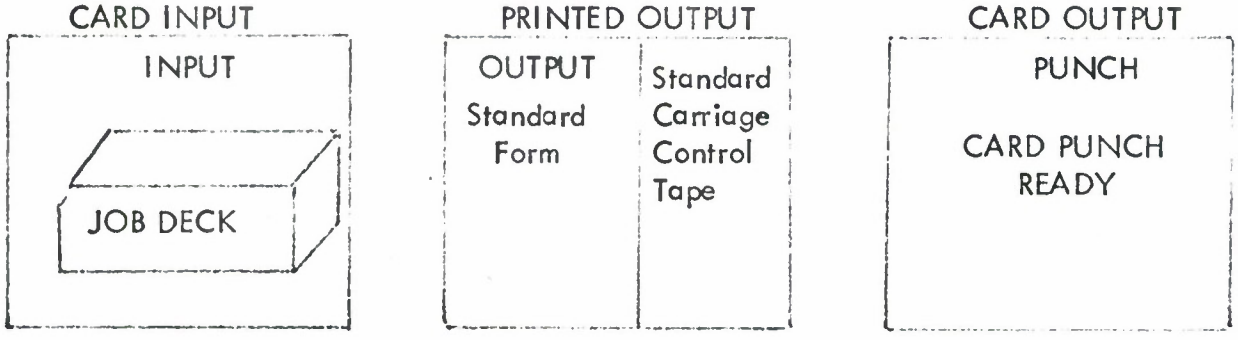
Function: JCVSRP

Computer: CDC-6400

Operating Philosophy: Load Binary Deck and Go

Stage: INPUT





JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA.
JOB, 93007, 10, 10, 35000.
REQUEST, TAPE3, HI. (XXXX/NORING)

XXXX = Population File Reel Number

LOAD (INPUT)
EXECUTE (JCVSRP)

(End of Record Card)
(Binary Program Deck - JCVSRP)
(End of Record Card)
(Control Card - RP)
(End of File Card)

JCVS USAGE FORM

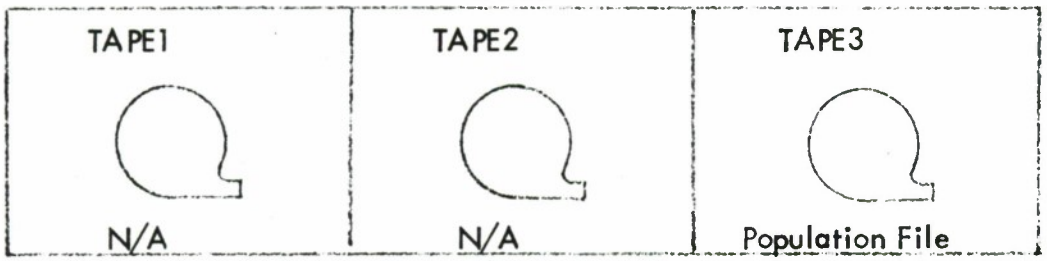
Function: JCVSRP

Computer: CDC-6400

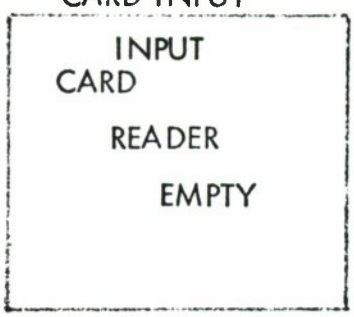
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

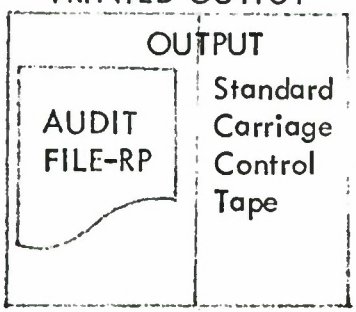
TAPES



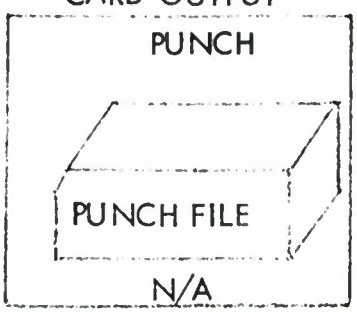
CARD INPUT



PRINTED OUTPUT



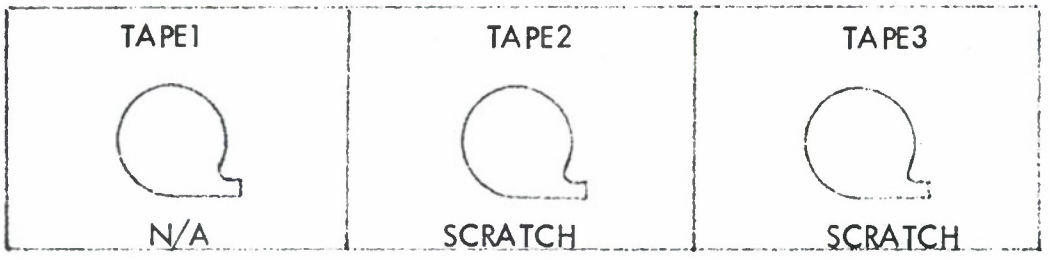
CARD OUTPUT



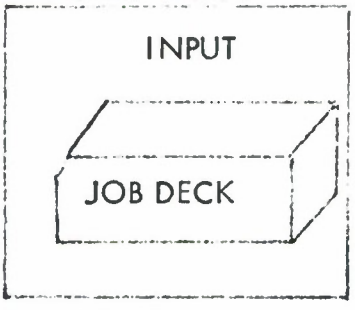
JCVS USAGE FORM

Function: INIPOP1
Computer: CDC-6400
Operating Philosophy: Compile Source Program and Go
Stage: INPUT

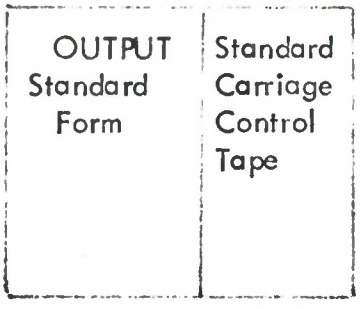
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

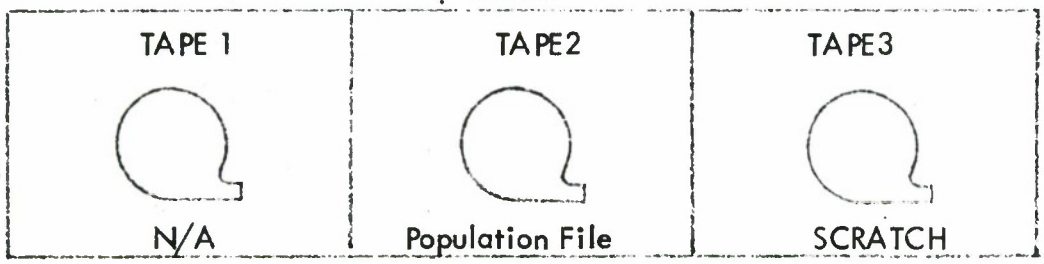
```
SEQUENCE, 14156, SMA.
JOB, 93007, 10, 10, 35000.  INIPOP1.

COBOL (LXRM).
LGO.
```

(End of Record Card)
(COBOL Source Program Deck - INIPOP)
(End of Record Card)
(Control Card - IP)
(Current File - PF Deck)
(End of File Card)

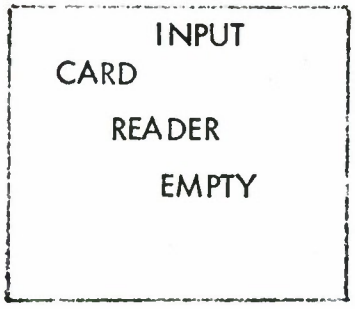
JCVS USAGE FORM

Function: INIPOP1
Computer: CDC-6400
Operating Philosophy: Compile Source Program and Go
Stage: OUTPUT

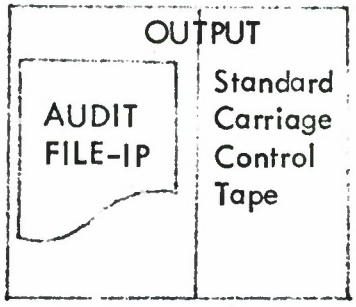
TAPES



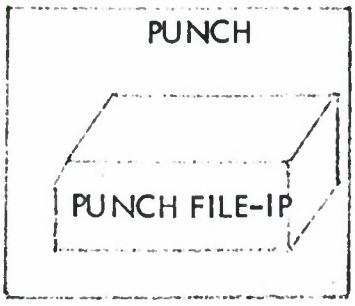
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

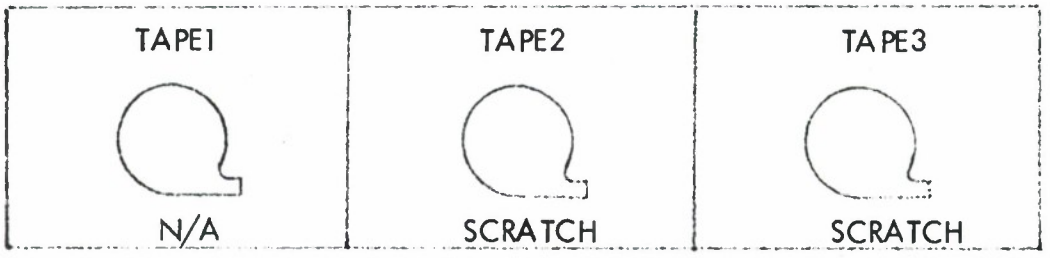
Function: INIPOP1

Computer: CDC-6400

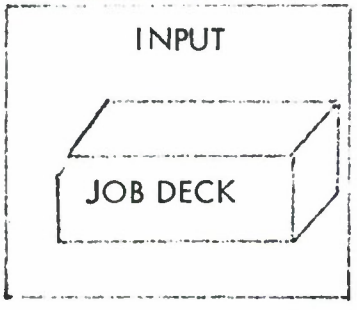
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT

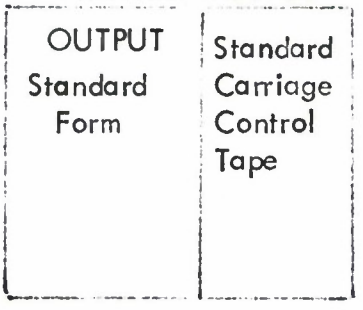
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA.
JOB, 93007, 10, 10, 35000.

LOAD (INPUT)
EXECUTE (INIPOP)

(End of Record Card)
(Binary Program Deck - INIPOP)
(End of Record Card)
(Control Card-IP)
(Current File-PF Deck)
(End of File Card)

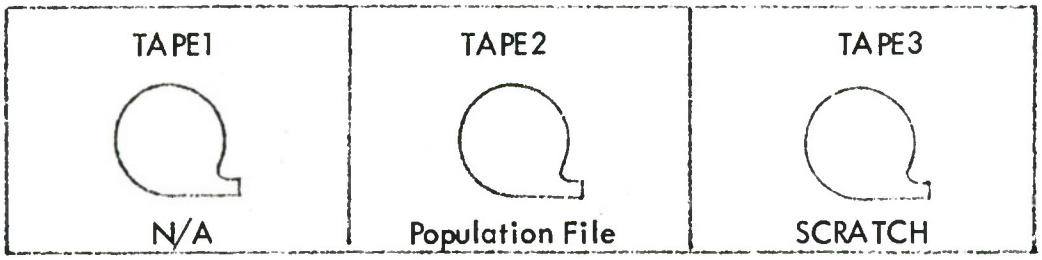
JCVS USAGE FORM

Function: INIPOP1
Computer: CDC-6400

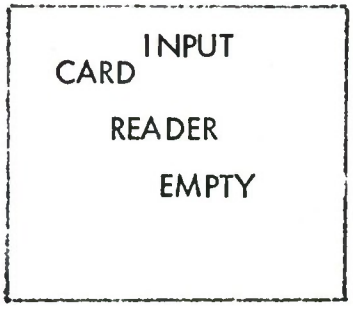
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

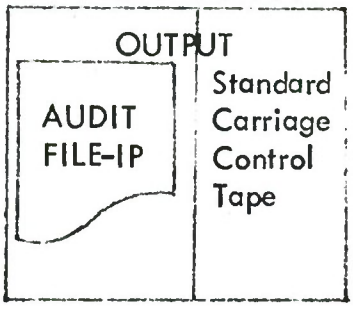
TAPES



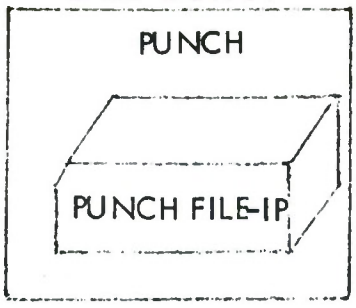
CARD INPUT



PRINTED OUTPUT

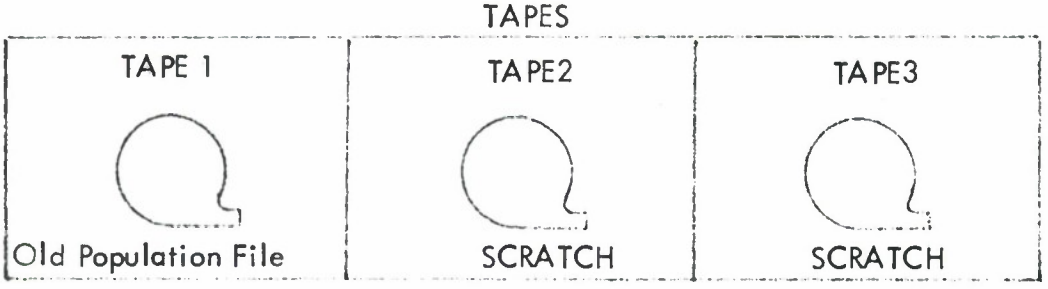


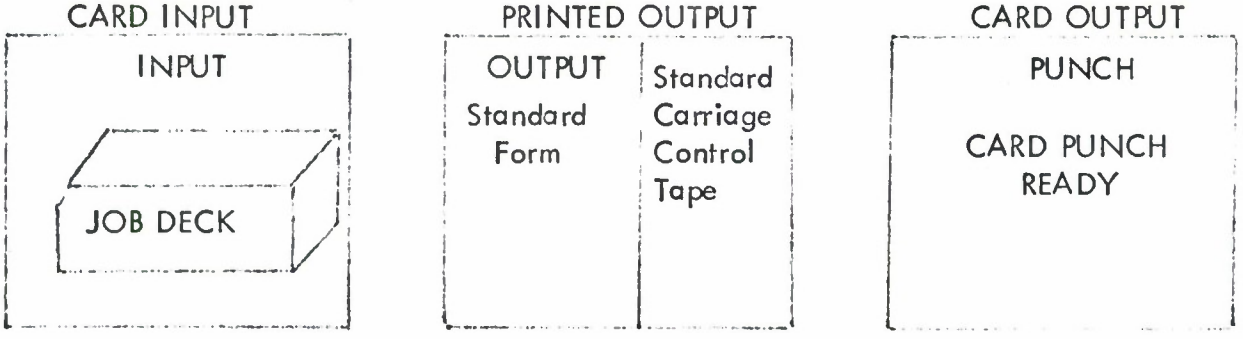
CARD OUTPUT



JCVS USAGE FORM

Function: INIPOP2
Computer: CDC-6400
Operating Philosophy: Compile Source Program and Go
Stage: INPUT





JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA.
JOB, 93007, 10, 10, 35000. INIPOP2.
REQUEST, TAPE1, HI. (XXXX/NORING)

XXXX = Population File Reel Number

COBOL (LXRM).
LGO.

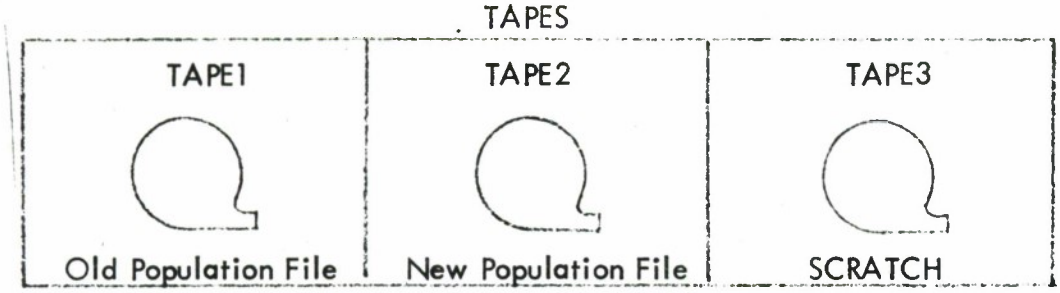
(End of Record Card)
(COBOL Source Program Deck - INIPOP)
(End of Record Card)
(Control Card-IP)
(End of File Card)

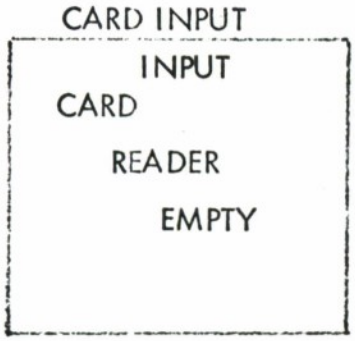
JCVS USAGE FORM

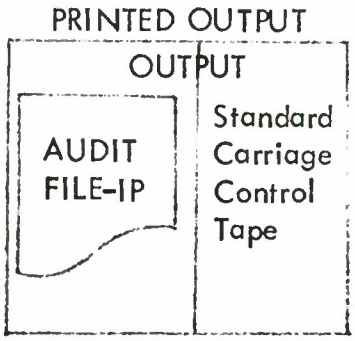
Function: INIPOP2
Computer: CDC-6400

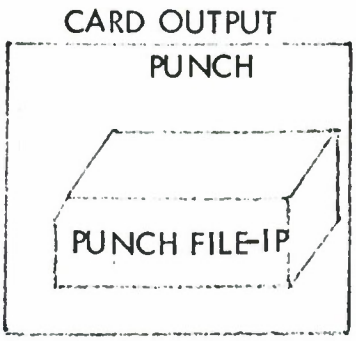
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT





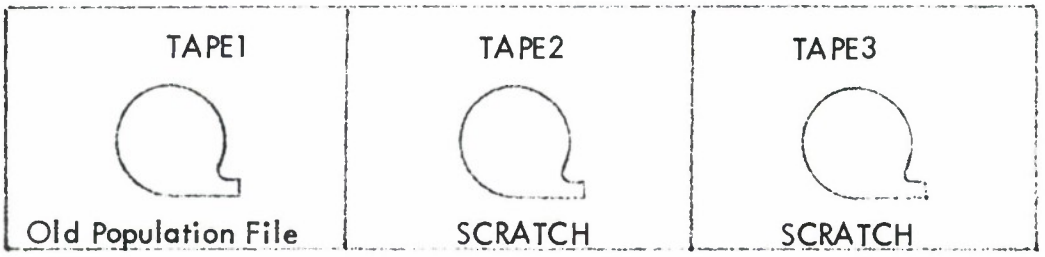




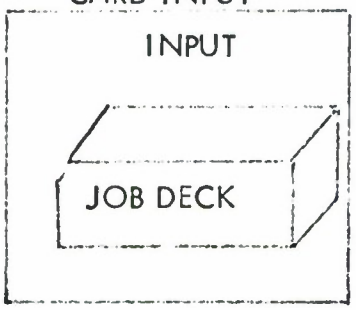
JCVS USAGE FORM

Function: INIPOP2
Computer: CDC-6400
Operating Philosophy: Load Binary Deck and Go
Stage: INPUT

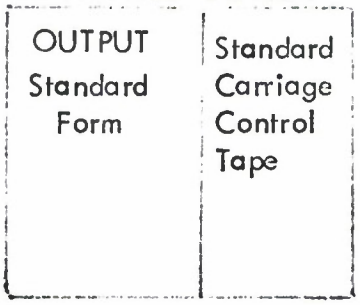
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

SEQUENCE, 14156, SMA.
JOB, 93007, 10, 10, 35000. INIPOP2.
REQUEST, TAPE1, HI. (XXXX/NORING)

XXXX = Population File Reel Number

LOAD (INPUT)
EXECUTE (INIPOP)

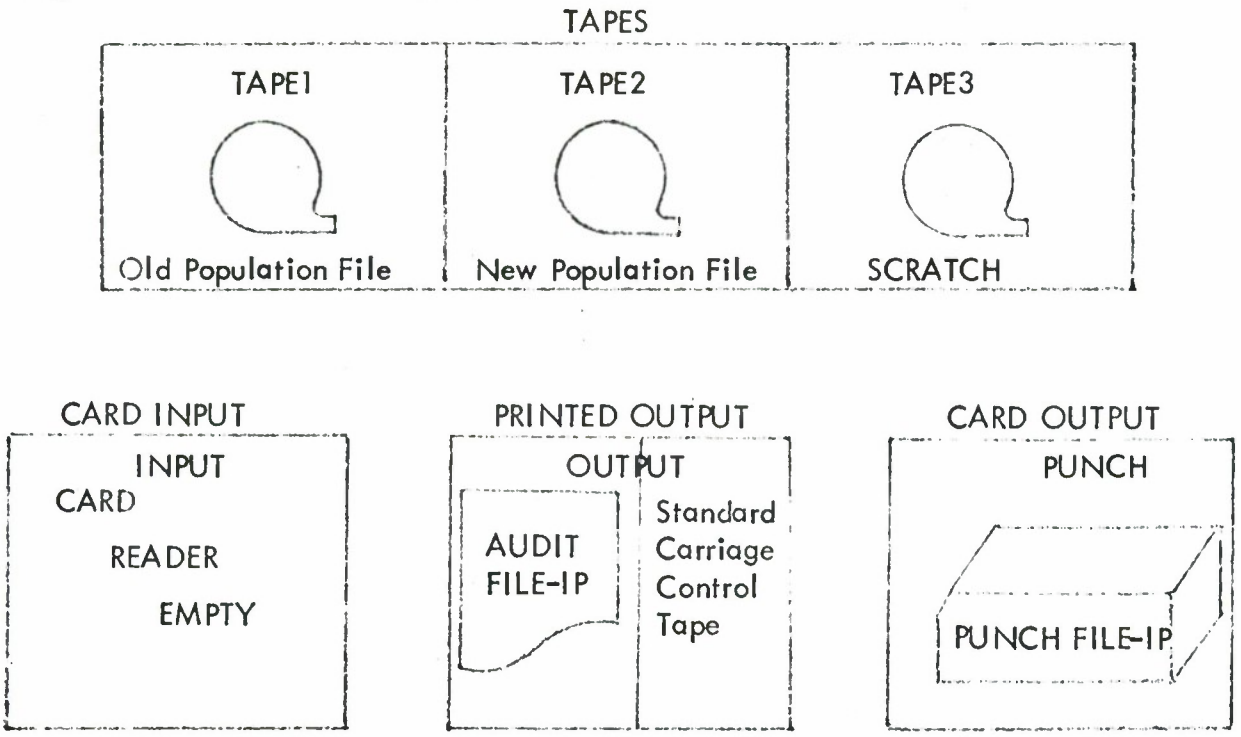
(End of Record Card)
(Binary Program Deck - INIPOP)
(End of Record Card)
(Control Card-IP)
(End of File Card)

JCVS USAGE FORM

Function: INIPOP2
Computer: CDC-6400

Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT



JCVS USAGE FORM

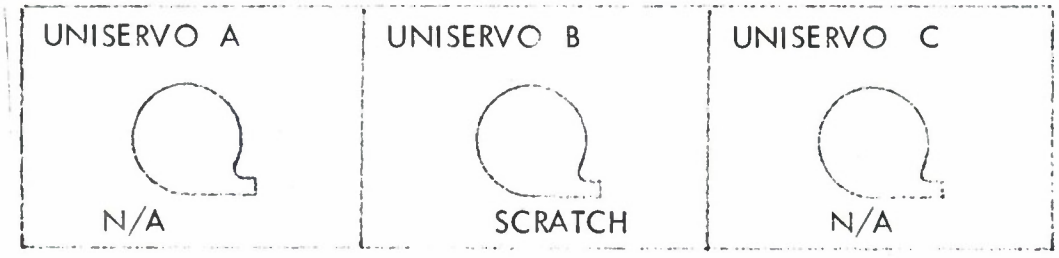
Function: POPFM1

Computer: UNIVAC - 1108

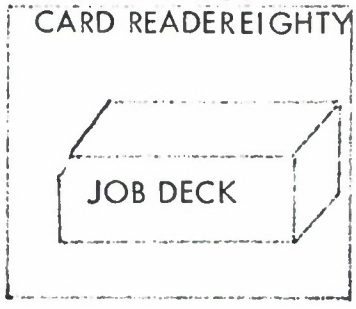
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

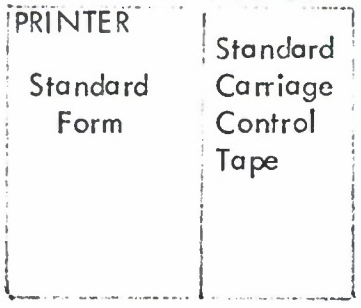
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

@ RUN 1 POPFM1, DDCG, 5, 300

@ ASG B = SAVE

@ BREI COB POPFM1

(COBOL Source Porgram Deck - POPFM)

@ XQT POPFM1

(Control Card - PF)
(Current File - PF Deck)

@ FIN

JCVS USAGE FORM

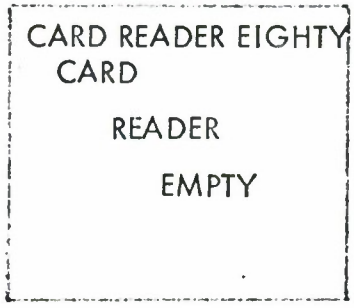
Function: POPFM1
Computer: UNIVAC - 1108

Operating Philosophy: Compile Source Program and Go

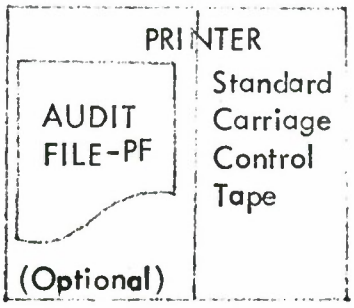
Stage: OUTPUT

TAPES

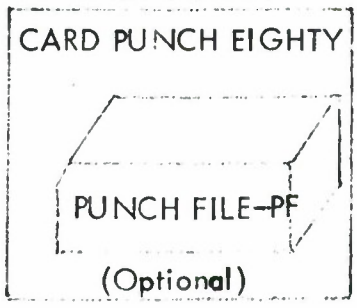
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

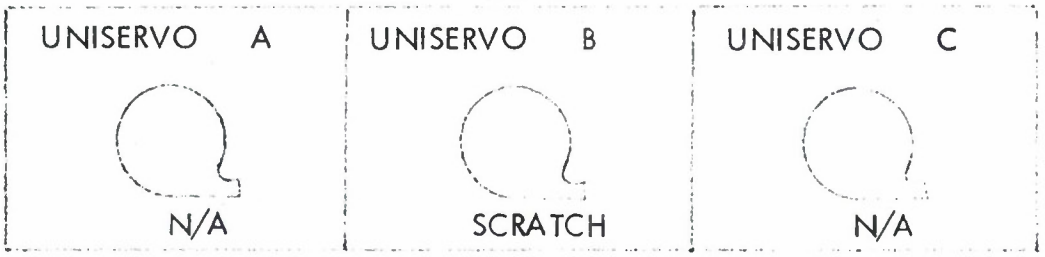
Function: POPFM1

Computer: UNIVAC - 1108

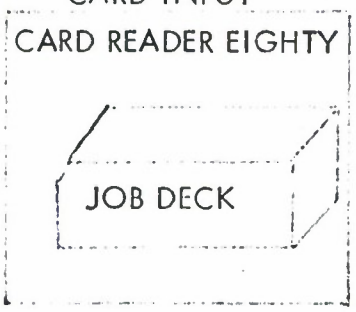
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT

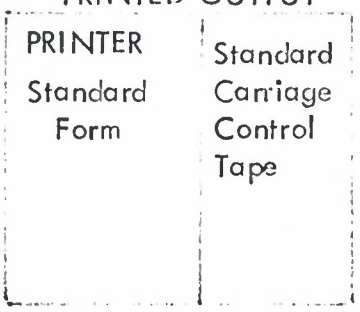
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)
@ RUN 1 POPFM1, DOLG,5,300
@ ASG B = SAVE

(Binary Program Deck - POPFM)

@ XQT,POPFM1

(Control Card - PF)
(Current File - PF Deck)

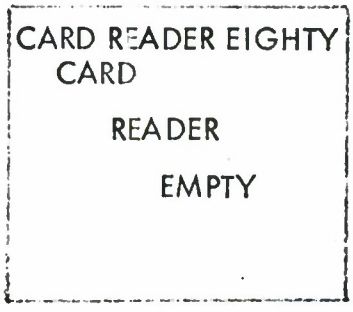
@ FIN

JCVS USAGE FORM

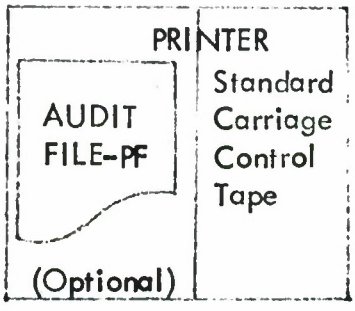
Function: POPFM1
Computer: UNIVAC - 1108
Operating Philosophy: Load Binary Deck and Go
Stage: OUTPUT

TAPES

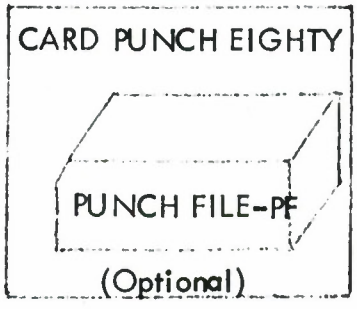
CARD INPUT



PRINTED OUTPUT



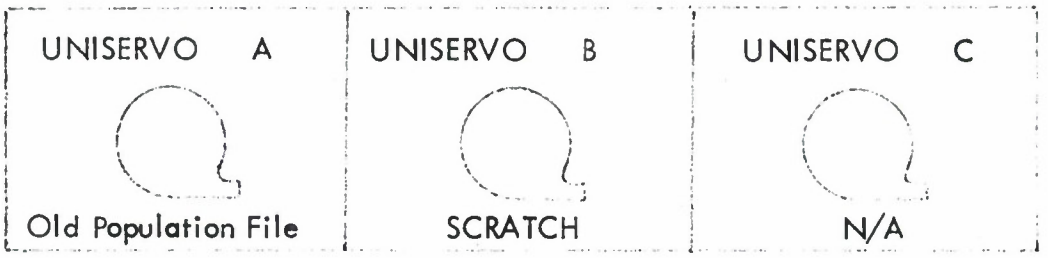
CARD OUTPUT



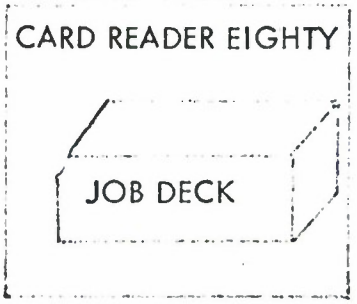
JCVS USAGE FORM

Function: POPFM2
Computer: UNIVAC -1108
Operating Philosophy: Compile Source Program and Go
Stage: INPUT

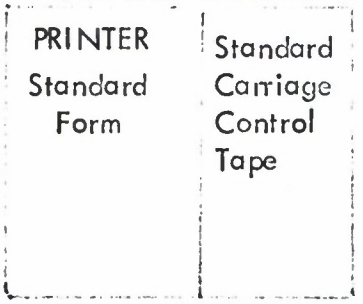
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)
@ RUN 1 POPFM2,DDLG,5,300
@ ASG,B,A = XXXX XXXX = POPFILE1 reel number
@ BREI COB POPFM2

(COBOL Source Program Deck - POPFM)

@ XQT POPFM2

(Control Card - PF)
(Current File - PF Deck)

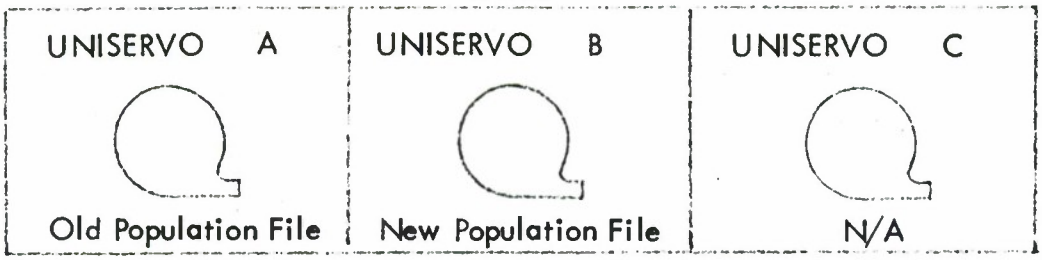
@ FIN

JCVS USAGE FORM

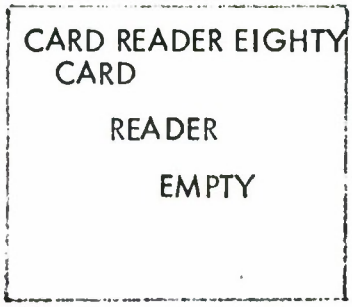
Function: POPFM2
Computer: UNIVAC - 1108

Operating Philosophy: Compile Source Program and Go
Stage: OUTPUT

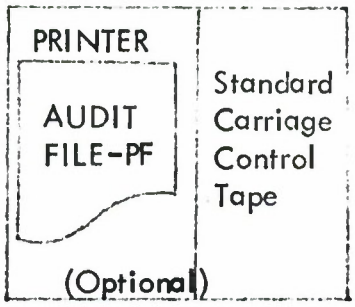
TAPES



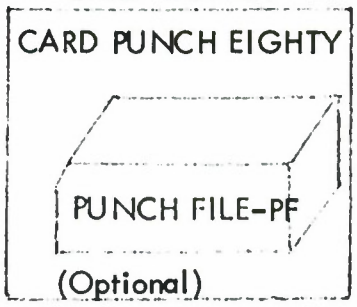
CARD INPUT



PRINTED OUTPUT



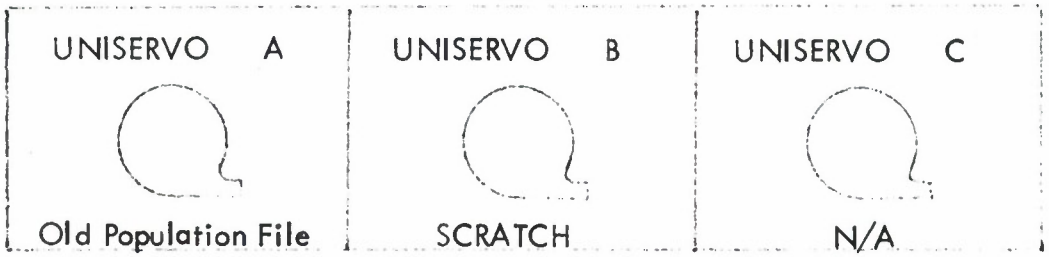
CARD OUTPUT



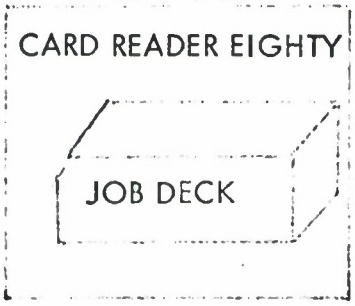
JCVS USAGE FORM

Function: POPFM2
Computer: UNIVAC - 1108
Operating Philosophy: Load Binary Deck and Go
Stage: INPUT

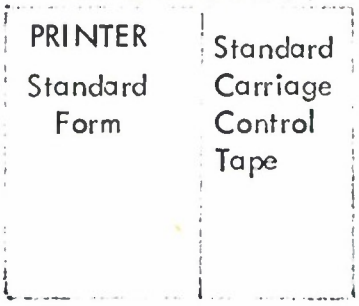
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

@ RUN 1 POPFM2,DDLG,5,300
@ ASG B A = XXXX

XXXX = POPFILE1 reel number

(Binary Program Deck - POPFM)

@ XQT POPFM2

(Control Card - PF)
(Current File - PF Deck)

@ FIN

JCVS USAGE FORM

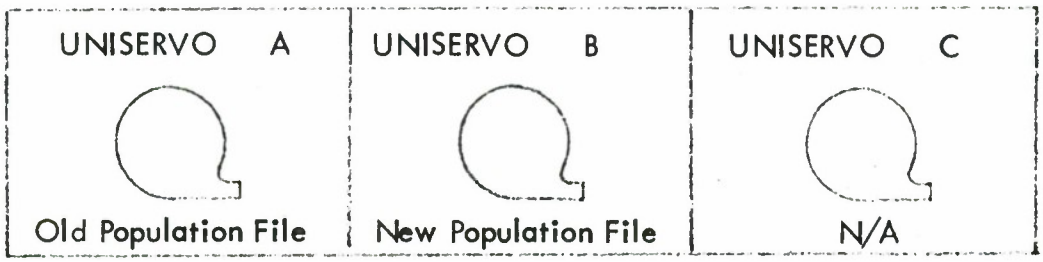
Function: POPFM2

Computer: UNIVAC - 1108

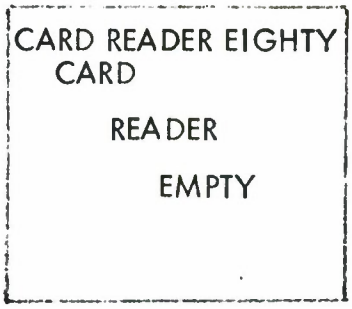
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

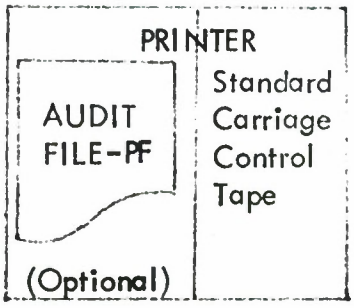
TAPES



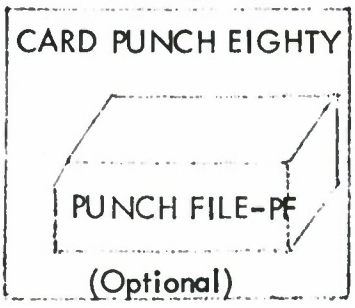
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

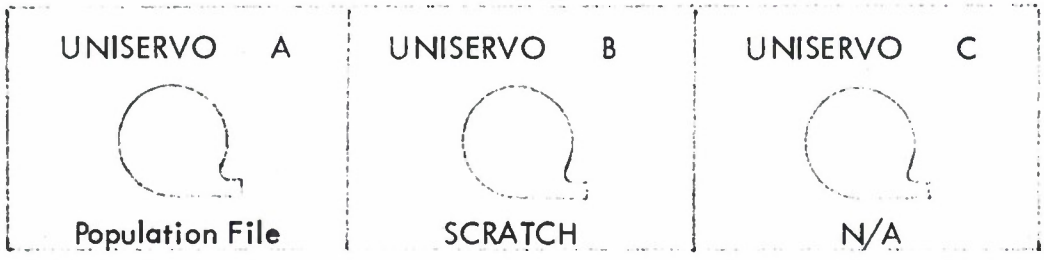
Function: SELECT

Computer: UNIVAC - 1108

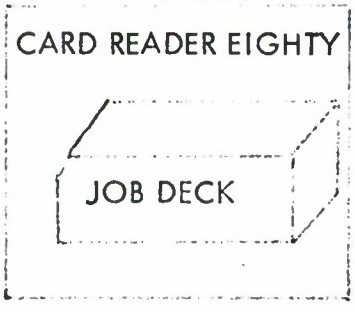
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

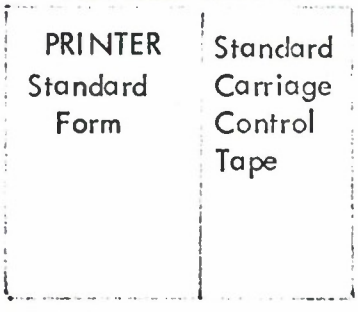
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

@ RUN 1 SELECT,DDCG,5,300
@ ASG A = XXXX,B = YYYY
@ BREI COB SELECT

XXXX = POPFILE1 reel number
YYYY = JOVSP reel number

(COBOL Source Program Deck - SELECT)

@ XQT SELECT

(Control Card - S)
(Test Selection File Deck)

@ FIN

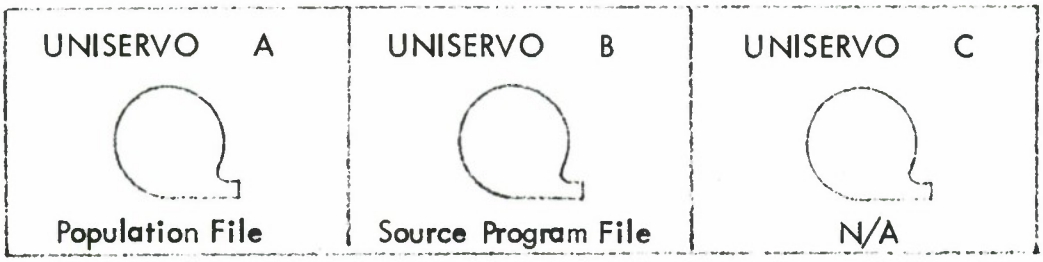
JCVS USAGE FORM

Function: SELECT
Computer: UNIVAC - 1108

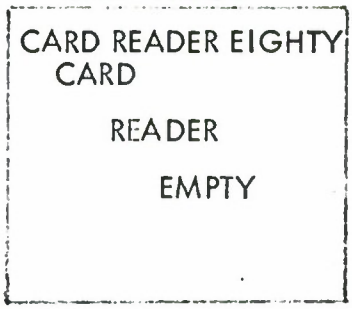
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

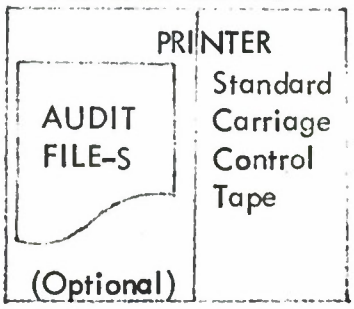
TAPES



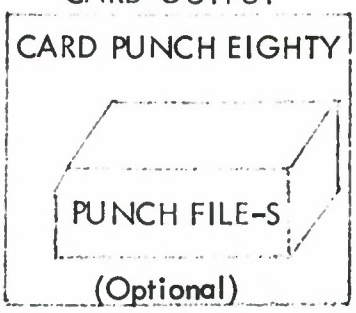
CARD INPUT



PRINTED OUTPUT



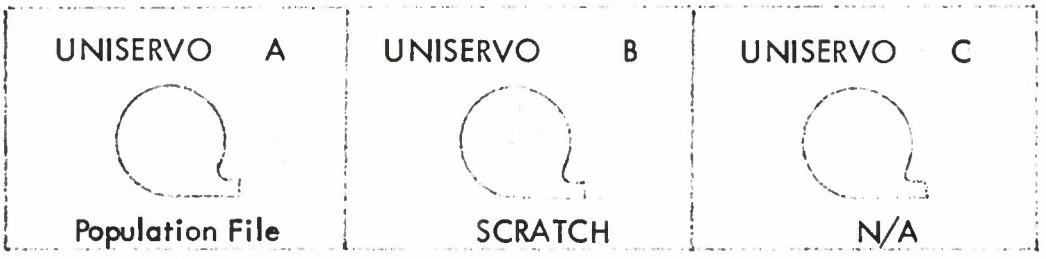
CARD OUTPUT



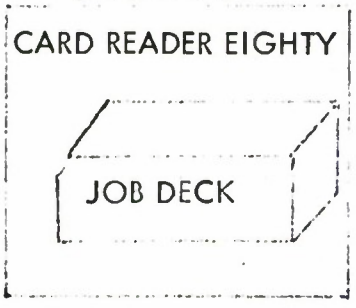
JCVS USAGE FORM

Function: SELECT
Computer: UNIVAC - 1108
Operating Philosophy: Load Binary Deck and Go
Stage: INPUT

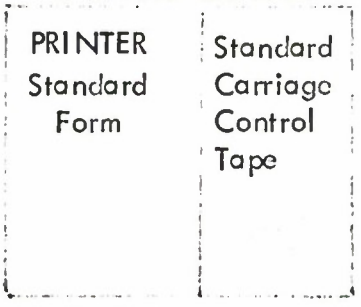
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

@ RUN SELECT,DDLG,5,300
@ ASG A = XXXX B = YYYY

XXXX = POPFILE1 reel number
YYYY = JOVSP reel number

(Binary Program Deck - Select)

@ XQT SELECT

(Control Card - S)
(Test Selection File Deck)

@ FIN

JCVS USAGE FORM

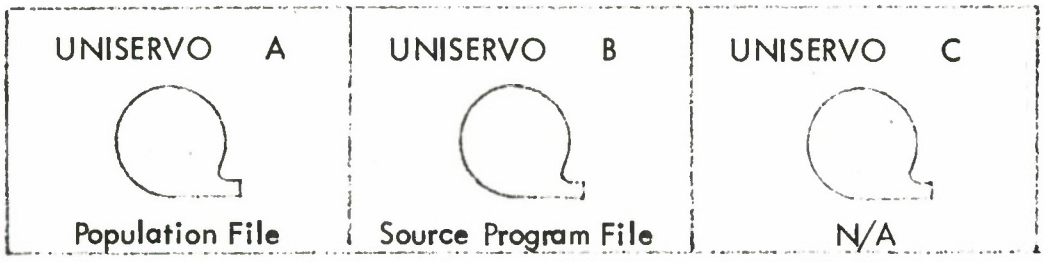
Function: SELECT

Computer: UNIVAC - 1108

Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

TAPES



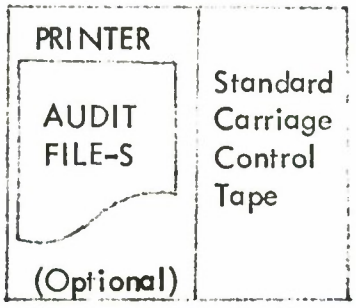
CARD INPUT

CARD READER EIGHTY CARD

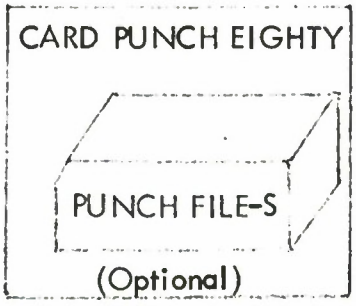
READER

EMPTY

PRINTED OUTPUT



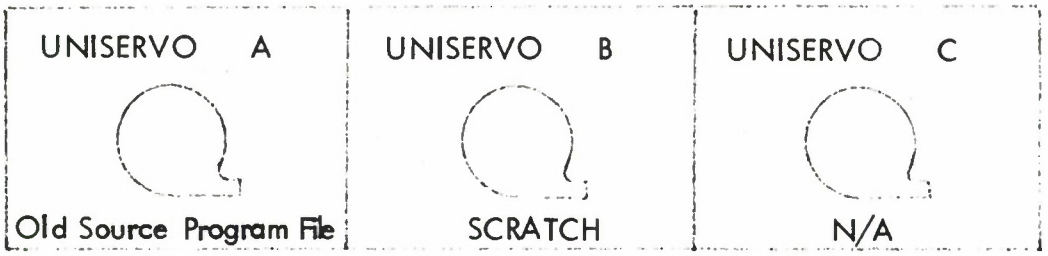
CARD OUTPUT



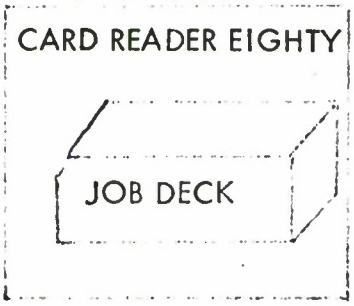
JCVS USAGE FORM

Function: SOPMM
Computer: UNIVAC - 1108
Operating Philosophy: Compile Source Program and Go
Stage: INPUT

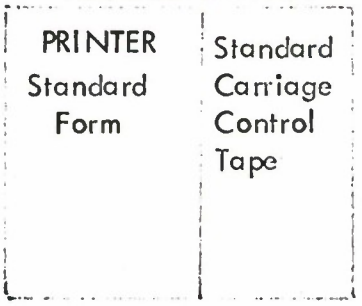
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

```
@ RUN SOPMM,DDLG,5,300                    XXXX = JOVSP reel number
@ ASG A = XXXX,B
@ BREI COB SOPMM

          (COBOL Source Program Deck - SOPMM)

@ XQT SOPMM

          (Control Card - SP)
          (Current File - SP Deck)

@ FIN
```

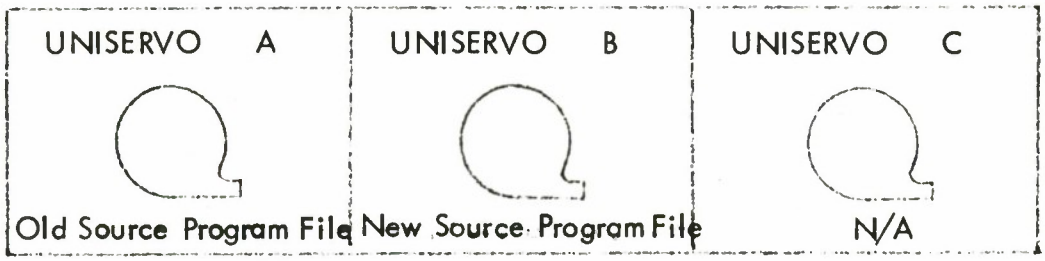
JCVS USAGE FORM

Function: SOPMM
Computer: UNIVAC - 1108

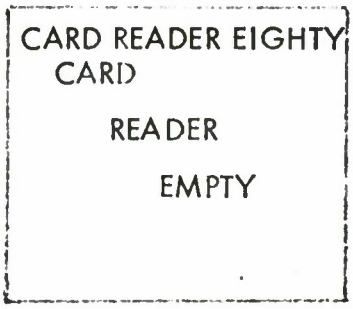
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

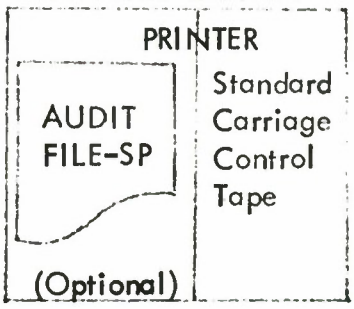
TAPES



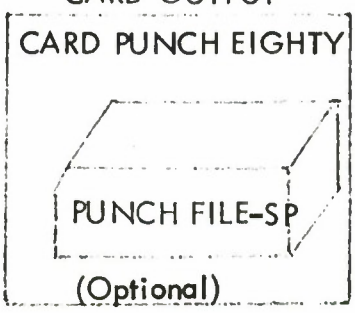
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

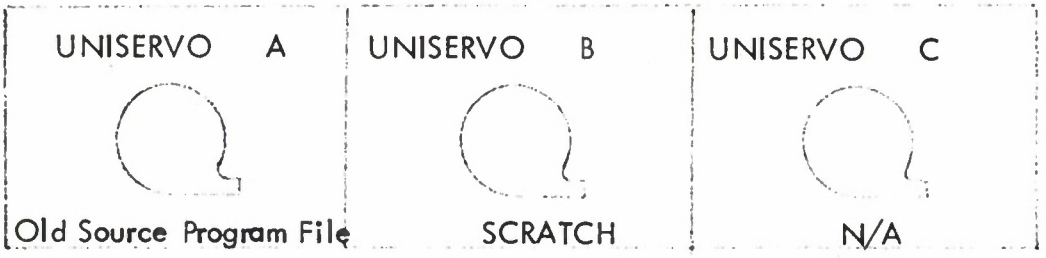
Function: SOPMM

Computer: UNIVAC - 1108

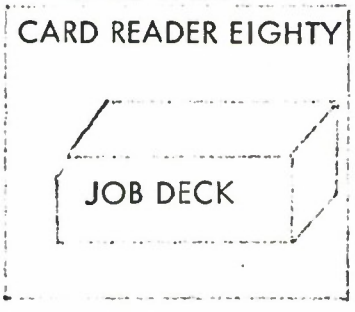
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT

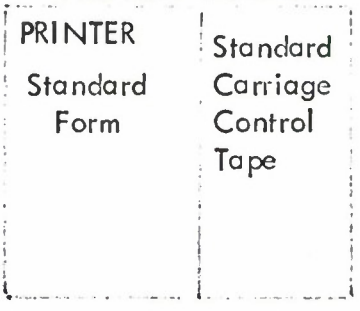
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

@ RUN SOPMM, DDLG,5,300
@ ASG A = XXXX,B

XXXX = JOVSP reel number

(Binary Program Deck - SOPMM)

@ XQT SOPMM

(Control Card - SP)
(Current File - SP Deck)

@ FIN

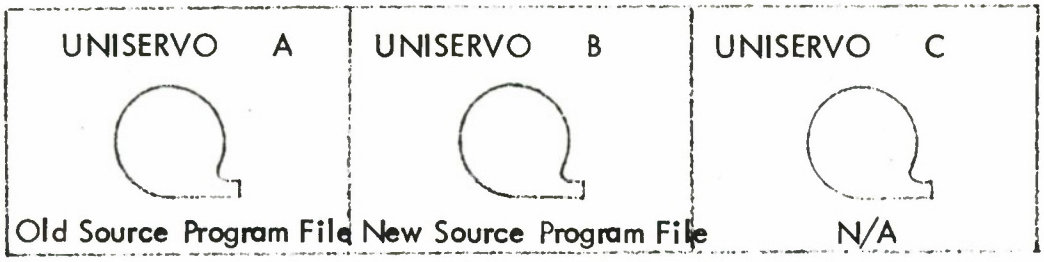
JCVS USAGE FORM

Function: SOPMM
Computer: UNIVAC - 1108

Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

TAPES



CARD INPUT

CARD READER EIGHTY CARD

READER

EMPTY

PRINTED OUTPUT

PRINTER

AUDIT FILE-SP

(Optional)

Standard Carriage Control Tape

CARD OUTPUT

CARD PUNCH EIGHTY

PUNCH FILE-SP

(Optional)

JCVS USAGE FORM

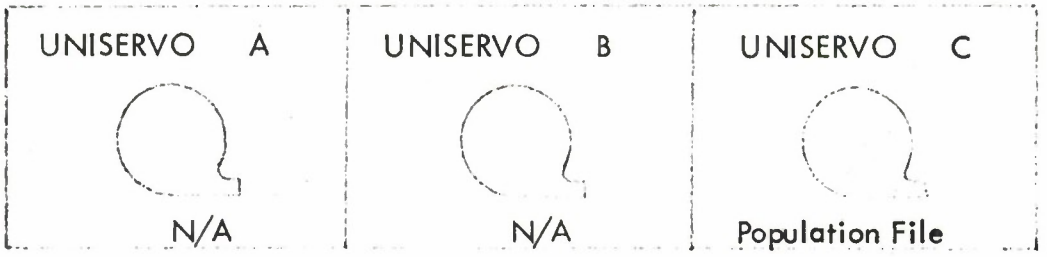
Function: JCVSRP

Computer: UNIVAC - 1108

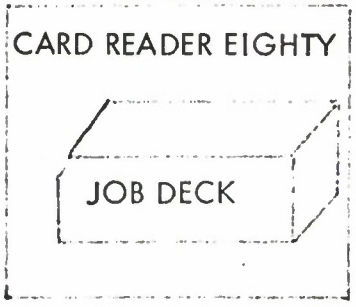
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

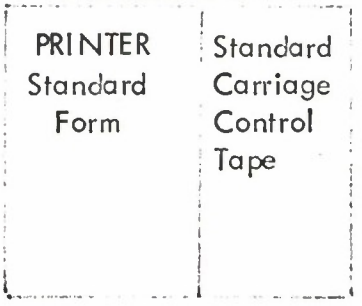
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

@ RUN 1 JCVSRP,DDCG,5,300
@ ASG C = XXXX
@ BREI COB JCVSRP

XXXX = POPFILE1 reel number

(COBOL Source Program Deck - JCVSRP)

@ XQT JCVSRP

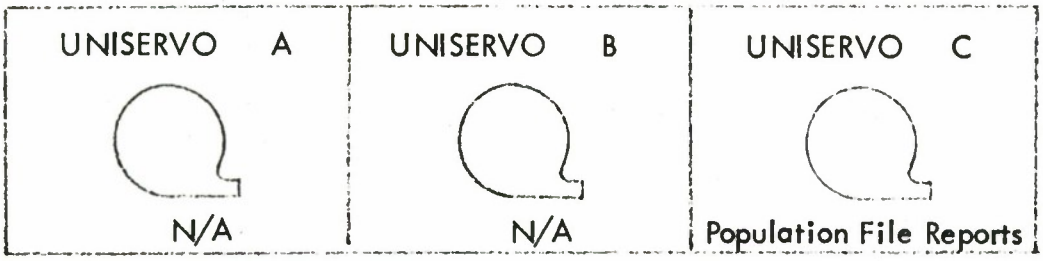
(Control Card - RP)

@ FIN

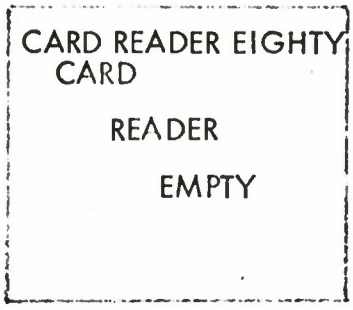
JCVS USAGE FORM

Function: JCVSRP
Computer: UNIVAC - 1108
Operating Philosophy: **Compile Source Program and Go**
Stage: OUTPUT

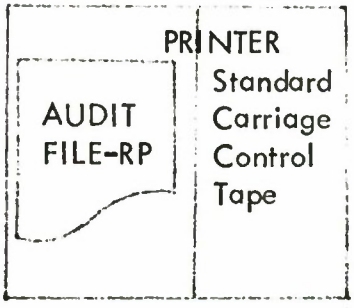
TAPES



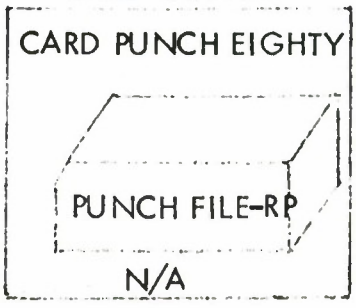
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

Function: JCVSRP

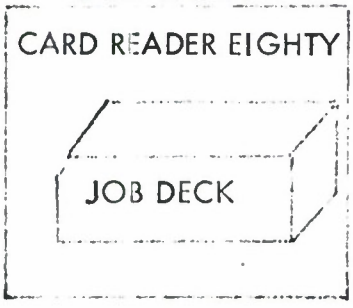
Computer: UNIVAC - 1108

Operating Philosophy: Load Binary Deck and Go

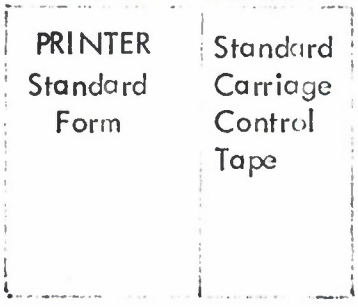
Stage: INPUT

TAPES

CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

@ RUN 1 JCVSRP,DDLG,5,300
@ ASG C = XXXX

XXXXX = POPFILE1 reel number

(Binary Program Deck - JCVSRP)

@ XQT JCVSRP

(Control Card - IP)

@ FIN

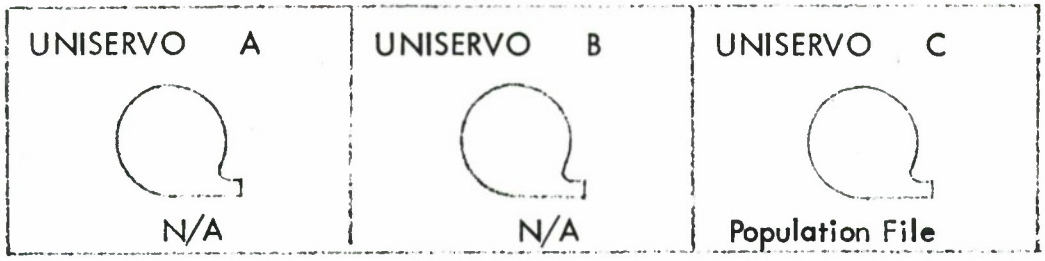
JCVS USAGE FORM

Function: JCVSRP
Computer: UNIVAC - 1108

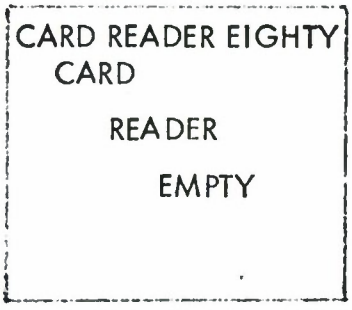
Operating Philosophy: **Load Binary Deck and Go**

Stage: OUTPUT

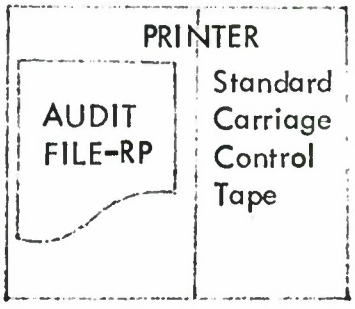
TAPES



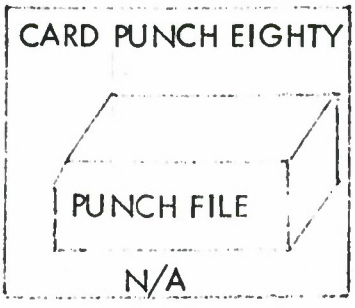
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

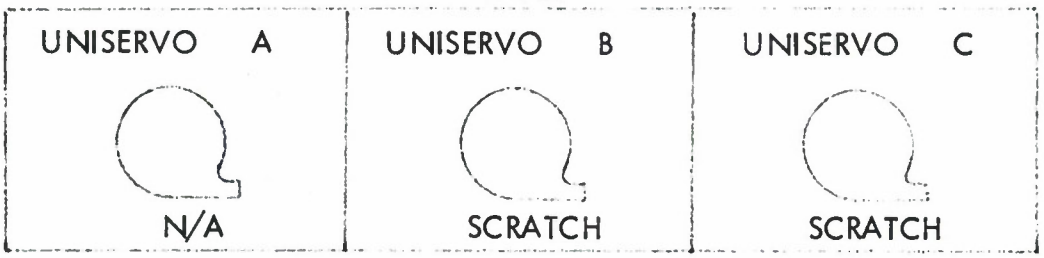
Function: INIPOP1

Computer: UNIVAC - 1108

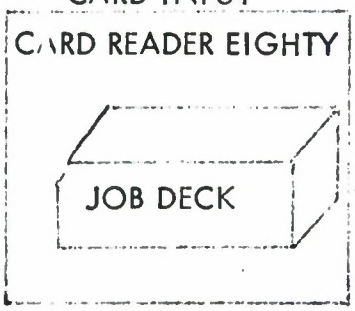
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

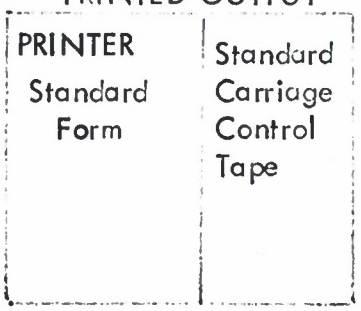
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

```
@ RUN 1 INIPOP,DD1CG,5,300
@ ASG B,C
@ BREI COB INIPOP

          (COBOL Source Program Deck - INIPOP)

@ XQT INIPOP

          (Control Card - IP)
          (Current File - PF Deck)

@ FIN
```

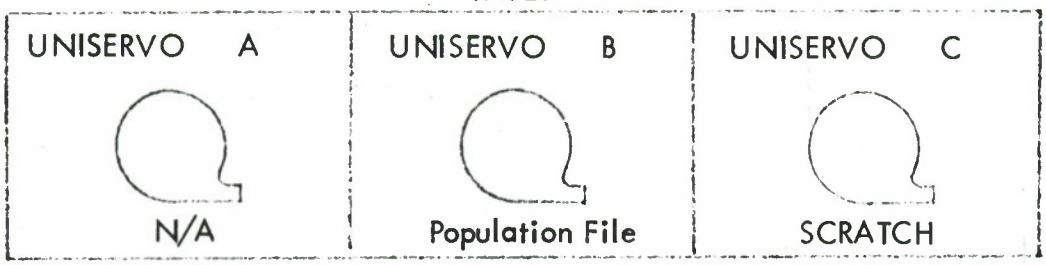
JCVS USAGE FORM

Function: INIPOP1
Computer: UNIVAC - 1108

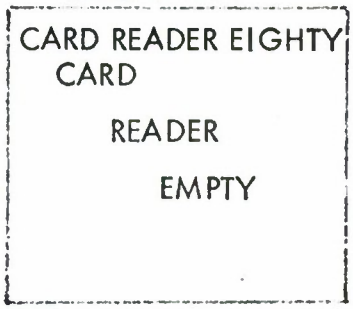
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

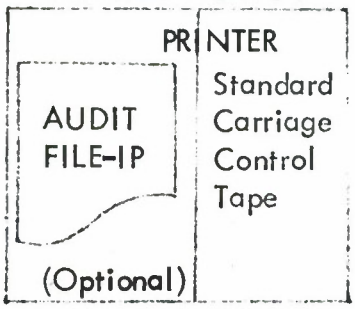
TAPES



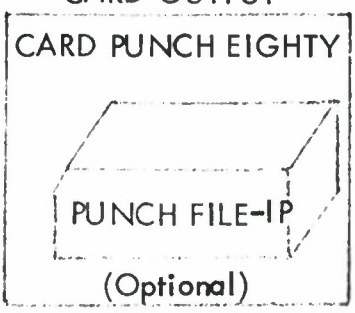
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

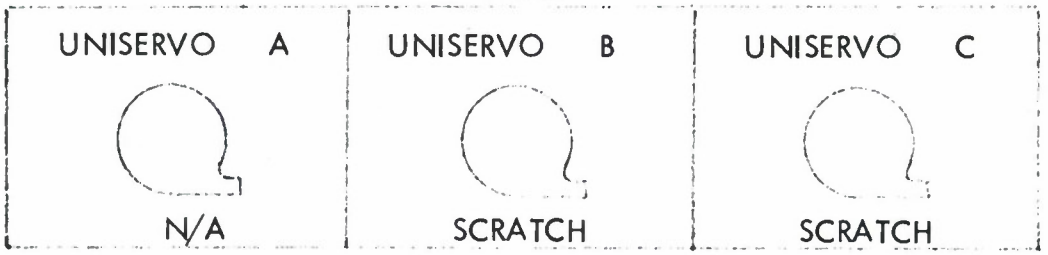
Function: INIPOP1

Computer: UNIVAC - 1108

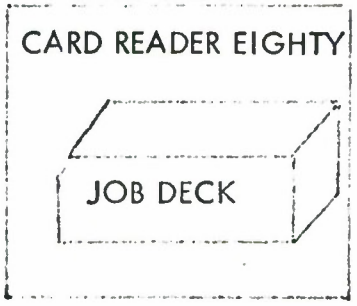
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT

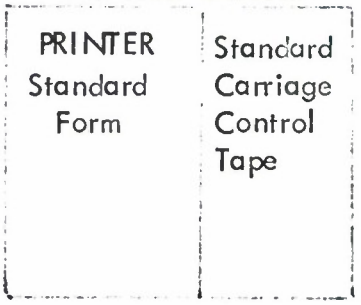
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

@ RUN 1 INIPOP,DD1LG,5,300
@ ASG B,C

(Binary Program Deck - INIPOP)

@ XQT INIPOP

(Control Card - IP)
(Current File - PF Deck)

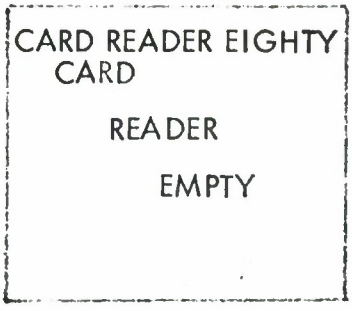
@ FIN

JCVS USAGE FORM

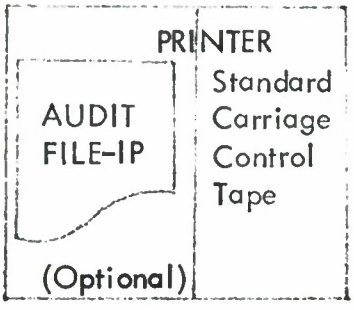
Function: INIPOP1
Computer: UNIVAC - 1108
Operating Philosophy: Load Binary Deck and Go
Stage: OUTPUT

TAPES

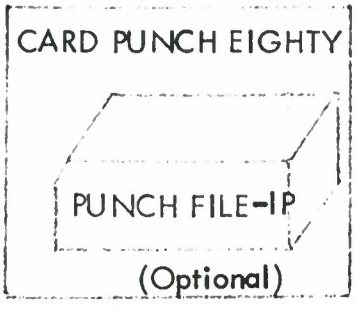
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

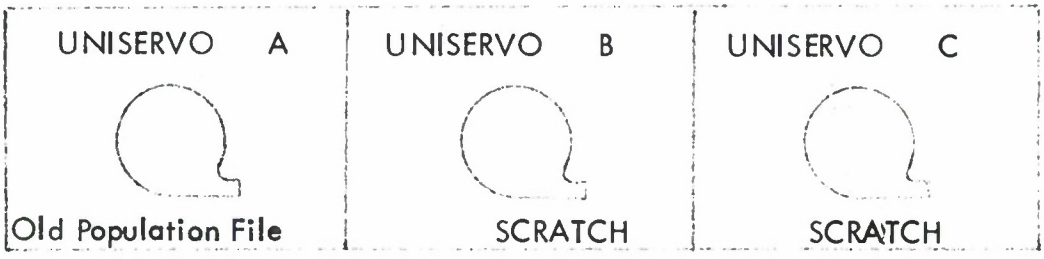
Function: INIPOP2

Computer: UNIVAC - 1108

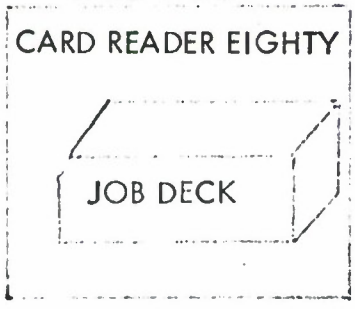
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

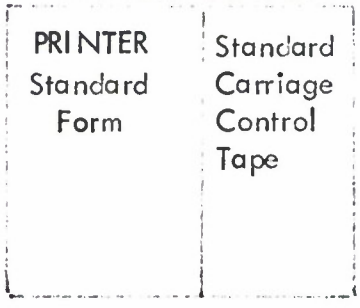
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

(1)

```
@ RUN 1 INIPOP,DD2CG,5,300
@ ASG A = XXXX,B,C
@ BREI COB INIPOP
```

XXXX = POPFILE1 reel number

(COBOL Source Program Deck - INIPOP)

```
@ XQT INIPOP
```

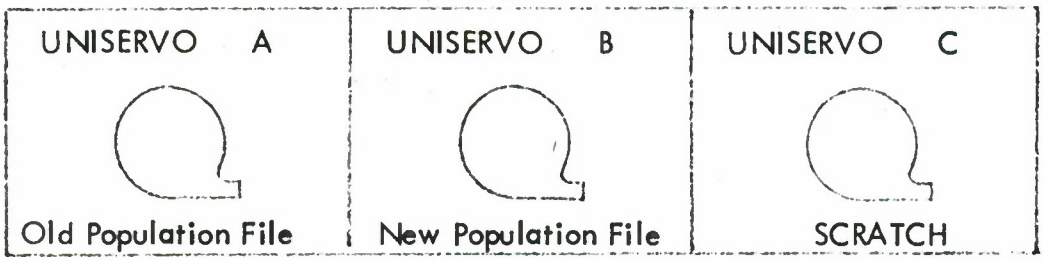
(Control Card - IP)

```
@ FIN
```

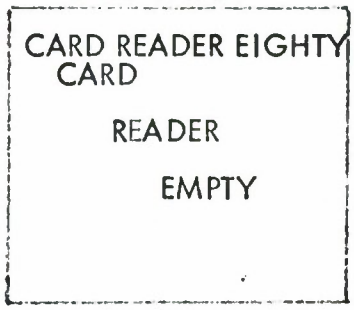
JCVS USAGE FORM

Function: INIPOP2
Computer: UNIVAC - 1108
Operating Philosophy: Compile Source Program and Go
Stage: OUTPUT

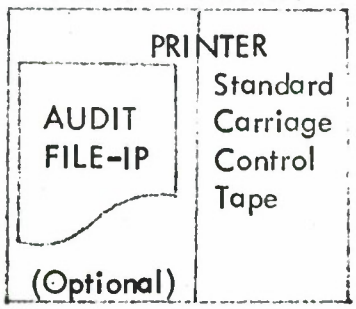
TAPES



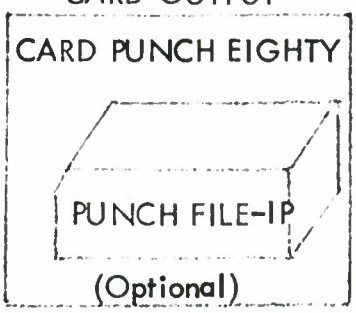
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

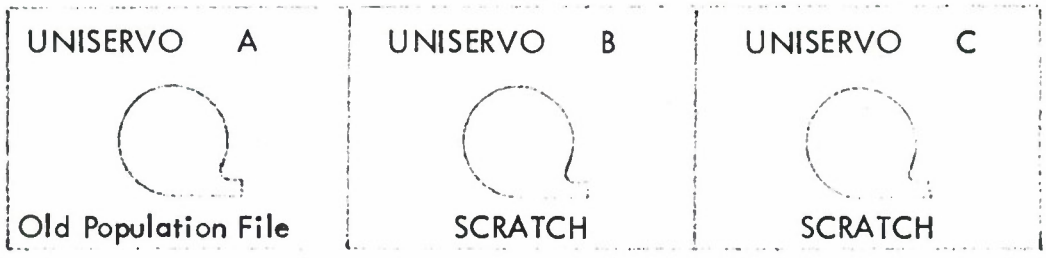
Function: INIPOP2

Computer: UNIVAC - 1108

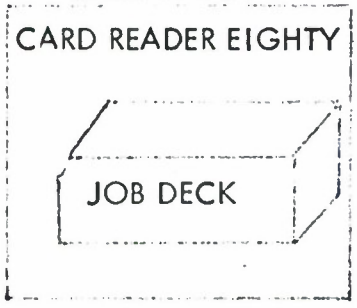
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT

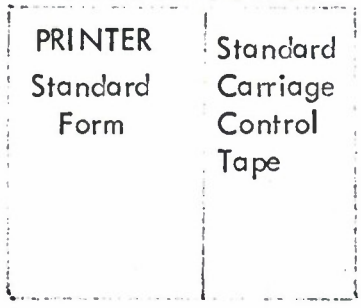
TAPES



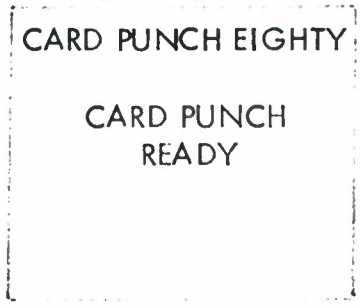
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JOB DECK STRUCTURE

(1)

@ RUN 1 INIPOP,DD2LG,5,300
@ ASG A = XXXX,B,C

XXXX = POPFILE1 reel number

(Binary Program Deck INIPOP)

@ XQT INIPOP

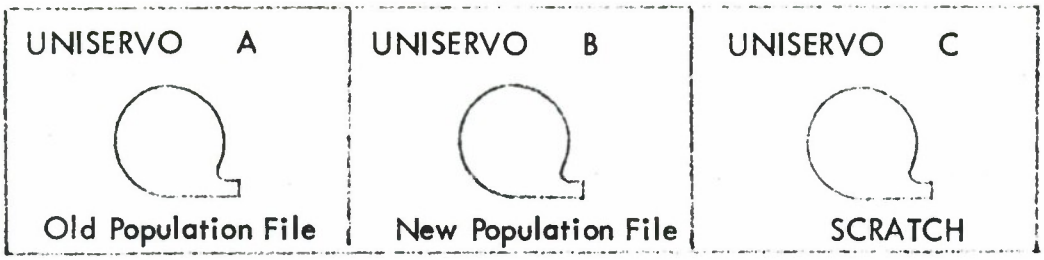
(Control Card - IP)

@ FIN

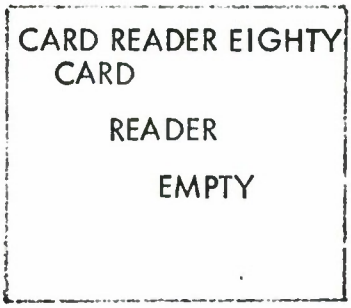
JCVS USAGE FORM

Function: INIPOP2
Computer: UNIVAC - 1108
Operating Philosophy: Load Binary Deck and Go
Stage: OUTPUT

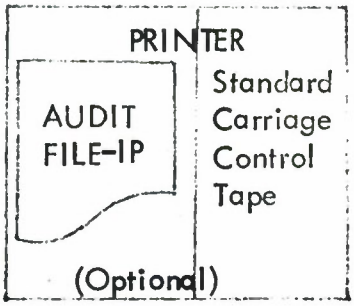
TAPES



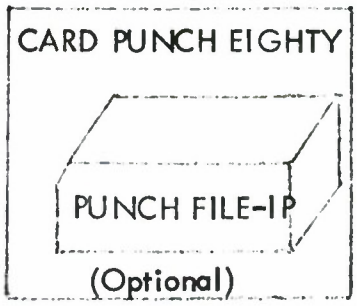
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

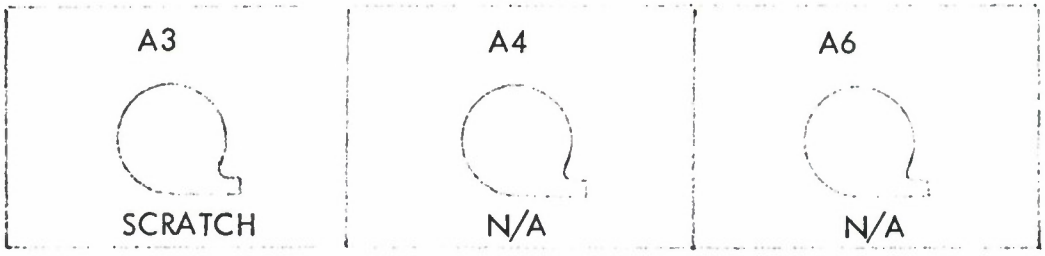
Function: POPFM1

Computer: GE-635

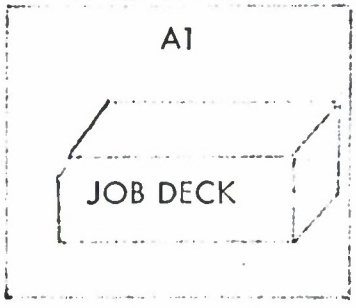
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

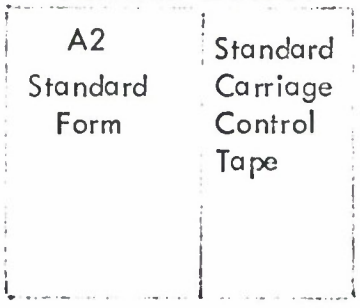
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

```
(1)         (8)             (16)
$           IDENT           3154203,DATDY
$           COBOL
$           INCODE          IBMC

            (COBOL Source Program Deck - POPFM)

$           EXECUTE         DUMP
$           LIMITS          15,32000
$           SYSOUT          A2
$           TAPE            A3,X3S,,POPFILE1,,SAVE
$           SYSOUT          A5
$           DATA            A1

            (Control Card - PF)
            (Current File - PF Deck)

$           ENDJOB
***EOF
```

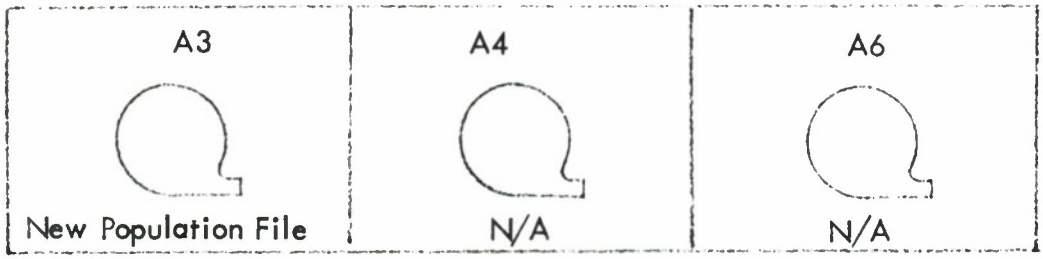
JCVS USAGE FORM

Function: POPFM1
Computer: GE-635

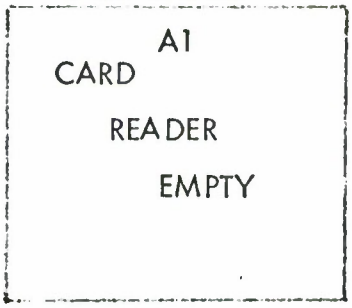
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

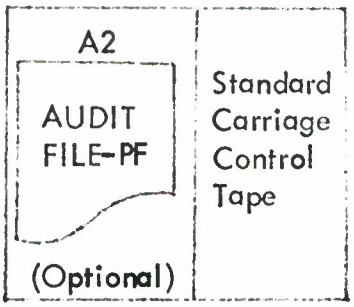
TAPES



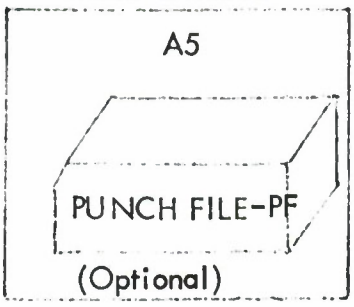
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

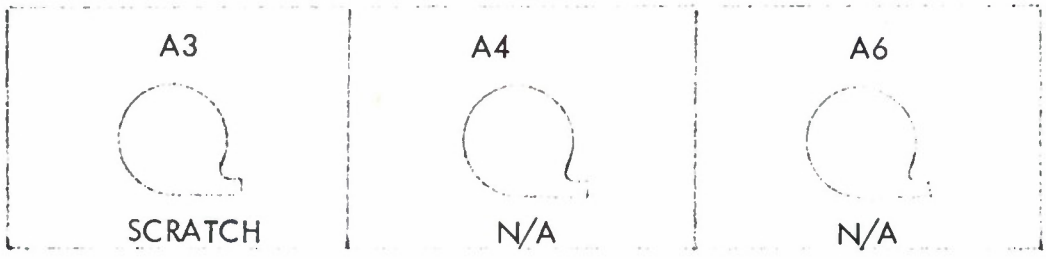
Function: POPFM1

Computer: GE-635

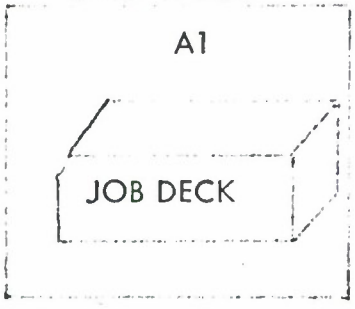
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT

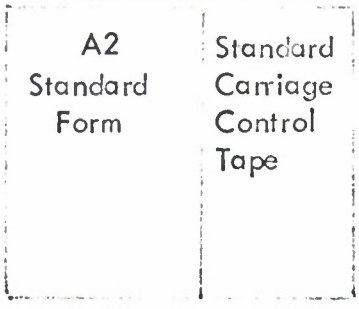
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

```
(1)         (8)             (16)
$           IDENT           3154203,DATDY
$           OPTION          COBOL

            (Binary Program Deck - POPFM)

$           EXECUTE         DUMP
$           LIMITS          15,32000
$           SYSOUT          A2
$           TAPE            A3,X3S,,POPFILE1,,XXXX          XXXX = reel number
$           SYSOUT          A5
$           DATA            A1
            (Control Card - PF)
            (Current File - PF Deck)

$           ENDJOB
***EOF
```

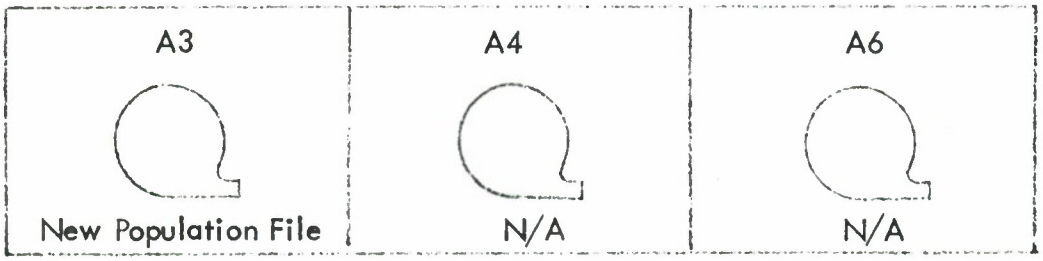
JCVS USAGE FORM

Function: POPFM1
Computer: GE-635

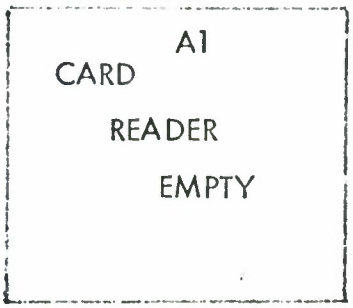
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

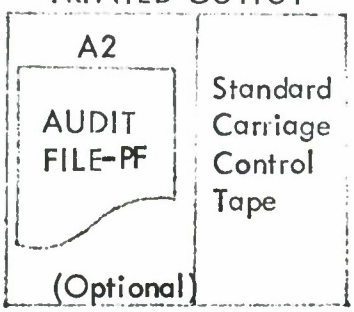
TAPES



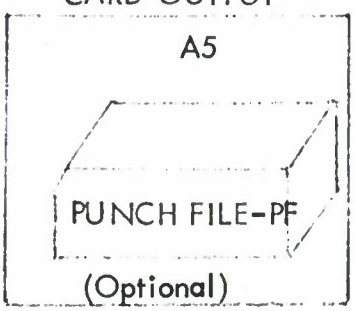
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

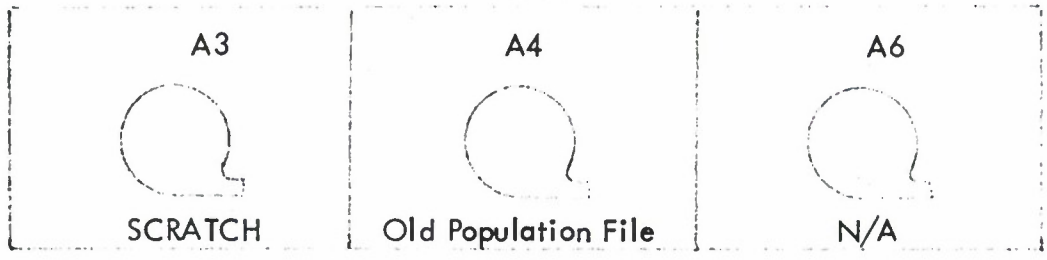
Function: POPFM2

Computer: GE-635

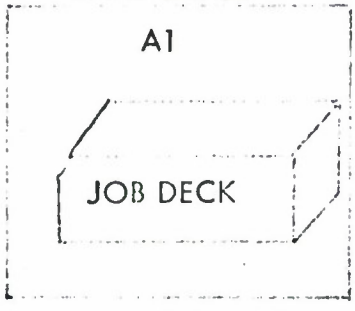
Operating Philosophy: Compile Source Program and Go

Stage: INPUT

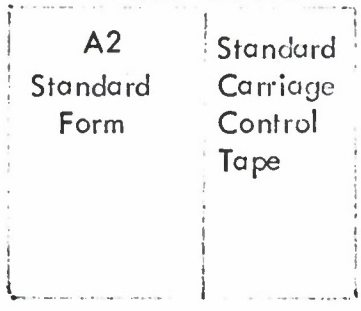
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

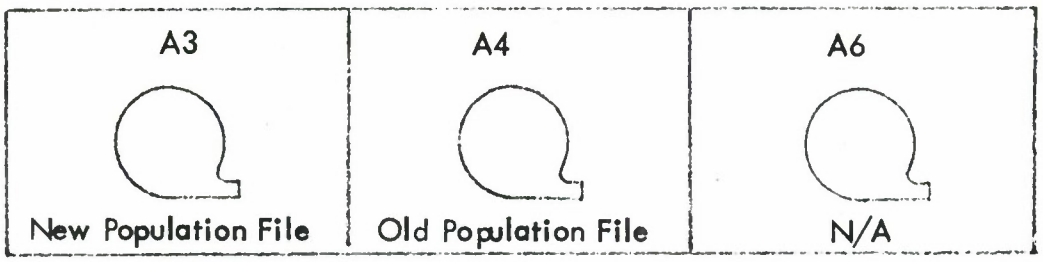
JOB DECK STRUCTURE

```
(1)         (8)          (16)
$           IDENT        3154203,DATDY
$           COBOL
$           INCODE       IBMC

            (COBOL Source Program Deck - POPFM)

$           EXECUTE      DUMP
$           LIMITS       15,32000
$           SYSOUT       A2
$           TAPE         A3,X3S
$           TAPE         A4,X4D POPFILE1,,XXXX        XXXX = reel number
$           SYSOUT       A5
$           DATA         A1

            (Control Card - PF)
            (Current File - PF Deck)

$           ENDJOB
***EOF
```

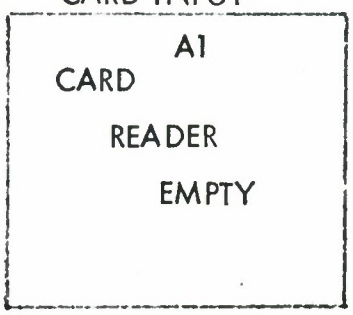
JCVS USAGE FORM

Function: POPFM2
Computer: GE-635
Operating Philosophy: Compile Source Program and Go
Stage: OUTPUT

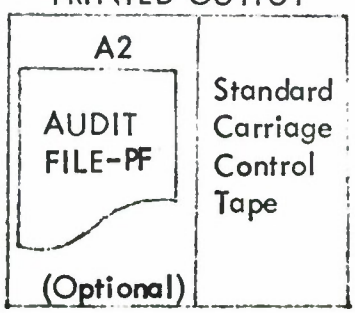
TAPES



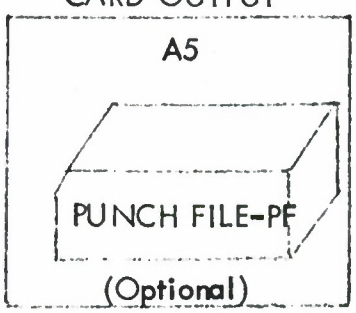
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



JCVS USAGE FORM

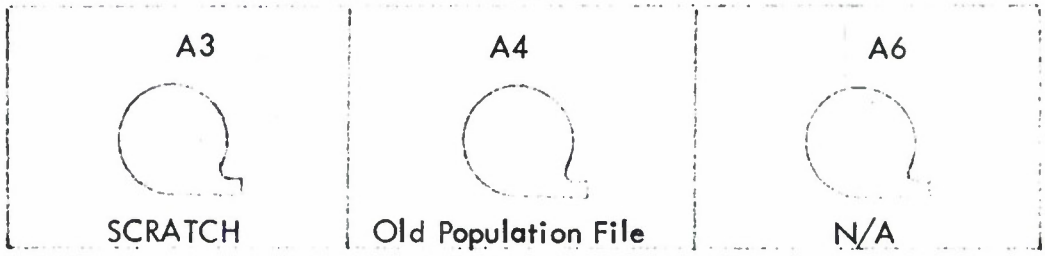
Function: POPFM2

Computer: GE-635

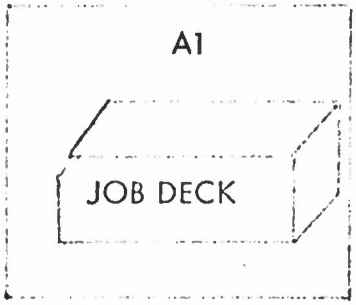
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT

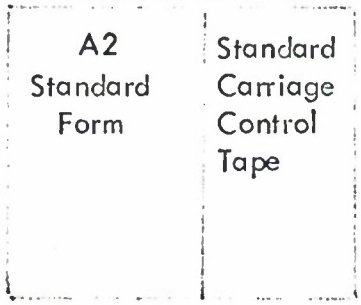
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

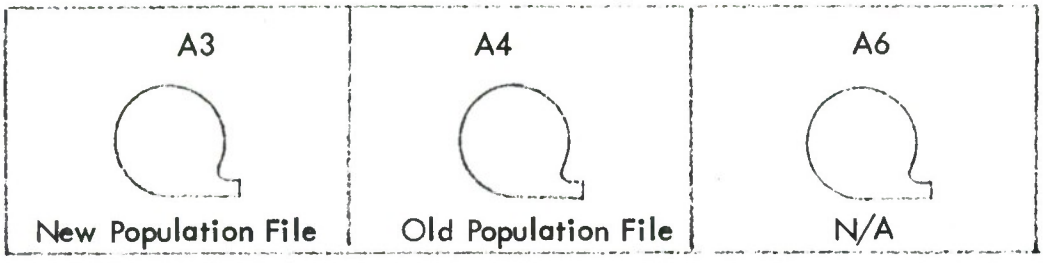
```
(1)          (8)          (16)
$            IDENT        3154203,DATDY
$            OPTION       COBOL

             (Binary Program Deck - POPFM)

$            EXECUTE      DUMP
$            LIMITS       15,32000
$            SYSOUT       A2
$            TAPE         A3,X3S
$            TAPE         A4,X4D,,POPFILE1,,XXXX        XXXX=reel number
$            SYSOUT       A5
$            DATA         A1

             (Control Card - PF)
             (Current File - PF Deck)

$            ENDJOB
***EOF
```

JCVS USAGE FORM

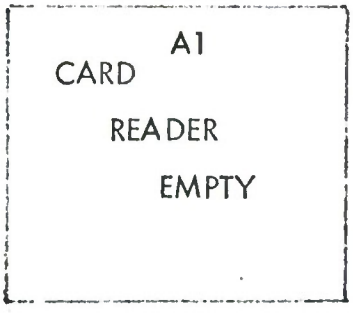
Function: POPFM2
Computer: GE-635

Operating Philosophy: Load Binary Deck and Go
Stage: OUTPUT

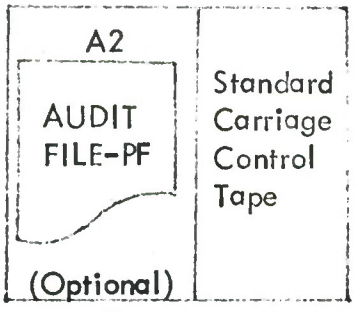
TAPES



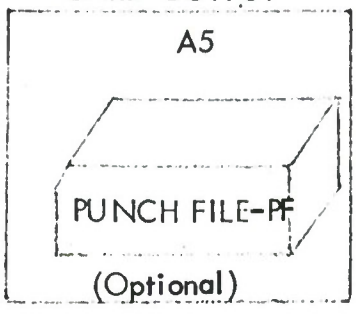
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



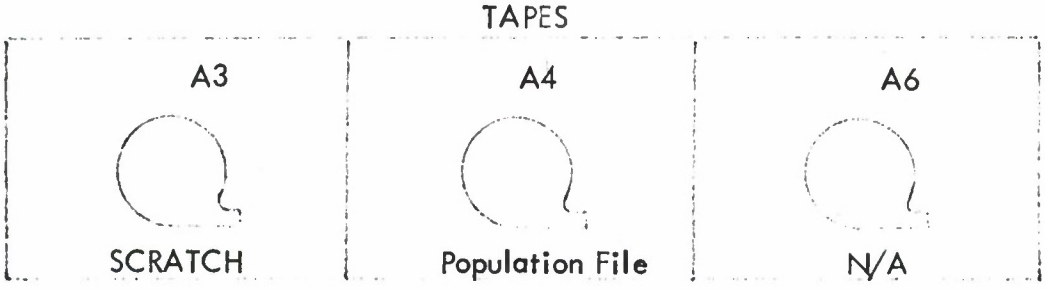
JCVS USAGE FORM

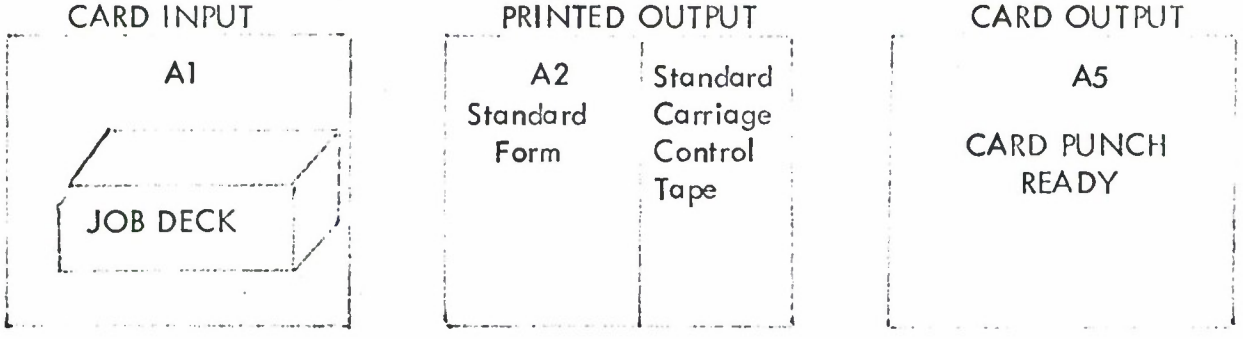
Function: SELECT

Computer: GE-635

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

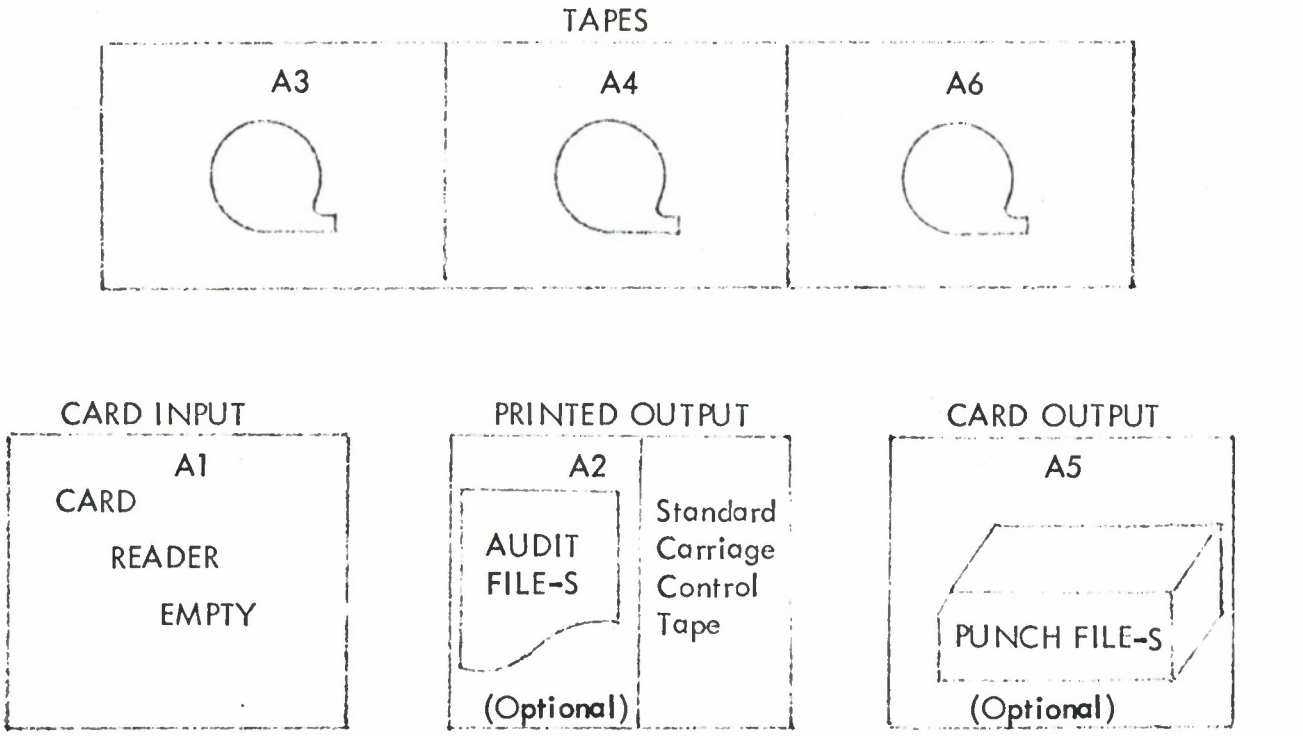
```
(1)          (8)              (16)
$            IDENT            3154203,DATDY
$            COBOL
$            INCODE IBMC

             (COBOL Source Program Deck - SJCVS)

$            EXECUTE          DUMP
$            LIMITS           15,32000
$            TAPE             A3,X3S,,SAVE,,JOVSP
$            SYSOUT           A5
$            SYSOUT           A2
$            DATA             A1

             (Control Card - S)
             (Test Selection File Deck)

$            ENDJOB
***EOF
```

JCVS USAGE FORM

Function: SELECT
Computer: GE-635

Operating Philosophy: **Compile Source Program and Go**

Stage: OUTPUT



JCVS USAGE FORM

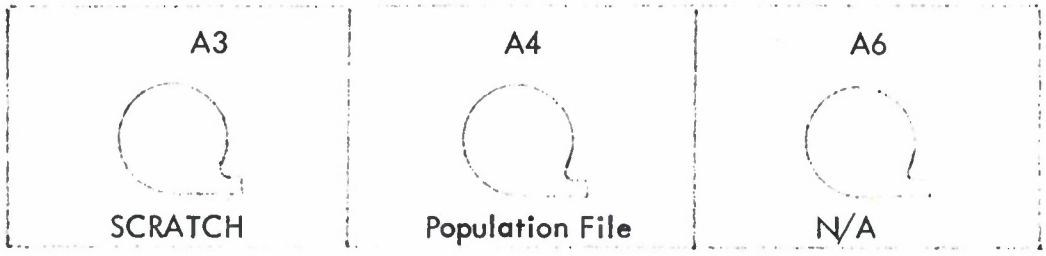
Function: SELECT

Computer: GE-635

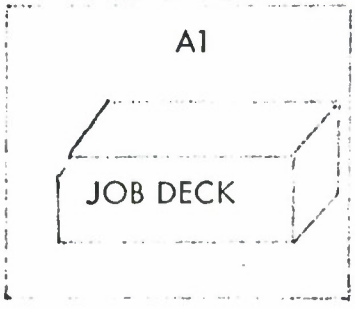
Operating Philosophy: Load Binary Deck and Go

Stage: INPUT

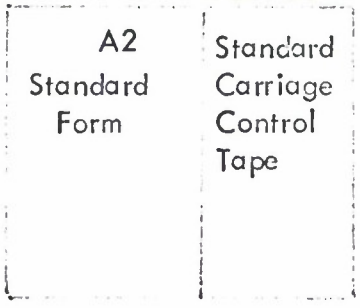
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

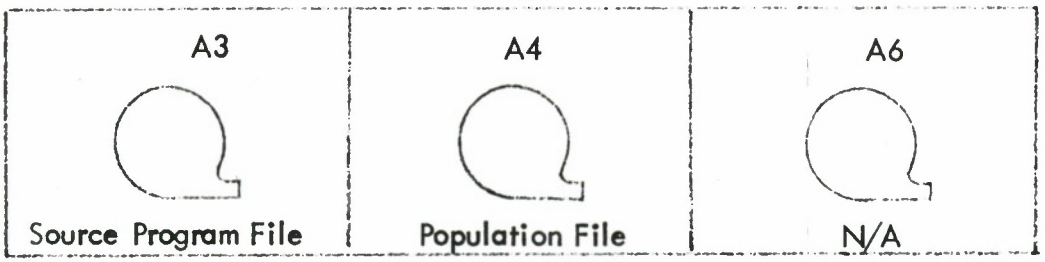
JOB DECK STRUCTURE

```
(1)             (8)             (16)
 $              IDENT           3154203,DATDY
 $              OPTION          COBOL

                (Binary Program Deck - SJCVS)

 $              EXECUTE         DUMP
 $              LIMITS          15,32000
 $              TAPE            A3,X3S,,SAVE,,JOVSP
 $              SYSOUT          A5
 $              SYSOUT          A2
 $              DATA            A1

                (Control Card - S)
                (Test Selection File Deck)

 $              ENDJOB
***EOF
```

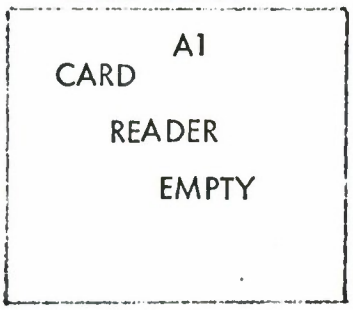
JCVS USAGE FORM

Function: SELECT
Computer: GE-635
Operating Philosophy: Load Binary Deck and Go
Stage: OUTPUT

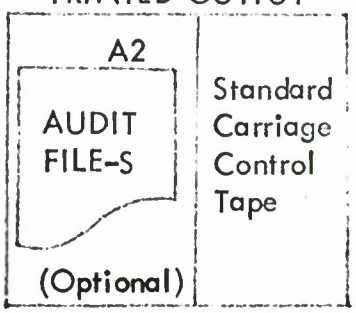
TAPES



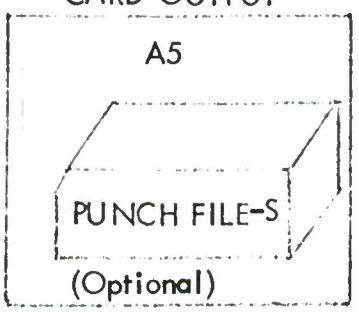
CARD INPUT



PRINTED OUTPUT

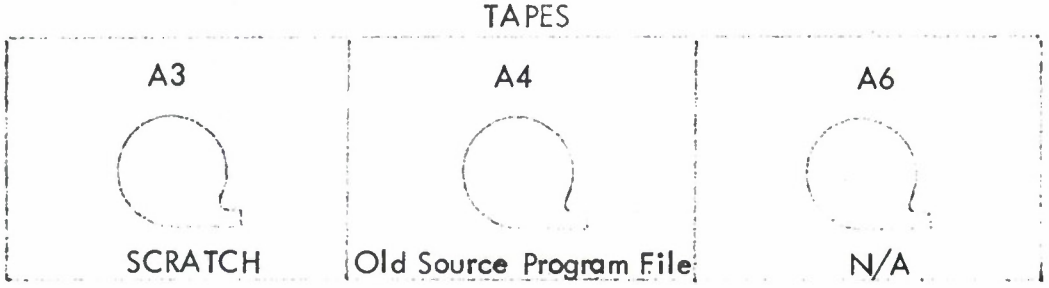


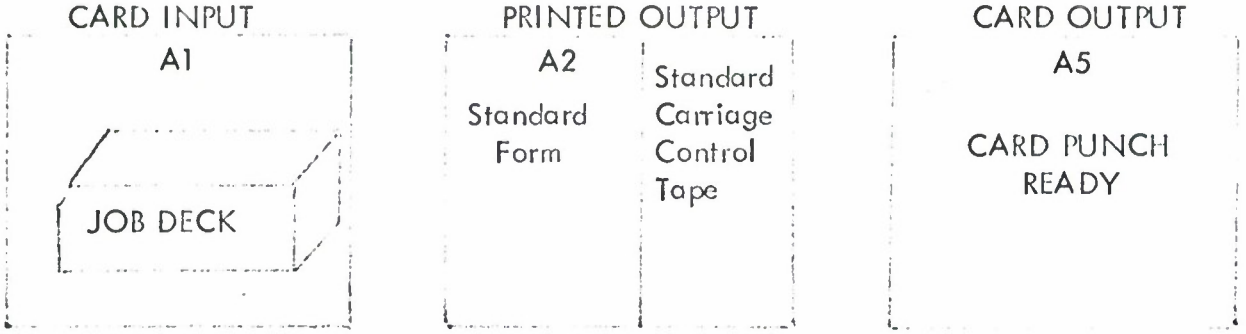
CARD OUTPUT



JCVS USAGE FORM

Function: SOPMM
Computer: GE-635
Operating Philosophy: Compile Source Program and Go
Stage: INPUT





JOB DECK STRUCTURE

```
(1)         (8)                 (16)
$           IDENT               3154203,DATDY
$           COBOL
$           INCODE              IBMC

            (COBOL Source Program Deck - SOPMM)

$           EXECUTE             DUMP
$           LIMITS              15,32000
$           SYSOUT              A2
$           TAPE                A3,X3S
$           TAPE                A4,X4S,,JOVSP,,XXXX          XXXX = reel number
$           SYSOUT              A5
$           DATA                A1

            (Control Card - SP)
            (Current File - SP - Deck)

$           ENDJOB
***EOF
```

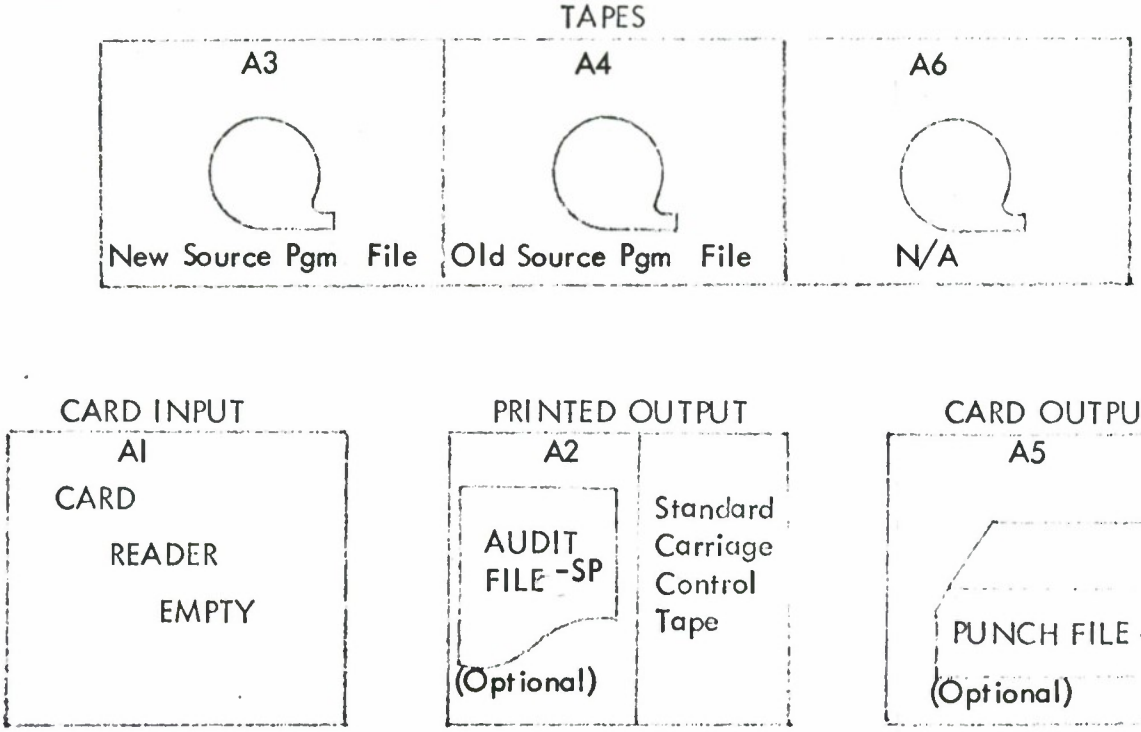
JCVS USAGE FORM

Function: SOPMM

Computer: GE-635

Operating Philosophy: Compile Source Program and Go

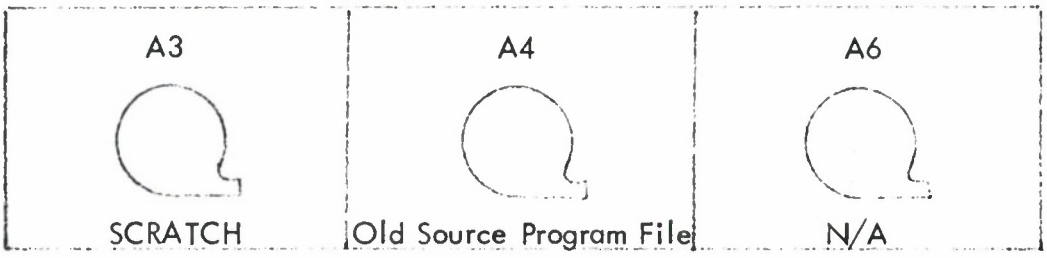
Stage: OUTPUT



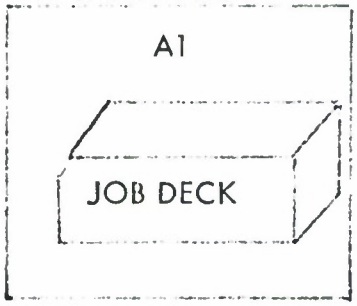
JCVS USAGE FORM

Function: SOPMM
Computer: GE-635
Operating Philosophy: Load Binary Deck and Go
Stage: INPUT

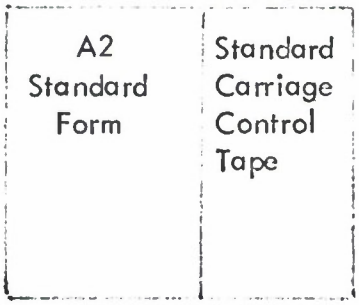
TAPES



CARD INPUT



PRINTED OUTPUT



CARD OUTPUT

JOB DECK STRUCTURE

```
(1)         (8)          (16)
$           IDENT        3154203,DATDY
$           OPTION       COBOL

            (Binary Program Deck - SOPMM)

$           EXECUTE      DUMP
$           LIMITS       15,32000
$           SYSOUT       A2
$           TAPE         A3,X3S
$           TAPE         A4,X4S,,JOVSP,,XXXX        XXXX = reel number
$           SYSOUT       A5
$           DATA         A1

            (Control Card - SP)
            (Current File - SP Deck)

$           ENDJOB
***EOF
```

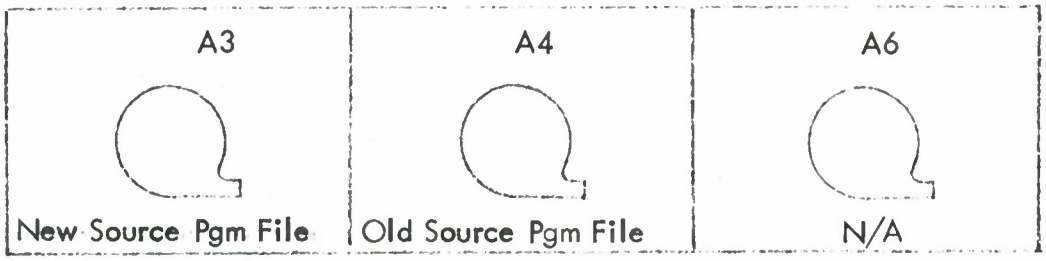
JCVS USAGE FORM

Function: SOPMM
Computer: GE-635

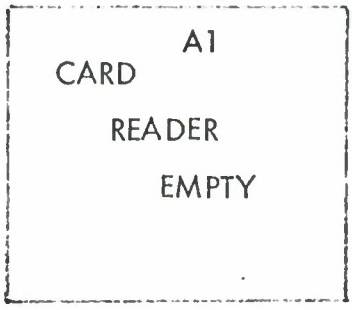
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

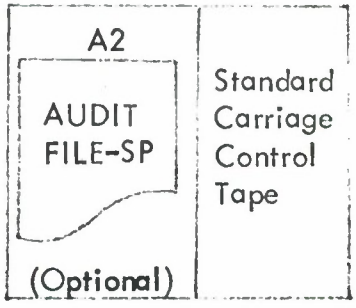
TAPES



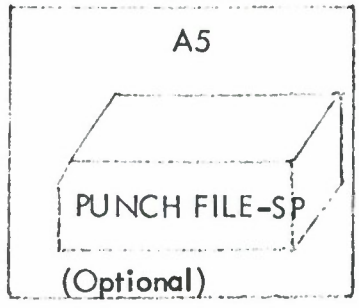
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



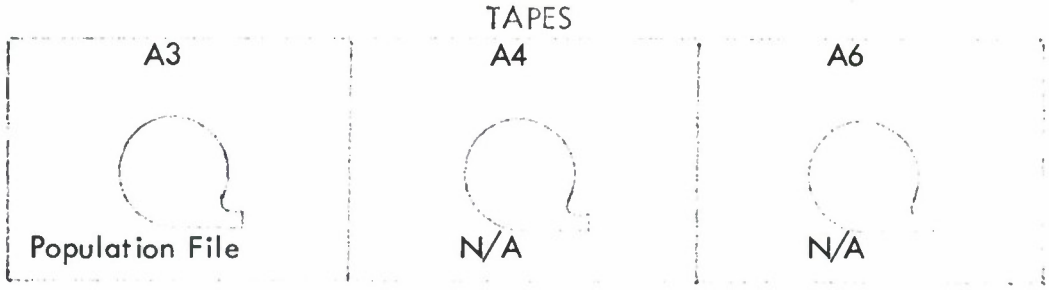
JCVS USAGE FORM

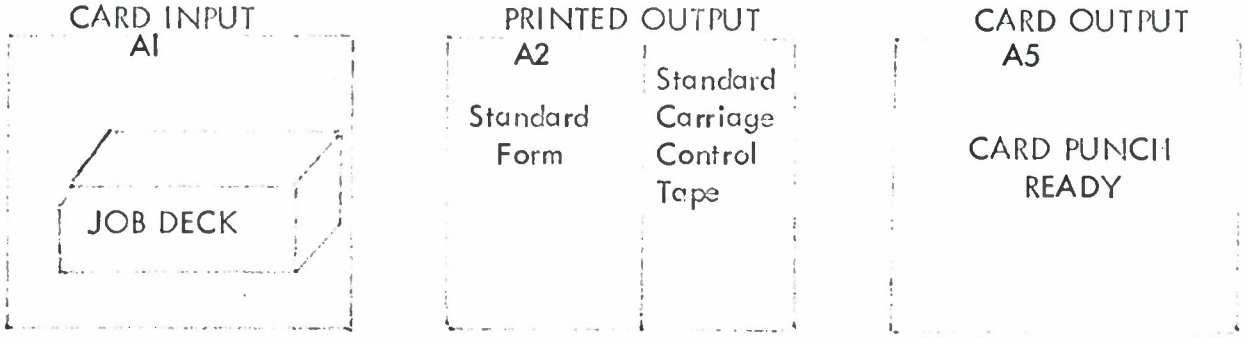
Function: JCVSRP

Computer: GE-635

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

```
(1)         (8)         (16)

$           IDENT       3154203, DATDY
$           COBOL
$           INCODE      IBMC

          (COBOL Source Program Deck - JCVSRP)

$           EXECUTE     DUMP
$           LIMITS      15,32000
$           SYSOUT      A2
$           TAPE        A3, X3D,,POPFILE1,,XXXX        XXXX = reel number
$           DATA        A1

          (Control Card - RP)

$           ENDJOB
***EOF
```

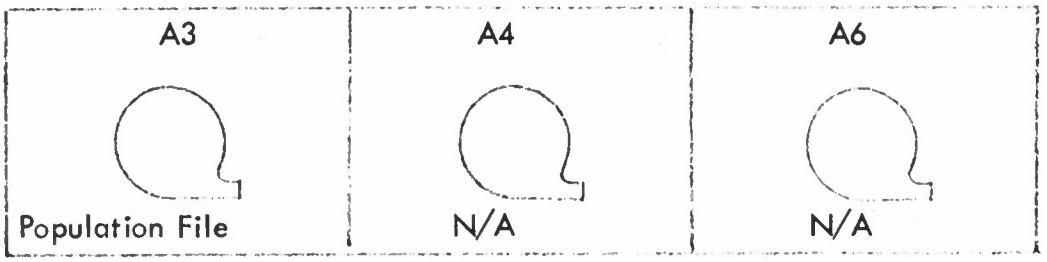
JCVS USAGE FORM

Function: JCVSRP
Computer: GE-635

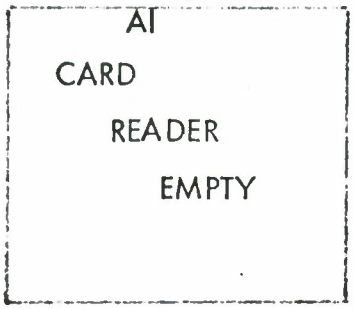
Operating Philosophy: Compile Source Program Deck and Go

Stage: OUTPUT

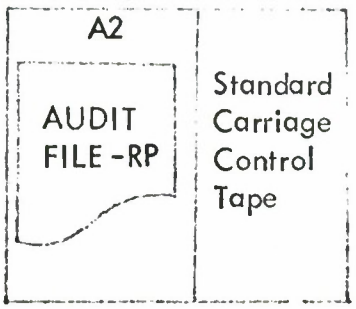
TAPES



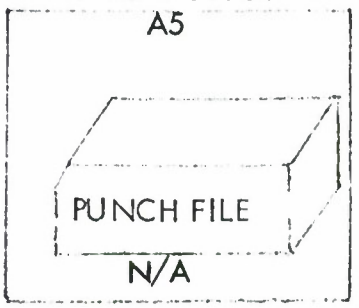
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



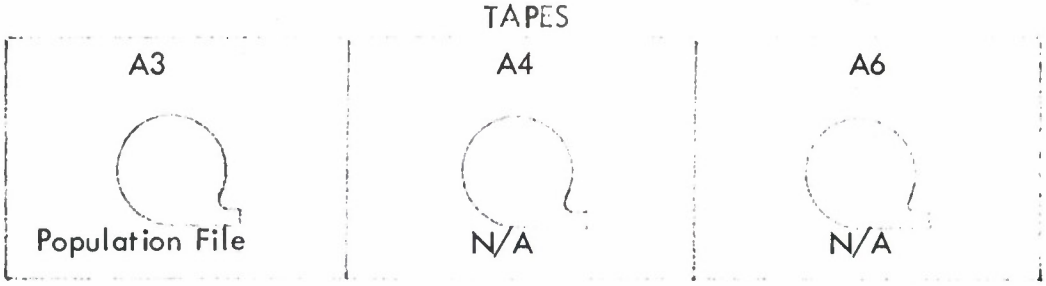
JCVS USAGE FORM

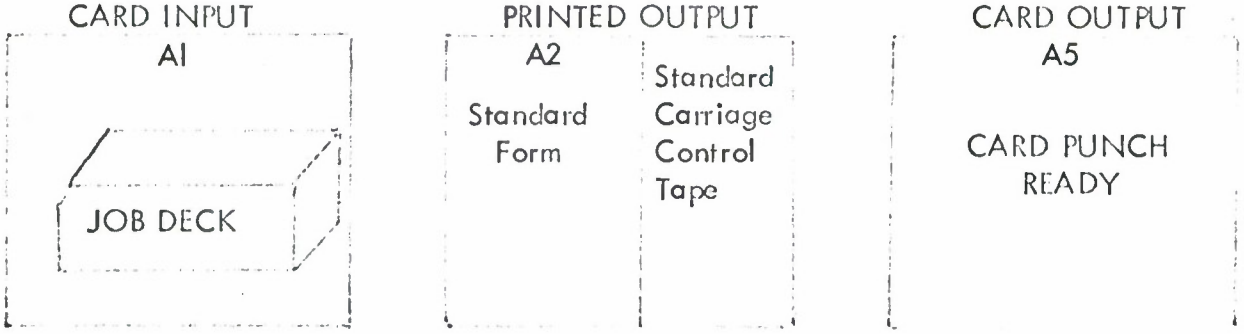
Function: JCVSRP

Computer: GE-635

Operating Philosophy: Load Binary Deck and Go

Stage: INPUT





JOB DECK STRUCTURE

```
(1)         (8)          (16)

$           IDENT        3154203,DATDY
$           OPTION       COBOL
            (Binary Program Deck - JCVSRP)

$           EXECUTE      DUMP
$           LIMITS       15,32000
$           SYSOUT       A2
$           TAPE         A3,X3D,,POPFILE1,, XXXX        XXXX = reel number
$           DATA         A1

            (Control Card - RP)

$           ENDJOB
***EOF
```

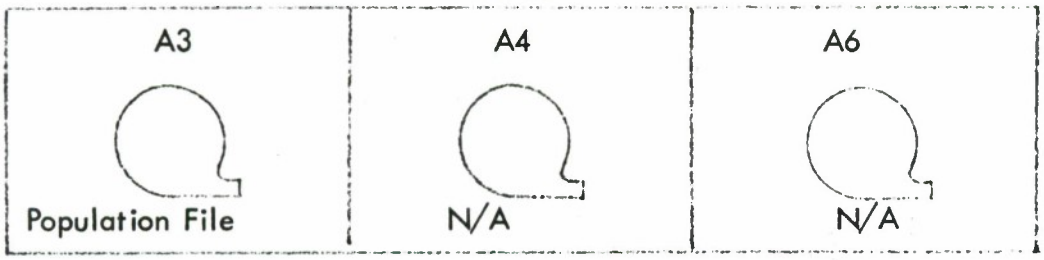
JCVS USAGE FORM

Function: JCVSRP
Computer: GE-635

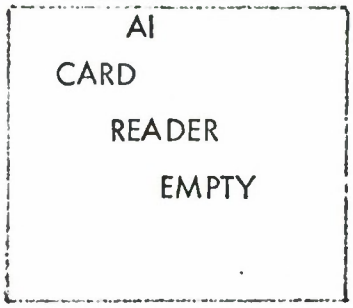
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

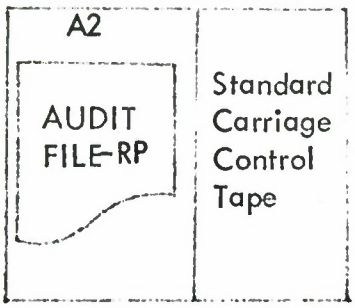
TAPES



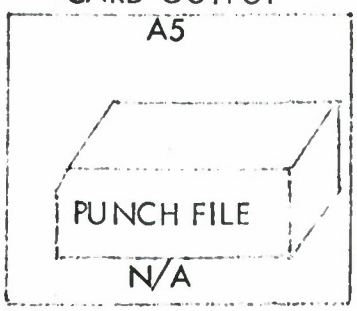
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



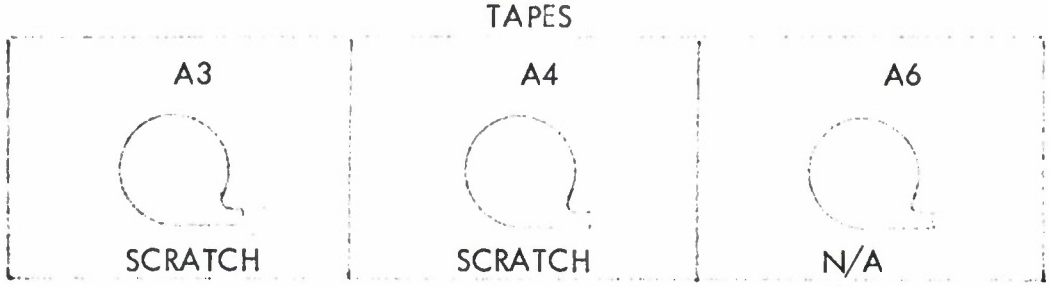
JCVS USAGE FORM

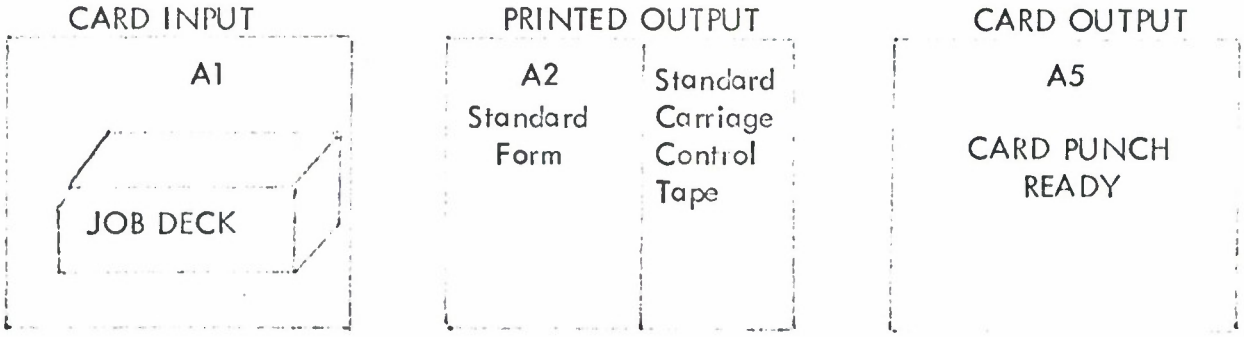
Function: INIPOP1

Computer: GE-635

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

```
(1)         (8)             (16)
$           IDENT           3154203,DATDY
$           COBOL
$           INCODE IBMC

            (COBOL Source Program Deck - INIPOP)

$           EXECUTE         DUMP
$           LIMITS          15,32000
$           SYSOUT          A2
$           TAPE            A3,X3S
$           TAPE            A4,X4R
$           SYSOUT          A5
$           DATA            A1

            (Control Card - IP)
            (Current File- PF Deck)

$           ENDJOB
***EOF
```

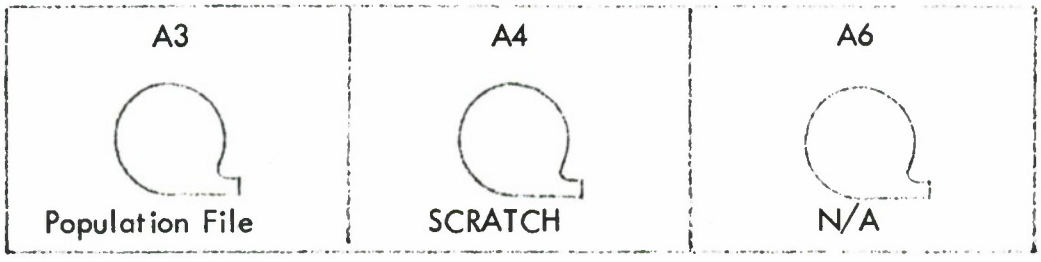
JCVS USAGE FORM

Function: INIPOP1
Computer: GE-635

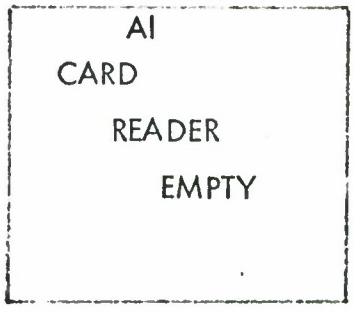
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

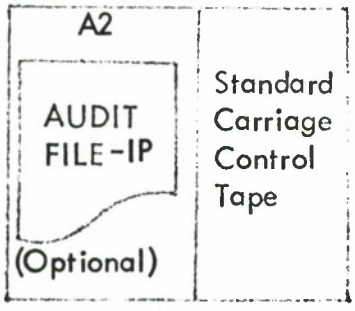
TAPES



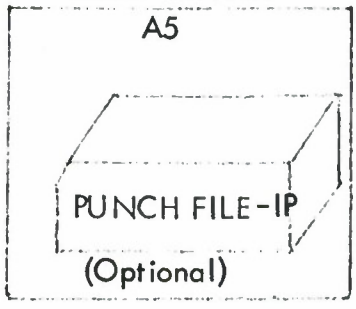
CARD INPUT



PRINTED OUTPUT

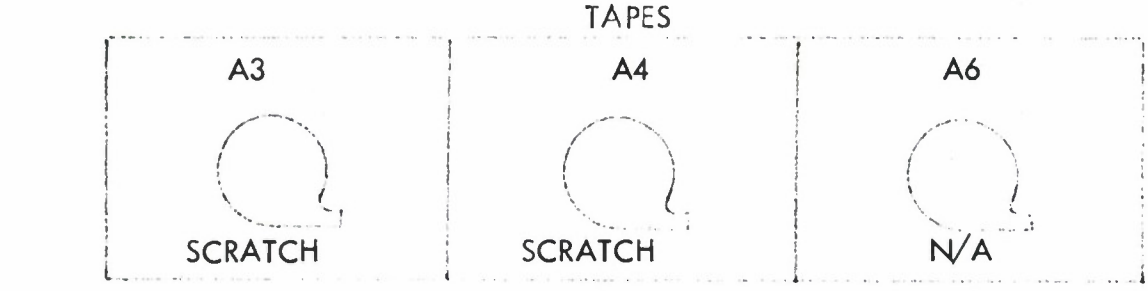


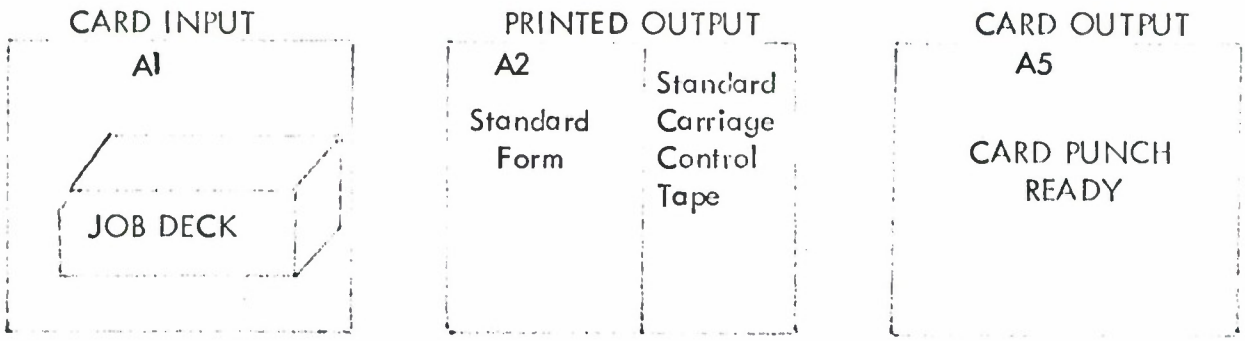
CARD OUTPUT



JCVS USAGE FORM

Function: INIPOP1
Computer: GE-635
Operating Philosophy: Load Binary Deck and Go
Stage: INPUT





JOB DECK STRUCTURE

```
(1)         (8)         (16)

$           IDENT       3154203, DATDY
$           OPTION      COBOL

          ( Binary Program Deck - INIPOP)

$           EXECUTE     DUMP
$           LIMITS      15,32000
$           SYSOUT      A2
$           TAPE        A3,X3S
$           TAPE        A4,X4R
$           SYSOUT      A5
$           DATA        A1

            (Control Card - IP)
            (Current File - PF Deck)

$           ENDJOB
***EOF
```

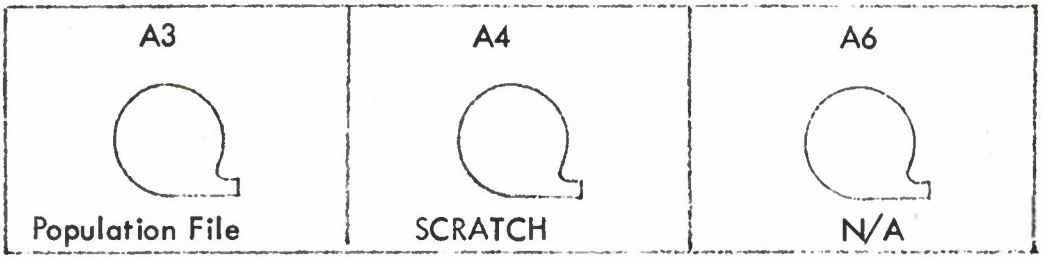
JCVS USAGE FORM

Function: INIPOP1
Computer: GE-635

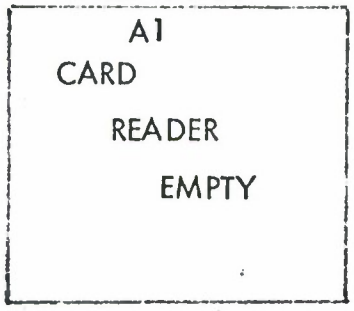
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

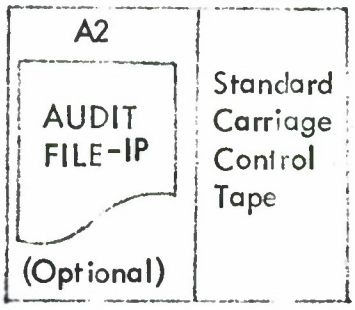
TAPES



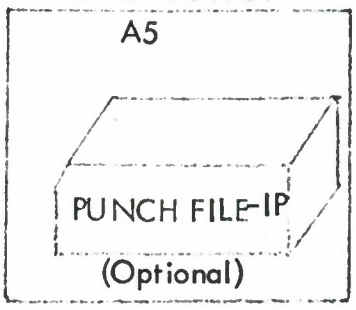
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



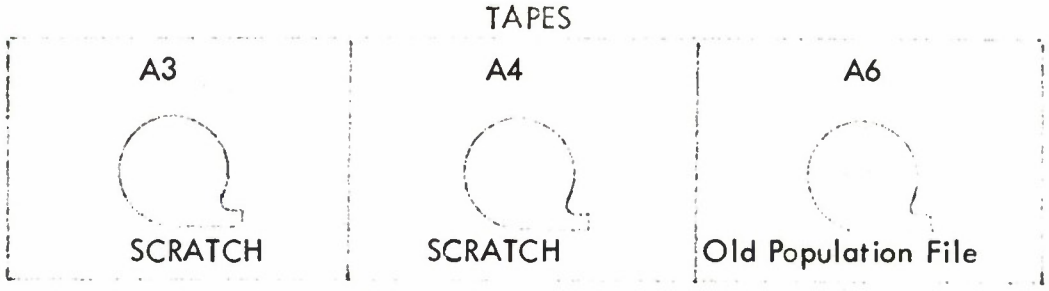
JCVS USAGE FORM

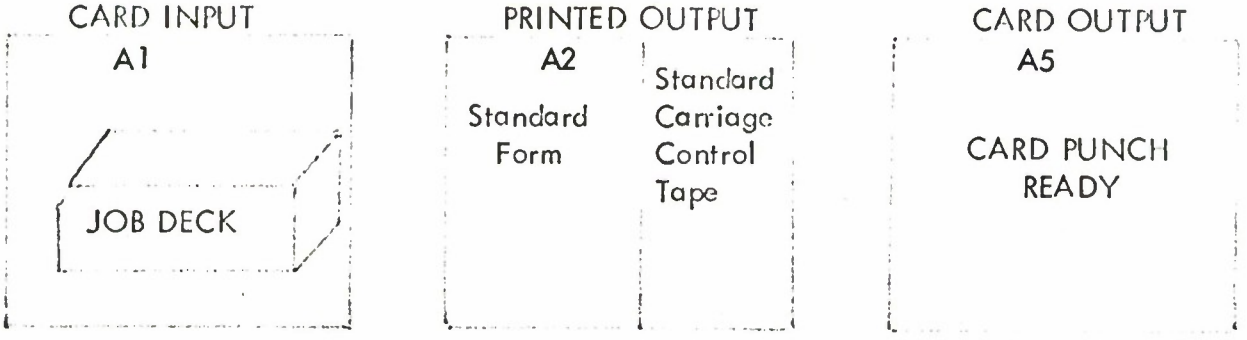
Function: INIPOP 2

Computer: GE-635

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

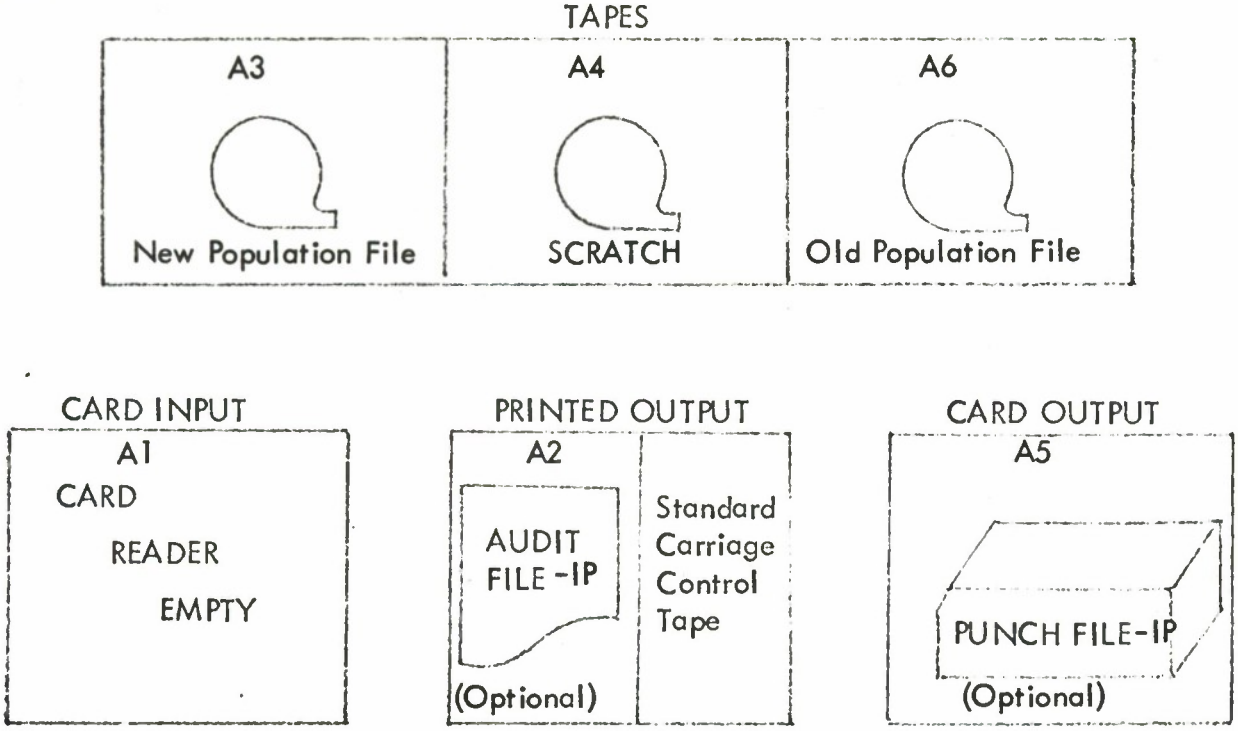
```
(1)          (8)          (16)

$            IDENT        3154203,DATDY
$            COBOL
$            INCODE       IBMC

   (COBOL Source Program Deck - INIPOP)

$            EXECUTE      DUMP
$            LIMITS       15,32000
$            SYSOUT       A2
$            TAPE         A3,X3S
$            TAPE         A4,X4R
$            TAPE         A6,X6R,,POPFILE1,,XXXX          XXXX = reel number
$            DATA         A1

             (Control Card - IP)
             (Current File - PF Deck)

$            ENDJOB
***EOF
```

JCVS USAGE FORM

Function: INIPOP2
Computer: GE-635

Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT



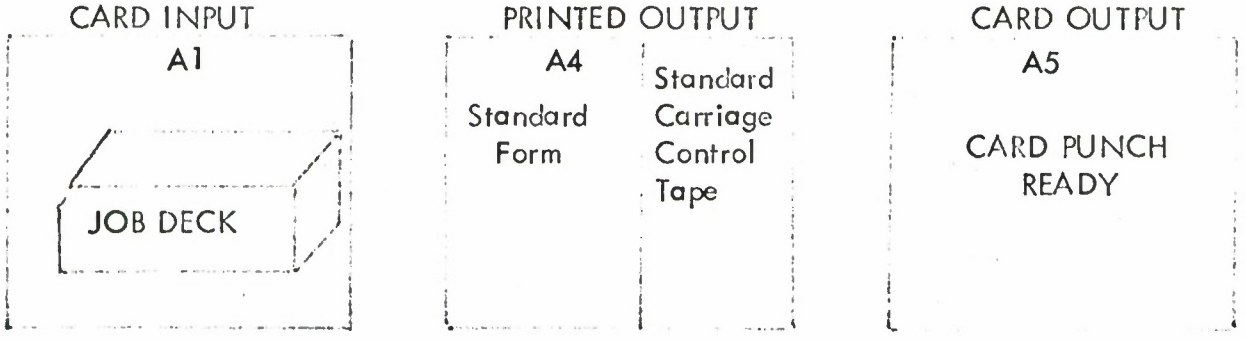
JCVS USAGE FORM

Function: INIPOP 2

Computer: GE-635

Operating Philosophy: Load Binary Deck and Go

Stage: INPUT



JOB DECK STRUCTURE

```
(1)         (8)         (16)

$           IDENT       3154203,DATDY
$           OPTION      COBOL

            (Binary Program Deck - INIPOP)

$           EXECUTE     DUMP
$           LIMITS      15,32000
$           SYSOUT      A2
$           TAPE        A3,X3S
$           TAPE        A4,X4R
$           TAPE        A6,X6R,,POPFILE1,,XXXX          XXXX = reel number
$           SYSOUT      A5
$           DATA        A1

            (Control Card - IP)
            (Current File - PF Deck)

$           ENDJOB
***EOF
```

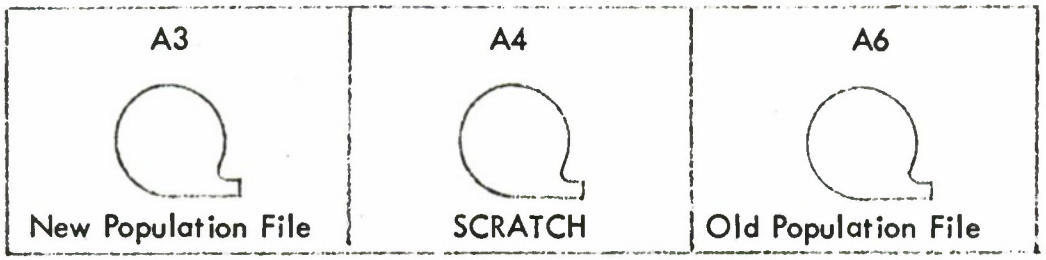
JCVS USAGE FORM

Function: INIPOP2
Computer: GE-635

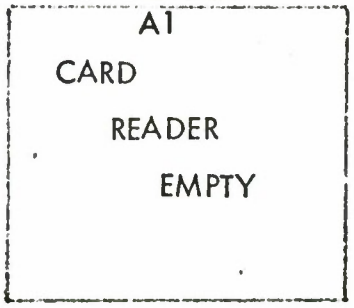
Operating Philosophy: Load Binary Deck and Go

Stage: OUTPUT

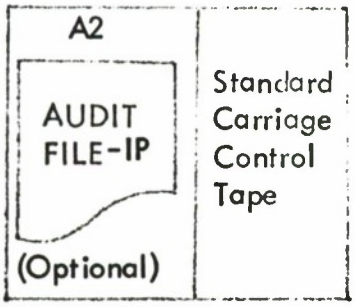
TAPES



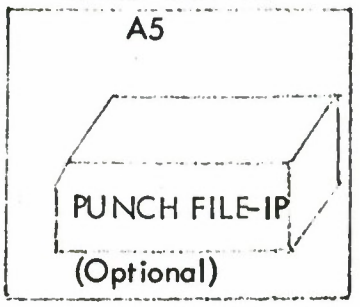
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



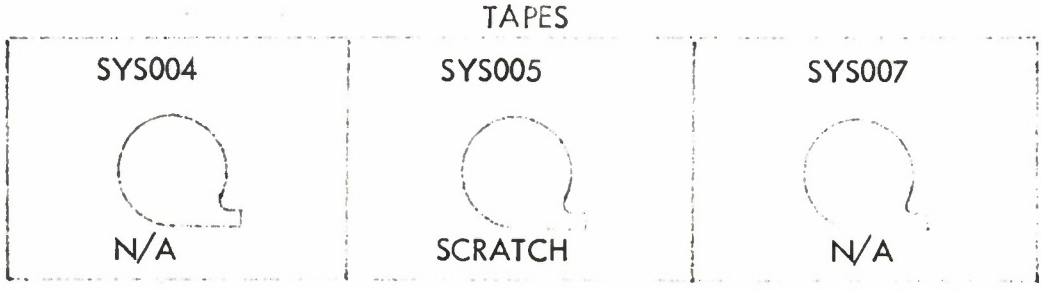
JCVS USAGE FORM

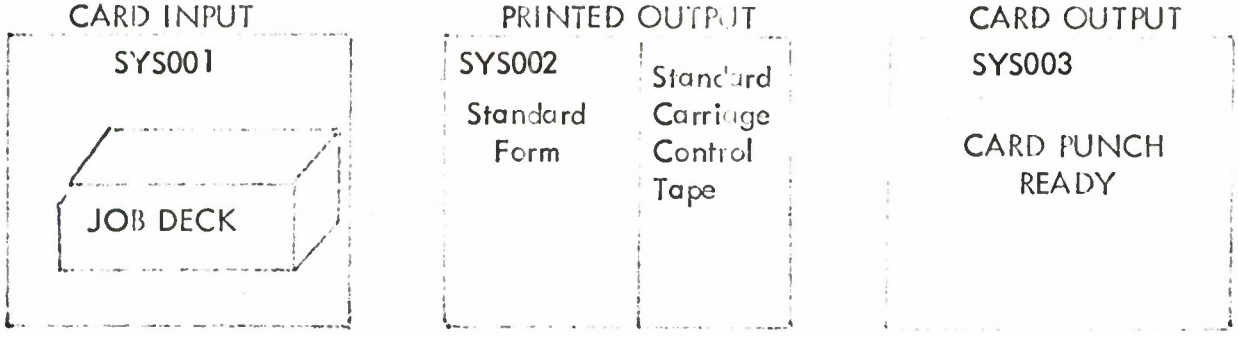
Function: POPFM1

Computer: IBM 360-50

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

```
/'POPFM1, JOB (799,028,010,1084,10,5),ANTCHAGNO,MSGLEVEL = 1
/'SI EXEC COBFCLG
/'COB.SYSIN DD*

          (COBOL Source Program Deck - POPFM)

/'GO. SYS002 DD SYSOUT = A
/'GO. SYS003 DD SYSOUT = B
/'GO. SYS005 DD UNIT = 2400, LABEL =(,NL), DISP = (,KEEP),DSN = POPFILE1
/'GO. SYSDUMP DD SYSOUT = A
/'GO. SYS001 DD*

          (Control Card - PF)
          (Current File - PF2 Deck)

/*
```

JCVS USAGE FORM

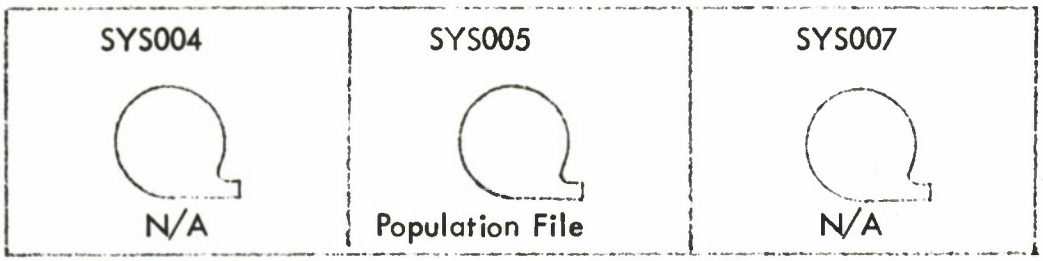
Function: POPFM1

Computer: IBM 360-50

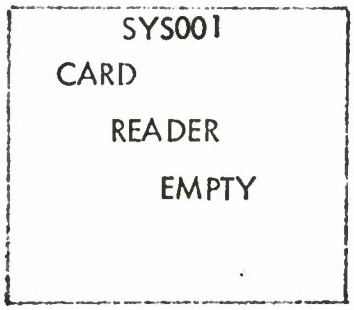
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

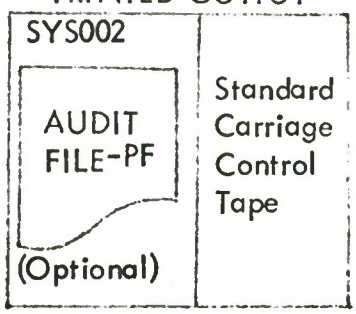
TAPES



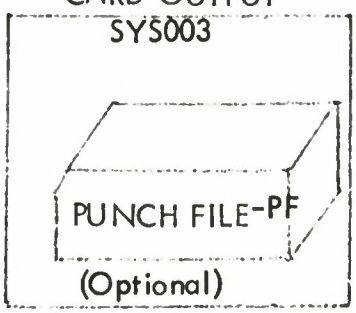
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



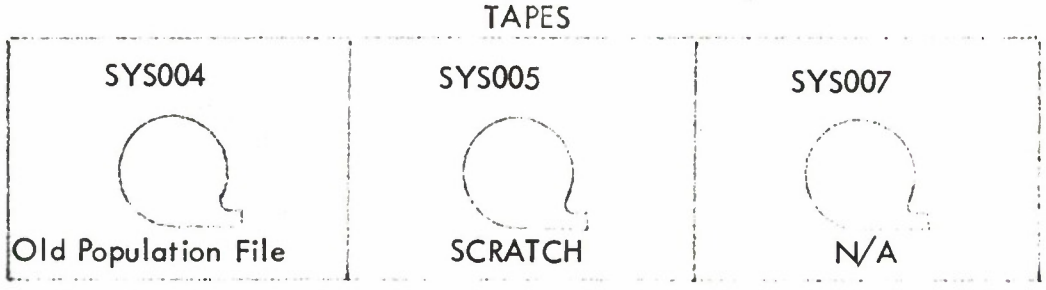
JCVS USAGE FORM

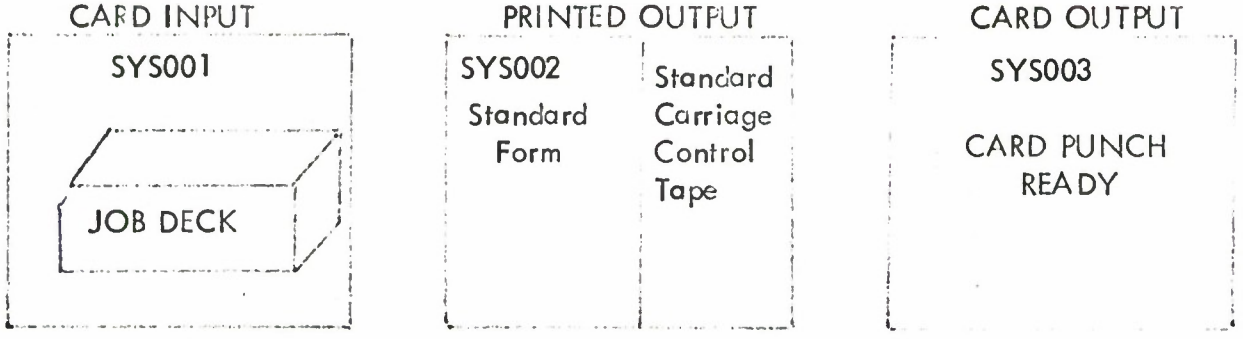
Function: POPFM2

Computer: IBM 360-50

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

```
//POPFM2 JOB (799,028,010,1084,10.5),ANTCHAGNO, MSGLEVEL = 1
//SI EXEC COBFCLG
//COB.SYSIN DD*

          (COBOL Source Program Deck - POPFM)

//GO.SYS002 DD SYSOUT = A
//GO.SYS003 DD SYSOUT = B
//GO.SYS004, DD UNIT = 2400,LABEL = (,NL), DISP =OLD, VOL = SER= 000649
//GO.SYS005, DD UNIT = 2400,LABEL = (,NL),DISP = (,DELETE)
//GO.SYSDUMP DD SYSOUT = A
//GO.SYS001 DD*

          (Control Card - PF)
          (Current File - PF2 Deck)

/*
```

JCVS USAGE FORM

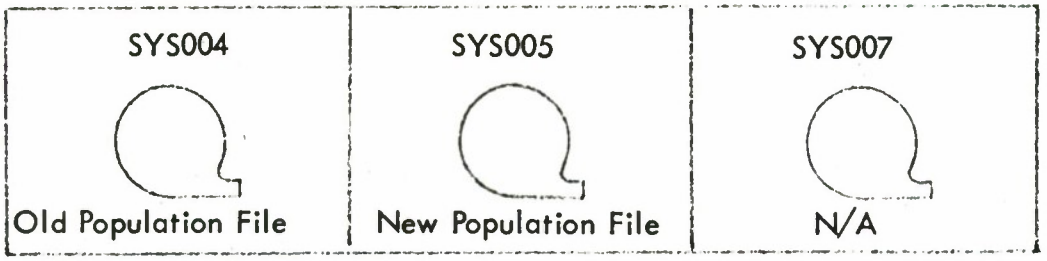
Function: POPFM2

Computer: IBM 360-50

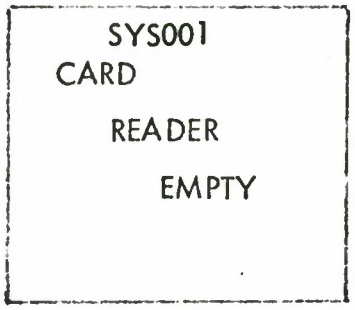
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

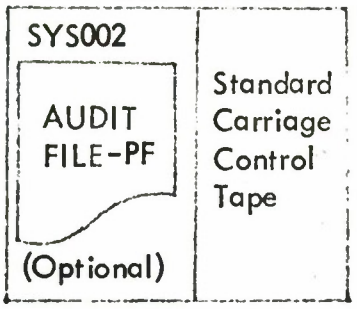
TAPES



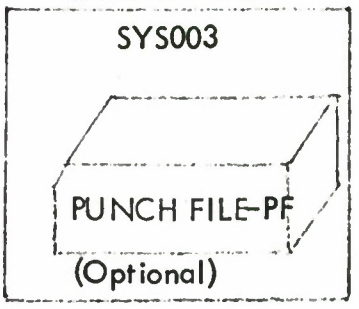
CARD INPUT



PRINTED OUTPUT

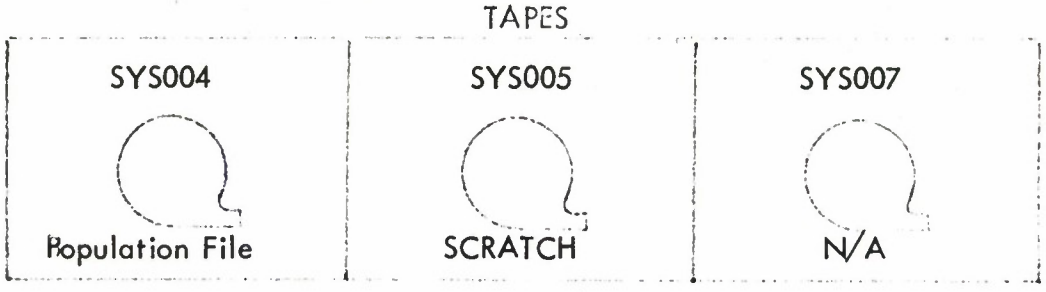


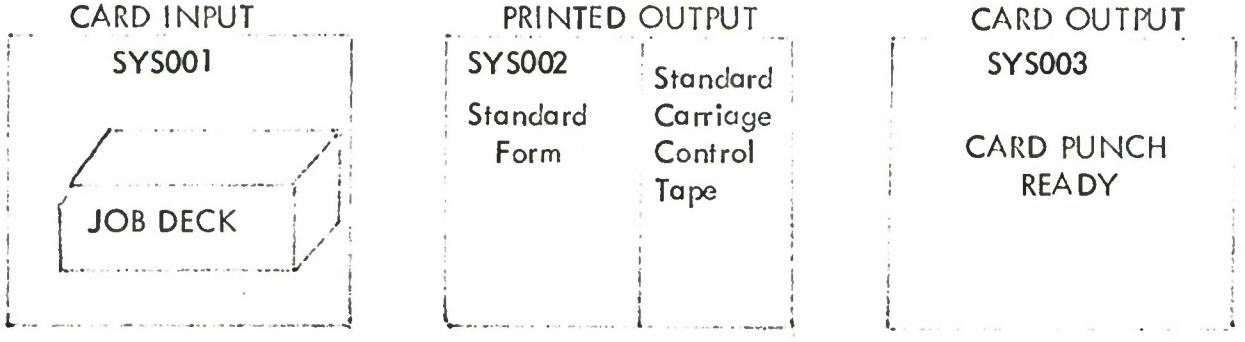
CARD OUTPUT



JCVS USAGE FORM

Function: SELECT
Computer: IBM 360-50
Operating Philosophy: Compile Source Program and Go
Stage: INPUT





JOB DECK STRUCTURE

```
//SELECT JOB (799,028,010,1084,10,5), ANTCHAGNO, MSGLEVEL = 1
//S1 EXEC COBFCLG
//COB.SYSIN DD*

          (COBOL Source Program Deck - SJCVS)

//GO.SYS002 DD SYSOUT = A
//GO.SYS003 DD SYSOUT = B
//GO.SYS004 DD UNIT = 2400, LABEL = (,NL), DISP = OLD, VOL = SER = 000649
//GO.SYS005 DD UNIT = 2400, LABEL = (,NL), DISP = (,KEEP), DSN = JOVSP
//GO.SYSDUMP DD SYSOUT = A
//GO.SYS001 DD*

          (Control Card - S)
          (Test Selection File Deck)

/*
```

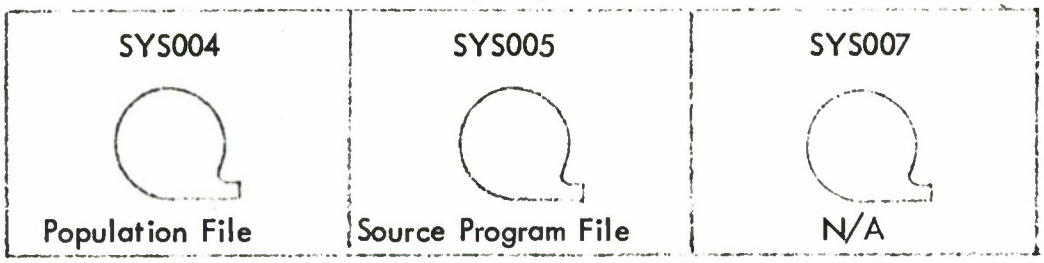
JCVS USAGE FORM

Function: SELECT
Computer: IBM 360-50

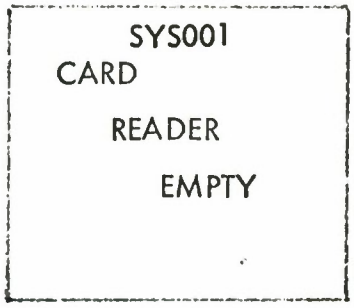
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

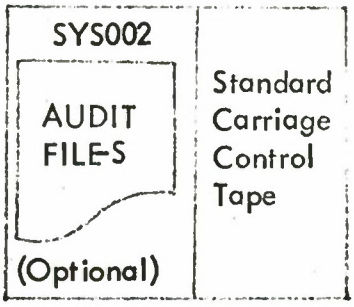
TAPES



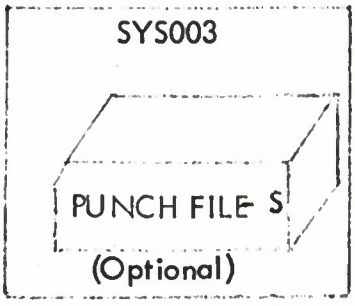
CARD INPUT



PRINTED OUTPUT

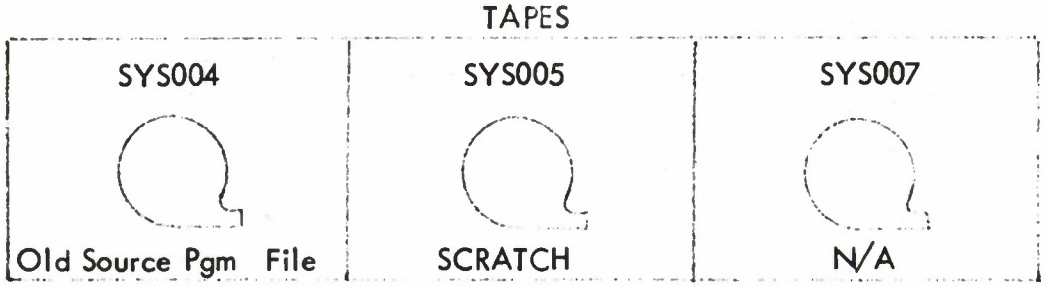


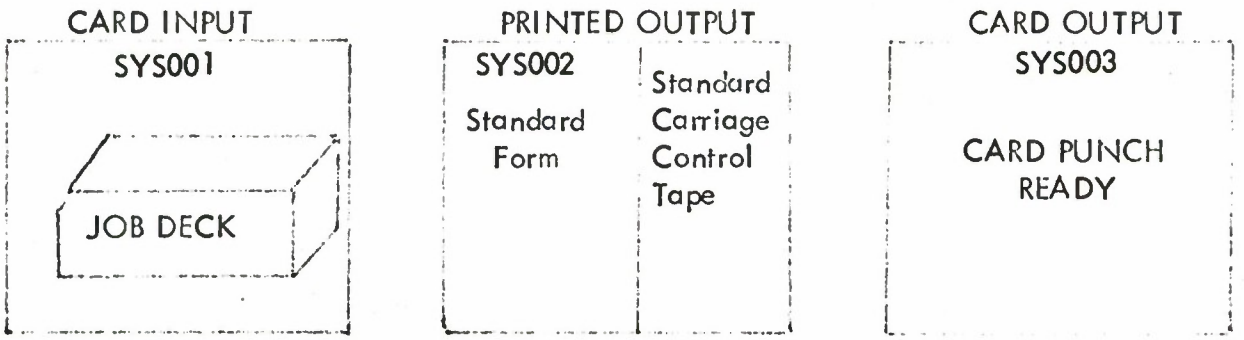
CARD OUTPUT



JCVS USAGE FORM

Function: SOPMM
Computer: IBM 360-50
Operating Philosophy: Compile Source Program and Go
Stage: INPUT





JOB DECK STRUCTURE

```
//SOPMM JOB (799,028,010,1084,10,5), ANTCHAGNO, MSGLEVEL = 1
//SI EXEC COBFCLG
//COB.SYSIN DD*

          (COBOL SOURCE PROGRAM DECK)

//GO.SYS002 DD SYSOUT = A
//GO.SYS003 DD SYSOUT = B
//GO.SYS004 DD UNIT = 2400,LABEL = (,NL),DISP-OLD,VOL = SER = 000570
//GO.SYS005 DD UNIT = 2400,LABEL = (,NL),DISP = (,DELETE)
//GO.SYSDUMP DD SYSOUT = A
//GO.SYS001 DD*

          (Control Card -SP)
          (Current File -SP2 Deck)

/*
```

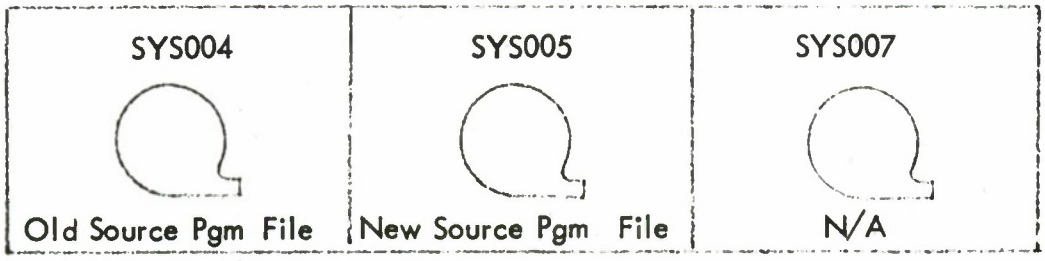
JCVS USAGE FORM

Function: SOPMM
Computer: IBM 360-50

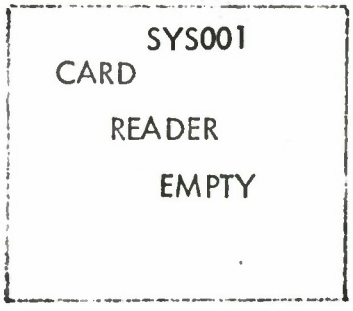
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

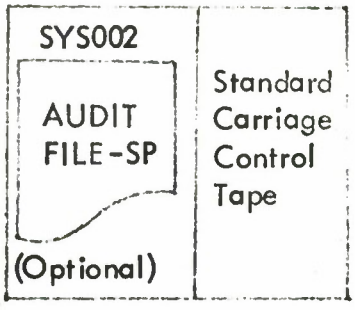
TAPES



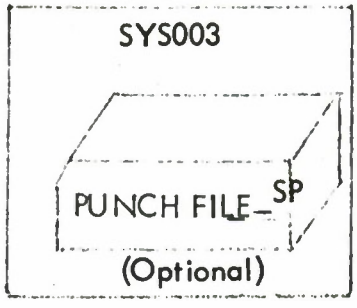
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



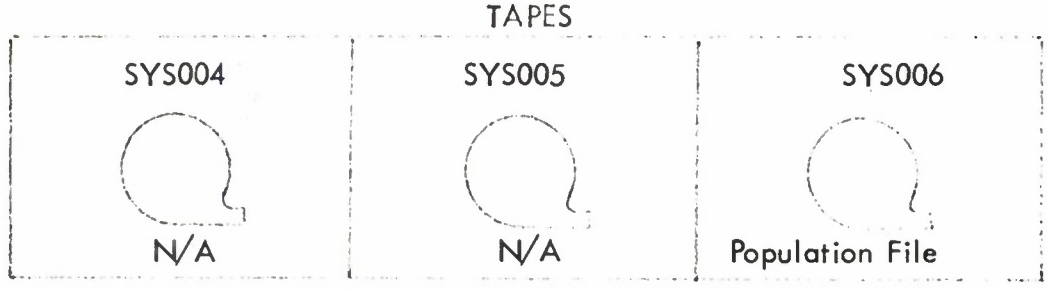
JCVS USAGE FORM

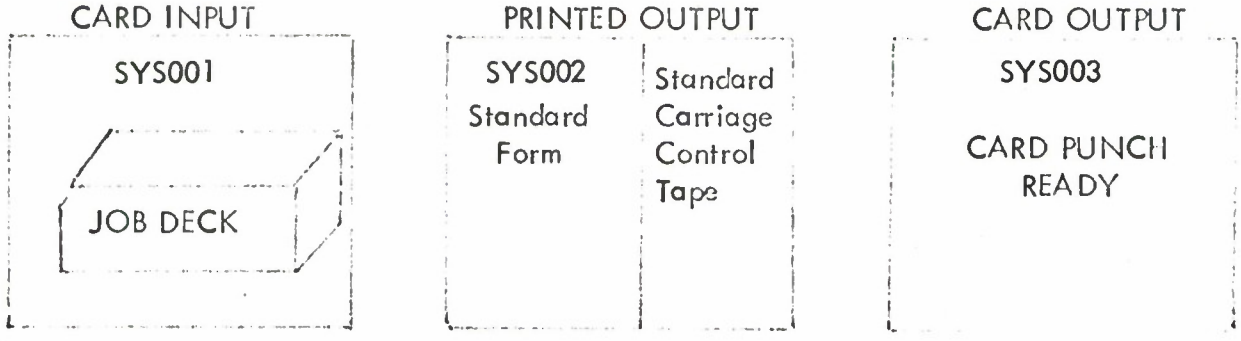
Function: JCVSRP

Computer: IBM 360-50

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

```
//JCVSRP JOB (799,028,010,1084,10,5),ANTCHAGNO, MSGLEVEL = 1
//S1 EXEC COB FCLG
//COB.SYSIN DD*

          (COBOL Source Program Deck - JCVSRP)

//GO.SYS002 DD SYSOUT = A
//GO.SYS002 DD UNIT = 2400,LABEL = (,NL),DISP = OLD,VOL = SER = 000649
//GO.SYSDUMP DD SYSOUT = A
//GO.SYS001 DD*

          (Control Card - RP)


/*
```

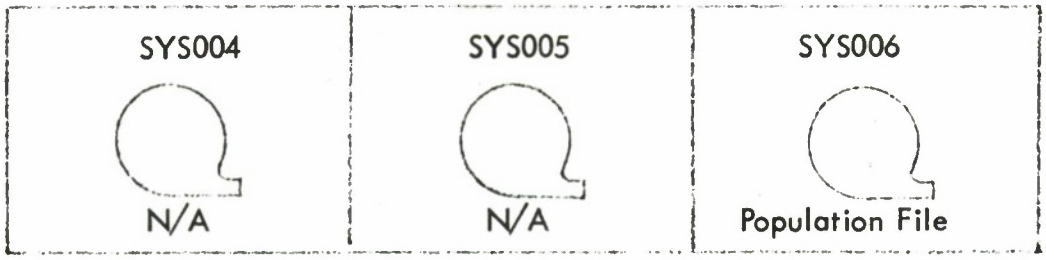
JCVS USAGE FORM

Function: JCVSRP
Computer: IBM 360-50

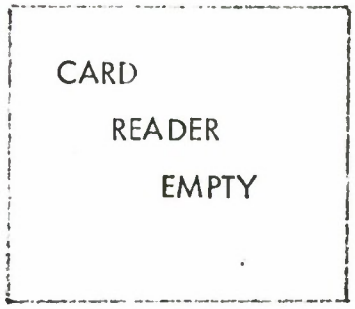
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

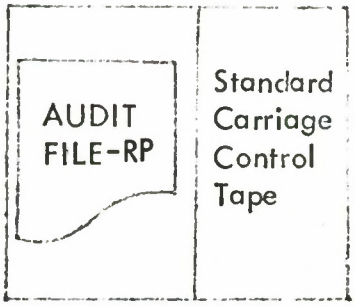
TAPES



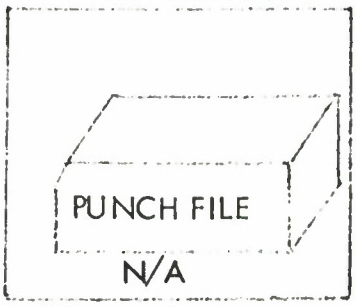
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



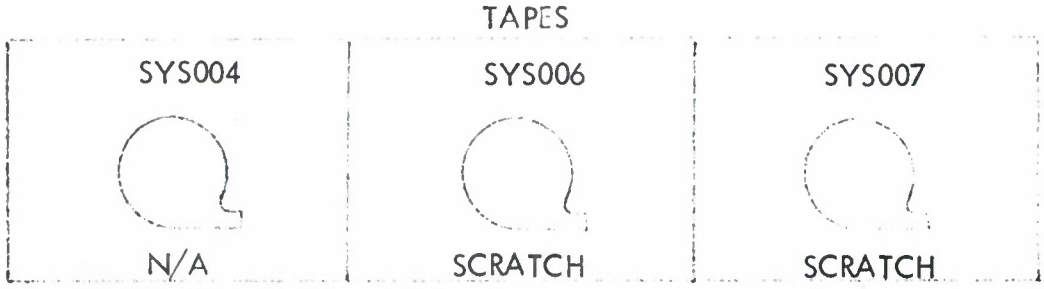
JCVS USAGE FORM

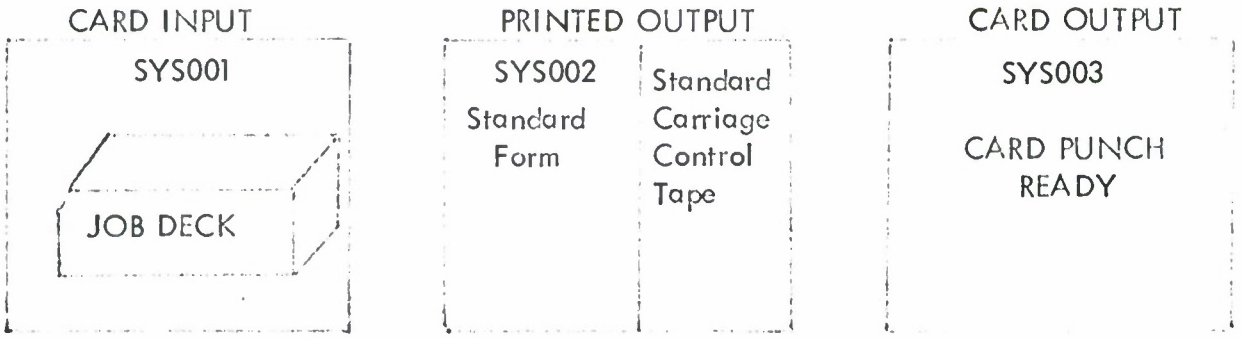
Function: INIPOP1

Computer: IBM 360-50

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

```
//INIPOP,JOB,(799,028,OLD,1084,10,5), ANTCHAGNO,MSGLEVEL=1
//SI,EXEC COBFCLG
//COB.SYSIN DD*

              (COBOL Source Program Deck - INIPOP)

//GO.SYS002  DD SYSOUT = A
//GO.SYS003  DD SYSOUT = B
//GO.SYS006  DD UNIT = (2400,DEFER),LABEL = (,NL),DISP = (,DELETE),
//              DSN = MSTRFILE
//GO.SYS007  DD UNIT = 2400,LABEL = (,NL),DISP = (,DELETE)
//GO.SYSDUMP DD SYSOUT = A
//GO.SYS001 DD**

              (Control Card - IP)
              (Current File - PF Deck)

/*
```

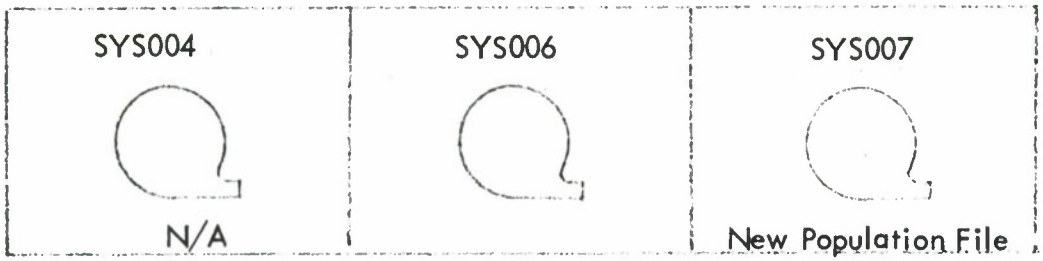
JCVS USAGE FORM

Function: INIPOP1
Computer: IBM 360-50

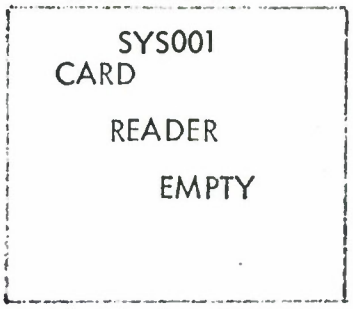
Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

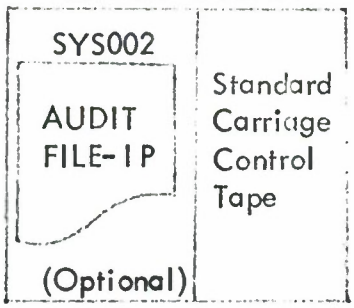
TAPES



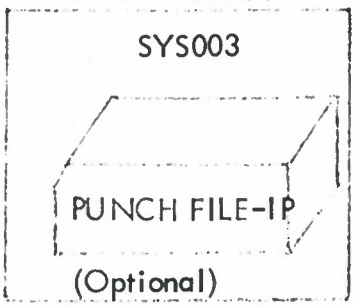
CARD INPUT



PRINTED OUTPUT



CARD OUTPUT



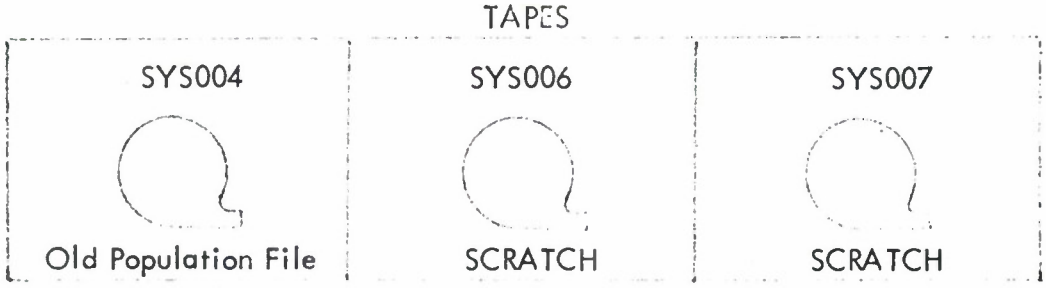
JCVS USAGE FORM

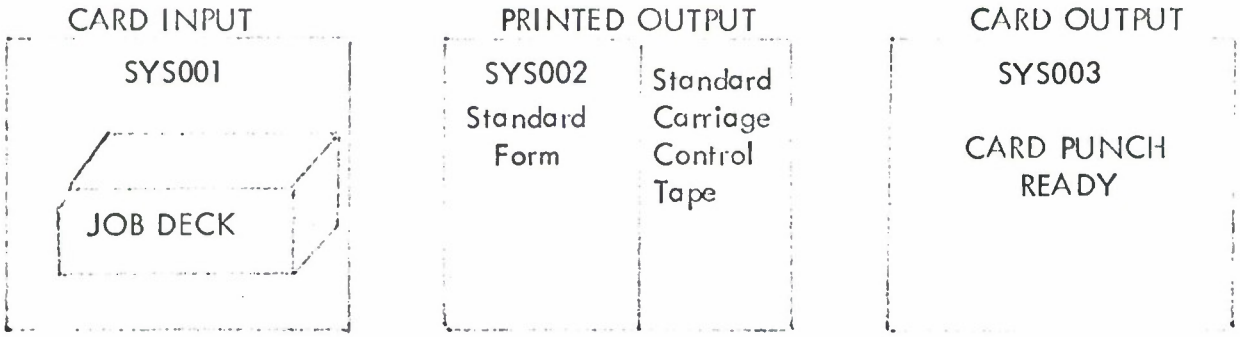
Function: INIPOP2

Computer: IBM 360-50

Operating Philosophy: Compile Source Program and Go

Stage: INPUT





JOB DECK STRUCTURE

```
//POPFM2 JOB (799,028,010,1084,10,5), ANTCHAGNO, MSGLEVEL = 1
//SI EXEC COBFCLG
//COB.SYSIN DD*
```

(COBOL Source Program Deck - INIPOP)

```
//GO.SYS002  DD SYSOUT = A
//GO.SYS003  DD SYSOUT = B
//GO.SYS006  DD UNIT = 2400 LABEL = (, NL), DISP = OLD, VOL = SER 000649
//GO.SYS007  DD UNIT = 2400, LABEL = (, NL), DISP = (, DELETE)
//GO.SYSDUMP DD SYSOUT = A
//GO.SYS001 DD*
```

(Control Card - IP)

```
/*
```

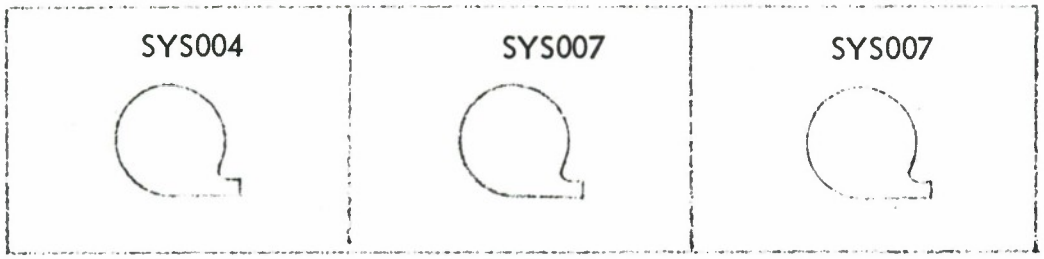
JCVS USAGE FORM

Function: INIPOP2
Computer: IBM 360-50

Operating Philosophy: Compile Source Program and Go

Stage: OUTPUT

TAPES



CARD INPUT

CARD
READER
EMPTY

PRINTED OUTPUT

AUDIT
FILE-IP
(Optional)

Standard
Carriage
Control
Tape

CARD OUTPUT

PUNCH FILE-IP
(Optional)

APPENDIX II

SYSTEM HEADER CARD 2

This appendix contains the System Header 2 cards which contain the JCVS model number and the operating system name for each of the five computers.

GE-635

OVIAL COMPILER VALIDATION SYSTEM 1 GECOS

CDC-6400

OVIAL COMPILER VALIDATION SYSTEM 1 SCOPE

B-5500

OVIAL COMPILER VALIDATION SYSTEM 1 MCP

UNIVAC-1108

OVIAL COMPILER VALIDATION SYSTEM 1 EXEC2

IBM 360-50

OVIAL COMPILER VALIDATION SYSTEM 1 HASP

System Header Card 2

APPENDIX III

ENVIRONMENTAL HARDWARE CARDS

This appendix contains a listing of the three environmental hardware cards associated with each of the five computers. These cards contain tape designations, core sizes and print control character designations when applicable.

GE-635

A1	A2 FOR LISTING	65K	0001A00
A5 FOR CARDS	A3		0001A00
A4	A6		0001A00

CDC-6400

INPUT	OUTPUT	01137K	0001A00
PUNCH	TAPE1		0001A00
TAPE2	TAPE3		0001A00

B-5500

READER	PRINTER	65K	0001A00
PUNCH	TAPE		0001A00
TAPE	TAPE		0001A00

UNIVAC-1108

CARD-READER-EIGHTY	PRINTER	65K	0001A00
CARD-PUNCH-EIGHTY	UNISERVO A		0001A00
UNISERVO B	UNISERVO C		0001A00

IBM 360-50

'SYS001' UNIT-RECORD 2540R	'SYS002' UNIT-RECORD 2540P	65K	0001A00
'SYS003' UNIT-RECORD 2540P	'SYS004' UTILITY 2400 UNIT		0001A00
'SYS005' UTILITY 2400 UNIT	'SYS006' UTILITY 2400 UNIT		0001A00

APPENDIX IV

ENVIRONMENTAL SOFTWARE CARDS

This appendix contains a listing of the operating system control cards and the JOVIAL control cards required to signify a JOVIAL source program.

GE-635

```
      IDENT    3154203,DATDY
      JOVIAL
      FORTRAN
      EXECUTE  DUMP
      LIMITS   15,35000
      ENDJOB
*EOF
```

Environmental Software Cards

APPENDIX V

TYPICAL MODULES

This appendix contains a listing of a few typical Population File modules.

```
''MODULE 5220 - CED 2454 ''                                          5220A00
''TEST USE OF FLOATING CONSTANTS,VARIABLES''                         5220J00
ITEM FA5220 F P 1.0$  ITEM FB5220 F P 4.0$                           5220J00
ITEM FC5220 F P 0.0$                                                 5220J00
                                                                     5220J00
        IFEITH FA5220 EQ FB5220$ GOTO LZ5220$                        5220J00
        ORIF 1.0 EQ FA5220$ FC5220=3.0$                              5220J00
        ORIF FA5220 EQ FC5220$ GOTO LZ5220$                          5220J00
LA52209 ORIF 1$ GOTO LZ5220$ END                                     5220J00
        IFEITH FB5220 EQ 1.0$ GOTO LA5220$                           5220J01
        ORIF FA5220 EQ 1.0$ GOTO LB5220$                             5220J01
        ORIF 1$ GOTO LZ5220$ END                                     5220J01
        GOTO LZ5220$                                                 5220J01
LB52209 IFEITH 1.0 EQ 2.0$ FC5220=1.0$                               5220J01
        ORIF 2.0 EQ FA5220$ FC5220=1.0$                              5220J01
        ORIF FB5220 EQ 2.0$ GOTO LC5220$                             5220J01
        ORIF 1$ GOTO LZ5220$ END ''ERROR IF HERE''                   5220J01
        GOTO LZ5220$                                                 5220J01
LC52209 GOTO LY5220$                                                 5220J01
LZ5220.                                                              5220J02
OUT1=40H( MODULE 5220 TEST FAILED. CED2454        )$                 5220J02
OUTERR(OUT1)$ GOTO LX5220$ ''EXIT''                                  5220J02
LY5220.                                                              5220J02
OUT1=40H( MODULE 5220 TEST SUCCESSFUL.            )$                 5220J02
OUTERR(OUT1)$                                                        5220J02
LX5220.                                                              5220J02
''MODULE 5230 - CED 2454 ''                                          5230A00
''TEST USE OF STATUS CONSTANTS,VARIABLES''                           5230J00
ITEM SA5230 S V(A) V(B) V(C) P V(B)$                                 5230J00
ITEM SB5230 S V(X) V(Y) V(Z) P V(Z)$                                 5230J00
ITEM SC5230 S V(NO) V(YES) P V(YES)$                                 5230J00
ITEM SD5230 S V(NO) V(YES) V(MAYBE) P V(YES)$                        5230J00
        IFEITH V(A) EQ SA5230$ GOTO LZ5230$                          5230J00
        ORIF SB5230 EQ V(X)$ SC5230=V(NO)$                           5230J00
        ORIF SB5230 EQ SA5230$ GOTO LZ5230$                          5230J00
        ORIF 1$ GOTO LA5230$ END                                     5230J01
        GOTO LZ5230$                        ''ERROR''                5230J01
LA5230. IFEITH SD5230 EQ V(YES)$ GOTO LB5230$                        5230J01
        ORIF V(A) EQ SA5230$ GOTO LZ5230$                            5230J01
LB5230. ORIF V(YES) EQ SD5230$ SC5230=V(NO)$                         5230J01
        ORIF 1$ GOTO LZ5230$ END      ''ERROR''                      5230J01
        IFEITH SB5230 EQ V(Z)$ GOTO LC5230$                          5230J01
        ORIF 1$ GOTO LZ5230$ END      ''ERROR''                      5230J01
LC52309 GOTO LY5230$                                                 5230J01
LZ5230.                                                              5230J01
OUT1=40H( MODULE 5230 TEST FAILED. CED2454        )$                 5230J02
OUTERR(OUT1)$ GOTO LX5230$   ''EXIT''                                5230J02
LY5230.                                                              5230J02
OUT1=40H( MODULE 5230 TEST SUCCESSFUL.            )$                 5230J02
OUTERR(OUT1)$                                                        5230J02
LX5230.                                                              5230J02
''MODULE 5240 - CED 2454 ''                                          5240A00
''TEST USE OF TRANSMISSION CONSTANTS,VARIABLES''                     5240J00
ITEM TA5240 T 2 P 2T(AA)$                                            5240J00
ITEM TB5240 T 2 P 2T(BB)$                                            5240J00
ITEM TC5240 T 2 P 2T(  )$ ITEM TD5240 T 2$                           5240J00
```

APPENDIX VI

This appendix defines the test hierarchy for the JCVS as well as some highlights of JOVIAL as a language and validation in general.

APPENDIX VI

General

This appendix describes the development philosophy of the JOVIAL J3 Population File, including a brief history of the JOVIAL language; an exposition of all validation concepts used in the development of the Population File; the JOVIAL language organization used to identify features to be tested; the JOVIAL language Test Hierarchy; and problems encountered in the development of this file.

Validation

A JOVIAL compiler is said to be validated if each feature conforms to the individual language specifications called features as described in the AFM 100-24. Each feature has been individually considered in terms of its intent and one or more tests have been developed exercising the various options provided by this feature.

Every option provided by every feature in the language is exercised at least once in the tests comprising the Population File. When combinations of feature options were required to insure the validity of a feature, in several instances only a subset of the possible combinations were included in the Population File.

JOVIAL History

The JOVIAL language was originally developed in 1958, four years after the development of the first programming language, FORTRAN. It is a procedure oriented higher-order programming language. JOVIAL, a derivative of ALGOL 58, was designed specifically to describe computerized solutions to command and control problems.

As stated by AFM 100-24,

> "The prime motivation for the development of JOVIAL was the desire to have a common, powerful, easily understandable and mechanically translatable programming language suitable for wide-range applications."

In addition to the above requirements, the language was to adhere to the following design goals?

1. Centralized data communication facilities
2. Machine independence
3. Logical and Algebraic expresseion ccpabilities
4. Symbol manipulation capabilities

5. Readability
6. Conciseness
7. Training Simplicity
8. Ease of maintainence

Based upan the aforementioned requirements and goals, the JOVIAL language greatly enhances the prablem definitional capabilities of the programmer. The following paragraphs illustrate the wisdom of the JOVIAL design.

Command and control problems are in general extremely large in terms af the data base to be gathered, manipulated and reported; and the variety of computations to be performed on the data base. Cansequently, the programming system necessary to solve this prablem is so vast that several hundred programmers may be required to perform the individual programming tasks. Because of the number of individual programs and programmers involved in a command and control development, program/programmer communication becomes a critical problem.

In order to alleviate this situation, a Communication Pool (COMPOOL) was develaped which serves as a central souce of data description. Centralizing all global data descriptions facilitates changing data item parameters and autamatically reflecting these changes thraughaut the machine language programs. This feature of the JOVIAL language alane has saved enormous amounts of time and money in several command and cantrol system developments.

Application Requirements

Programming languages are created in arder to respond to common sub-solutians within applicatian areas. Pragramming languages supply capabilities that satisfy these camman sub-solutions while suppressing the repetion and details of salutian.

Many af these capabilities are present in most languages and pravide for general applicatian requirements such as:

1. Pragram Cantral
2. Informatian Transfer
3. Input/Output Communication
4. Arithmetic Operations
5. Data Item Definitions
6. Storage Allocation - Static

to name a few.

Additianal pawer may be provided by a language by adding capabilities of a general nature that make the language useful problem solving tool for a broader class of problems or by adding more extensive capabilities but ariented tawards specific area.

Generally oriented features:

1. Algebraic Expression Evaluation
2. Logical Expressions Evaluation
3. Data Structure Definitions

Specifically oriented features:

1. Formula Manipulation
2. List Processing

Language Organization

The JOVIAL language was developed to respond to command and control applications. Each feature of the language may be interpreted as a language response to a programming function required by a command and control applications programmer. Using this notion as a point of departure, the JOVIAL language has been organized into the following programming functions in order to organize the identification of features to be tested.

1. Data Concepts
   - 1.1 Internal Data Concepts
     - 1.1.1 Data Definitions
       - 1.1.1.1 Constant Formulation
         - Integer - I
         - Fixed Point -A
         - Floating Point - F
         - Octal - O
         - Dual - D
         - Transmission Code - T
         - Hollerith - H
         - Boolean - B
         - Status - S
       - 1.1.1.2 Simple Data Definitions
         - Integer - I
         - Fixed Point - A
         - Floating Point - F
         - Dual - D
         - Transmission Code - T
         - Hollerith - H
         - Boolean - B
         - Status - S
       - 1.1.1.3 Structured Data Definitions
         - Tables
         - Arrays

1.1.1.4 Control Definitions
Item Switch
Index Switch

1.1.2 Data Referencing

1.1.2.1 Simple Items

1.1.2.2 Data Structure Items
Table Items
Array Items

1.1.2.3 Data Structure
Table Entries

1.1.2.4 Special Referencing
ALL
BIT
BYTE
CHAR
ENT
ENTRY
LOC
MANT
NENT
NWDSEN
ODD
POS

1.2 External Data Concepts

2. Procedure Concepts

2.1 Procedure Formations

2.1.1 Formulas

2.1.1.1 Numeric

2.1.1.2 Boolean

2.1.2 Relations

2.2 Program Organization Statements

2.2.1 PROGRAM

2.2.2 Subprogram Organization

2.2.2.1 Procedures
User Defined
PROC
CLOSE
Language Defined
REMQUO

2.2.2.2 Functions
User Defined
Language Defined
ABS
REM

2.2.3 RETURN

2.3 Executable Statements
- 2.3.1 Control Statements
  - 2.3.1.1 Unconditional Control Transfers
    - GOTO
    - STOP
  - 2.3.1.2 Conditional Control Transfers
    - IF
    - IFEITH
    - ORIF
  - 2.3.1.3 Iteration Control
    - FOR
    - TEST
- 2.3.2 Input/Output Statements
  - INPUT
  - OPEN INPUT
  - SHUT INPUT
  - OUTPUT
  - OPEN OUTPUT
  - SHUT OUTPUT
- 2.3.3 Replacement Statements
  - 2.3.3.1 Assignment Statement
  - 2.3.3.2 Exchange Statement

2.4 Compiler Directing Concepts
- 2.4.1 DEFINE
- 2.4.2 LIKE
- 2.4.3 OVERLAY
- 2.4.4 MODE
- 2.4.5 DIRECT, JOVIAL

JCVS Testing Concepts

The following sections discuss briefly the scope of the JCVS and the tests selected for inclusion in the Population File.

JCVS Scope

For purposes of the JCVS, the JOVIAL system to be tested is assumed to consist of a processor that compiles standard JOVIAL source program statements called the JOVIAL compiler and all programs and subroutines used by the JOVIAL object code generated from standard JOVIAL statements. The JCVS is designed to test both the compilation and execution of specific JOVIAL features.

Test Assumptians - Data

The faregoing JOVIAL language arganizatian has guided the identificatian af language features ta be tested. In arder to validate the JOVIAL campiler ideally, each variant af a specific language feature shauld be validated. The validatian af each feature variant af the JOVIAL language, hawever, is nat always passible. For example, haw can one determine that any value stored in a floating paint item is truly stared as a floating paint number; how can ane determine that a fixed paint canstant has actually been canverted ta a fixed point binary paint canstant. Laaking at infarmatian as it resides in the internal starage medium, we may observe a string of bits, hawever, the interpretation af this content is incanclusive. Consequently, some af the features provided by the JOVIAL language are not susceptible to validation independently. These featur es are generally the mare basic notions in the language and will be used canstantly in the Test Modules camprising the Paulatian File. With repeated carrect usage of these basic concepts, it is haped that the credibility af their required implementation will be cansiderably impraved.

With these thaughts in mind, the fallawing aspects of the data definitianal capabilities af the JOVIAL language will not be tested independently and will be assumed present in the language and carrectly implemented:

1. The ability to specify any item type and have it retained according ta its defining attributes.
2. The ability ta farmulate any constant type and have it retained according to its defining attributes.
3. The ability ta specify any data structure type (table, array, etc.) and have it retained according to its defining attributes.

The JOVIAL language provides the user with a myriad af aptians ta form constants, simple items, tables, and arrays. There are sa many data defining attributes possible in JOVIAL that exercising each aptian in an independent test is quite impassible. As a campromise, the test repertaire will use a subset af data definitians that exercise, at least ance, all af the data attributes available to define data items and structures. In additian, the repertoire will utilize every variatian pravided ta formulate constants with the exception af the dual item definitians which will be exercised in part anly. It goes withaut saying that the farmation of acceptable JOVIAL symbols (names, labels, etc.) will be exercised every time a symbal is farmed.

Test Assumptians - Procedures

The JOVIAL language provides the user with the ability ta process farmulas and relations; it provides far program arganizatian and it pravides certain campiler directing features. Every variant af each af these features will be tested at least ance. Further substantiatian af the ability af a feature ta perform its intended function will be supplied by its carrect use as a suppart statement in other test modules.

With these thoughts in mind, the following aspects of the procedural capabilities of the JOVIAL language will be assumed to be present in the language and correctly implemented:

1. The ability to name a statement with a label.
2. The fact that normal procedural control passes from one JOVIAL statement to the next.

Test Hierarchy

Although the language organization serves to compartmentalize the various features of the language, it remains for the test hierarchy to specify the order in which these features are to be tested. This order must be specified to insure that the supporting JOVIAL statements used to compare test modules in which they participate may be validated.

A further ordering must be prescribed when testing out data and procedural language elements. Since procedural statements, for the most part, make reference to pieces of data, it seems reasonable to assume that data declarations should be validated before procedural statements. As a general rule, when a data concept is to be validated, it will be defined, structured, preset, and referenced since these are the only data oriented concepts languages provide. When a procedural statement is to be tested, it will be invoked in order to examine whether the procedure performs its stated functions.

There exist language concepts that are inexorably linked together; switch declarations and switch invocations; procedure declarations and procedure calls, etc., that individually serve little useful function but when utilized in combination provide a powerful programming tool. These notions will be validated fully.

Axioms

The validity of JOVIAL test features must be determined by the execution of a number of JOVIAL statements called support statements. Since these statements are themselves JOVIAL statements, they must be validated as is any other JOVIAL statement. Once a JOVIAL statement has been validated, however, the statement may be used to check the results of the validations of other JOVIAL statements.

Following is a list of these JOVIAL concepts that are required as basic axioms. The ability to:

1. Define and preset a hollerith item.
2. Assign a hollerith constant to a hollerith variable.
3. Execute the GOTO statement-name.
4. Define a procedure, invoke a procedure, and return from a procedure; input parameter list, one variable.
5. IF clause.

These axioms will be validated first.

Following the Axiom validation will be the validation of the data and procedures. The complete order for listing including all DDI-NO references and all CED-NO cross references is given in the Test Hierarchy Outline.

Test Modules

Although the concept of test modules has been described in section 2.3 of the Users Manual, that description will be repeated here.

A Test Module is a collection of JOVIAL statements that test a particular feature of the JOVIAL compiler. The feature may be a JOVIAL concept, a single JOVIAL statement or a collection of JOVIAL statements. Included in each Test Module are the:

1. Test identification field
2. Input test data fields
3. Test results fields
4. Expected results fields
5. Initialization procedures
6. Test statements comprising the test
7. Results analysis procedures
8. Output procedures

Test Modules are located on the Population File in order of their test serial number, the DDI-NO. With each test statement is associated a sequence number within the DDI-NO that specifies the ordering of the statements within the DDI-NO.

In most cases, a Test Module can be considered as an independent JOVIAL source program. There are instances, however, when the data to be operated upon by one Test Module resides in another Test Module. Consequently, in these cases, the JOVIAL source program is not independent. Exit from all modules passes through the last statement of the module to the first statement of the following module or the TERM statement. Because of this feature, a JOVIAL test module may follow any other JOVIAL test module.

Mandatory Modules

Some test modules are not independent in the sense that they may be included by themselves in a generated JOVIAL source program. These test modules depend upon other test modules called mandatory modules in the Population File for either of two reasons:

1. The mandatory test module contains data definitions that are required by the dependent test module, or
2. The mandatory test module contains support statements whose validity must be established before a successful execution of the dependent module feature may be considered valid.

The five support statement Axioms are considered to be constantly mandatory and consequently are included in every generated JOVIAL source program.

All other mandatory maodules will be invoked by specific test modules. Every mandatory module will be invoked by at least one test module and the relationship between test modules and mandatory modules, if any exist, will be enumerated in the Test Hierarchy Outline.

Test Module Content

Each Test Module will be identified by a test serial number called the DDI-NO occupying columns 73-76 of every card in the Test Module. Within each Test Module, individual cards will be given sequence numbers which will occupy columns 78-80.

Identification information describing various aspects of the test module is provided in the Test Header Card (card sequence number 001, see Users Manual Section 4.1.2.2.1). The Test Name in this card will be identical to the name used in the various section headings of the Test Hierarchy Outline. For example, test module 0500 will have the Test Name DEFINE-PRESET H ITEM, the identical name used to entitle Section 2.1 of the Test Hierarchy Outline.

Any CED-NO's to which a test module refers will be given in the appropriate positions on the Test Header Card. For example, test module 0500 refers to both CED-NO's 2463 and 2464. These numbers are included in their respective fields on the Test Header Card.

Any mandatory DDI-NO upon which the test module depends is included in columns 59-62 of this card.

The second card (card sequence number 002) in every test module contains the classification (section number) of the Test Hierarchy Outline. Columns 3-22 contain the words CLASSIFICATION NUMBER. Columns 26-33 contain the classification number in the following form XX.XX.XX.

The third card (card sequence number 003) in every test module contains the following statement from column 3-50:

THIS MODULE TESTS THE ABILITY OF THE COMPILER TO. . .

The fourth card and subsequent cards in the test module are used to expand further on the test description.

Following the last descriptive card in the test are the test and support statements themselves.

Test Module Output

The results of each test module are printed in a standard form. At least two printed lines are always output. The first line always consists of:

TEST MODULE XXXX

where XXXX is the DDI-NO of the module under test. The second line prints either of two messages:

TEST SUCCESSFUL (optional commentary)

or

TEST FAILED (optional commentary)

A blank line is automatically supplied by the JCVS separating consecutive test results.

JCVS Input/Output Characteristics

Since the implementation of the JOVIAL language is not closely monitored, deviations in implementation can and often do occur. Implementors take it upon themselves to change certain of the language specifications for any of many reasons. In particular, the implementation of the input/output specifications of the language have varied markedly in the past from implementor to implementor.

In addition, the language specifications do not permit the user to apply formatting to any results achieved by a JOVIAL program. Consequently, in order to format output information either a higher order language that permits formatting or an assembly language must be used.

It was originally intended to display actual versus expected results. Since the input/output capabilities of JOVIAL are ill-defined to non-existent, the initial plans for presentation of output was modified. Since FORTRAN offers excellent formatting capabilities, it was decided to use FORTRAN subroutines whenever formatting was required.

The notion of displaying expected versus actual results was abandoned for purposes of this project when it became apparent that converting internally computed numerical JOVIAL results from binary to decimal would be accomplished through FORTRAN conversion programs rather than JOVIAL conversion programs. Consequently, the tests would be invalid because certain processes would be carried out outside of JOVIAL language implementation. As a result of the above mentioned JOVIAL inadequacies, the following only qualitative output messages were printed. Test results printed out under these conditions do not fully reveal the causes of errors in tests devoted to the accuracy of arithmetic operations. The results of syntax-semantics testing, however, are not impaired by these constraints.

The JOVIAL input/output specifications described in AFM 100-24 do not adequately describe certain aspects of the file:declaration. In particular it is left to the implementor to specify the device:name. It is unclear precisely what constitutes a device:name and if the device:name remains inflexible for one computer configuration or precisely how it varies. In addition, the relationships that exist between the JOVIAL defined input/output statuses and the computer configuration software or hardware is not clear. It may be impossible to reconcile the input/output concepts provided by JOVIAL with the input/output concepts provided by the hardware or software environment.

Until a more firm relationship can be established, no testing of the file:declaration and, consequently, of the JOVIAL input/output statements will be provided at this time. These features are considered to be non-standard features.

FORTRAN I/O Usage

Test module 9998 uses the FORTRAN I/O format statement

```
PRT (date-name) $
```

For each computer configuration this statement must be provided in a form compatible to the hardware and software environment.

Population File Conversion

The Population File is keypunched using the IBM 026 character set. Some of the equipment utilized on this project use different character sets. In general, only the card punches for the so-called special characters vary from character set to character set. A complete list of these special characters together with their punched card representations is given in the accompanying Character Set Table.

It may be desireable to convert the Population File from one character set representation to another. The JCVS provides a FORTRAN routine called CONVER that performs this conversion. This routine varies slightly from configuration to configuration but performs the same task.

In general, this deck is submitted to the computer in the following form:

1. Leading Operating System Control Cards
2. CONVER Source Program Deck
3. Data Card 1

4. Data Card 2
5. Data to be Converted (Card Deck)
6. Final Operating System Control Cards

Data Card 1 contains the special characters in the data deck following that are to undergo translation. Data Card 2 contains the special characters to which the original special characters encountered in the Data Deck will be converted.

Each character of each card in the Data Deck is tested for possible conversion. If a conversion is to be made, the original special character is looked up in a table developed from the corresponding special characters in Data Card 1 and Data Card 2. If a match is accomplished, the new special character is substituted.

Every character in Data Deck is tested in this way. If a card does in fact contain one or more characters to be converted, the converted card as well as the original card, is printed. If a card contains no characters to be converted, only the original card is printed.

Data Card 1 and Data Card 2 have identical formats described as follows:

Columns	Description
1-2	Number of special characters.
3-80	Each column on Data Card 1 contains a character to be converted while the corresponding column on Data Card 2 contains the character to be convered to.

Following are the deck structures for the four computers used on the project:

1) UNIVAC 1108

7/8 RUN 1CONVER, DOCG, 5,300

7/8 1 FOR CONVER

(CONVER Source Deck)

7/8 XQT CONVER

(Data Card 1)
(Data Card 2)
(Data Deck)

7/8 FIN

2) IBM 360-50

```
//CONVER, JOB(799,028,010,1084,10,5), ANTCHAGNO, MSGLEVE=1
//S1 EXEC FORTGCLG
//FORT.SYSIN DD *
   (CONVER Source Deck)
//GO.SYSIN DD *
   (Data Card 1)
   (Data Card 2)
   (Data Deck)
/*
```

3) CDC-6400

```
JOB, 93007,10,10,35000. CONVER
RUN(S)
LGO.
(End of Record Card)
(CONVER Source Deck)
(End of Record Card)
(Data Card 1)
(Data Card 2)
(Data Deck)
(End of File Card)
```

4) GE-635

```
$       IDENT       3154203,DATDY
$       FORTRAN
$       INCODE      IBMF
        (CONVER Source Deck)
$       OPTION      FORTRAN
$       EXECUTE
$       LIMITS      15,32000
$       DATA        01
        (Data Card 1)
        (Data Card 2)
$       DATA        05
        (Data Deck)
$       ENDJOB
***EOF
```

CHARACTER SET TABLE

The following characters from the JOVIAL character set require conversion when translating from the character set of one computer to that of the other. Following is a chart showing the Hollerith representation used by each computer.

Character	Columns	GE-635	CDC-6400	UNIVAC-1108	IBM 360-50
+	3	12	12	12	12-8-6
=	4	0-5-8	3-8	3-8	6-8
$	5	11-3-8	11-3-8	11-3-8	11-3-8
;	6	11-6-8	12-7-8	11-6-8	11-6-8
,	7	0-6-8	4-8	4-8	5-8
(	8	12-5-8	0-4-8	0-4-8	12-5-8
)	9	11-5-8	12-4-8	12-4-8	11-5-8
>	10	6-8	11-7-8	6-8	0-6-8
<	11	12-6-8	12-0	12-6-8	12-4-8

TEST HIERARCHY OUTLINE

1. SYNTAX

Tests of the following syntax will be performed in the various test modules of the Population File. The accompanying tables provide a cross reference to some of the uses of the specified syntactic types as indicated by the associated DDI-NO's.

1.1 Primitives

	DDI-NO
ABS	5100
ALL	4390
AND	5810 5815 6110 6115 6135
ARRAY	3500 3505 3510 3515 3520 3525 3530 3535 3540 3545
ASSIGN	This feature deals with direct code and will not be tested.
BEGIN	1500 1505 1510 1550 1555
BIT	4000 4005 4010 4015
BYTE	4080 4085 4090 4095
CHAR	This feature is machine dependent and will not be tested.
CLOSE	This feature is an I/O concept and will not be tested.
DEFINE	6500
DIRECT	No test of machine language concepts will be provided.
END	1500 1505 1510 1550 1555
ENT	4145 4150 4155 4160 4165
ENTRY	4120 4125 4130 4135 4140
EQ	1500 1505 1510 1550 1555
FILE	This feature is an I/O concept and will not be tested.
FOR	5340 5342 5344 5346 5348
GOTO	1500 1505 1510 1550 1555
GQ	5820 5825
GR	5820 5825
IF	5820 5825
IFEITH	5310 1445 1450 1455
INPUT	This feature is an I/O concept and will not be tested

ITEM	1000 1005 1010 1015 1020 1025 1030 1035
JOVIAL	No test of machine language concepts will be provided.
'LOC	4175 4178 4181 4187 4190
LQ	5820 5825
LS	5820 5825
MANT	This feature is machine dependent and will not be tested.
MODE	This feature is not tested.
NENT	4200 4205 4210 4215 4220 4225
NOT	6125 6130 6135
NQ	5820 5825
NWDSEN	4300 4305 4310
ODD	4340 4345 4350
OPEN	This feature is an I/O concept and will not be tested.
OR	5820 6100 6105 6120 6135
ORIF	5310 1445 1450 1455
OUTPUT	This feature is an I/O concept and will not be tested.
OVERLAY	6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750
POS	This feature is an I/O concept and will not be tested.
PROC	4465 4470 4475 4480 4485
'PROGRAM	4175 4196 4199
RETURN	5180 5185 5190
SHUT	This feature is an I/O concept and will not be tested.
START	0001
STOP	This feature is not tested.
STRING	3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650
SWITCH	3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965
TABLE	2500 2505 2510 2515 2520 2525 2530 2535 2540
TERM	9999
TEST	5390 5392 5394

1.2 Ideograms

+	5425 5430 5435 5440
-	5455 5460 5465

*	5425 5430 5435 5440
/	5440 5445 5450 5455
.	All test modules.
,	5510 5515 5505
=	6200
(	4465 4470 4475 4480
)	4465 4470 4475 4480
$	All test modules.
**	5455 5460 5475 5480
= =	6400 6405 6410 6415 6420 6425
"	All test modules.
($	4480 4490 4825
$)	4480 4490 4825
...	1000 1010 1015
(/	5500
/)	5500
(*	
*)	

1.3 Single Letter Symbols

1.3.1 Abbreviation 4465 4470 4475 4480

1.3.2 Loop Variable 5350 5355 5360

1.4 Names

1.4.1 Labels All test modules.

1.4.2 Identifires All test modules.

2. AXIOMS

Prove the axioms contained in the following required mandatory modules.

		DDI-NO
2.1	DEFINE-PRESET H ITEM	0500
2.2	ASSIGNMENT STAT H	0505
2.3	GOTO STAT-NAME	0510
2.4	PROCEDURE, ERR, PRINT	0515
2.5	IF CLAUSE	0520

3. DEFINE, PRESET AND REFERENCE USED DEFINED DATA

3.1 Simple Items

3.1.1 Define Simple Items

Classification Number		DDI-NO
3.1.1.1	Integer Item	1000
3.1.1.2	Floating Item	1005
3.1.1.3	Fixed Item	1010
3.1.1.4	Dual Item	1015
3.1.1.5	Hollerith Items	1020
3.1.1.6	Transmission Code Items	1025
3.1.1.7	Status Items	1030
3.1.1.8	Boolean Items	1035

3.1.2 Form Constants and Preset Simple Items

Classification Number		DDI-NO
3.1.2.1	Integer Item	1200
3.1.2.2	Floating Item	1205
3.1.2.3	Fixed Item	1210
3.1.2.4	Dual Items	1215
3.1.2.5	Hollerith Items	1220
3.1.2.6	Transmission Code Items	1225
3.1.2.7	Status Item	1230
3.1.2.8	Boolean Items	1235

3.1.3 Reference Defined and Preset Items

Classification Number		DDI-NO
3.1.3.1	Integer Item	1400
3.1.3.2	Floating Item	1405
3.1.3.3	Fixed Item	1410
3.1.3.4	Dual Item	1415
3.1.3.5	Hollerith Item	1420
3.1.3.6	Transmission Code Item	1425
3.1.3.7	Status Items	1430
3.1.3.8	Boolean Items	1435

3.2 Ordinary Tables

3.2.1 Ordinary Table Definition

Define ordinary tables of the following types.

Classification Number	Classification Name							DDI-NO	
3.2.1.1	TABLE	Name	V	int-1	S	N	$	1500	
3.2.1.2	TABLE	Name	V	int-1	S	M	$	1505	
3.2.1.3	TABLE	Name	V	int-1	S	D	$	1510	
3.2.1.4	TABLE	Name	V	int-1	-	-	$	1515	1605
3.2.1.5	TABLE	Name	R	int-1	S	D	$	1520	
3.2.1.6	TABLE	Name	R	int-1	S	M	$	1525	
3.2.1.7	TABLE	Name	R	int-1	S	N	$	1530	
3.2.1.8	TABLE	Name	R	int-1	S	-	$	1535	
3.2.1.9	TABLE	Name	V	int-1	P	M	$	1540	
3.2.1.10	TABLE	Name	V	int-1	P	D	$	1545	1630
3.2.1.11	TABLE	Name	V	int-1	P	N	$	1550	
3.2.1.12	TABLE	Name	V	int-1	P	M	$	1555	1625
3.2.1.13	TABLE	Name	R	int-1	P	N	$	1560	
3.2.1.14	TABLE	Name	R	int-1	P	D	$	1565	
3.2.1.15	TABLE	Name	R	int-1	P	M	$	1570	
3.2.1.16	TABLE	Name	R	int-1	-	N	$	1575	
3.2.1.17	TABLE	Name	R	int-1	-	M	$	1580	
3.2.1.18	TABLE	Name	R	int-1	-	D	$	1585	
3.2.1.19	TABLE	Name	V	int-1	-	N	$	1590	
3.2.1.20	TABLE	Name	V	int-1	-	M	$	1595	
3.2.1.21	TABLE	Name	V	int-1	-	D	$	1600	
3.2.1.22	TABLE	Name	R	int-1	-	-	$	1610	
3.2.1.23	TABLE	Name	V	int-1	S	-	$	1615	
3.2.1.24	TABLE	Name	V	int-1	P	-	$	1620	
3.2.1.25	TABLE	Name	R	int-1	-	-	$	1635	

3.2.2 Ordinary Table Items

Define, Preset and Reference all item types within the following table types.

Classification Number		DDI-NO		
3.2.2.1	TABLE Name V int-1 S $			
	Hollerith Item	1700		
	Transmission Code Item	1705		
	Integer Item	1710		
	Fixed Point Item	1715		
	Floating Point Item	1720		
	Status Item	1725		
	Boolean Item	1730		
	Dual Item	1735		
3.2.2.2	TABLE Name V int-1 P $			
	Hollerith Item	1740		
	Transmission Code Item	1745		
	Integer Item	1750		
	Fixed Point Item	1755		
	Floating Point Item	1760		
	Status Item	1765		
	Boolean Item	1770		
	Dual Item	1775		
3.2.2.3	TABLE Name R int-1 P $			
	Hollerith Item	1780		
	Transmission Code Item	1785		
	Integer Item	1790		
	Fixed Point Item	1795		
	Floating Point Item	1800		
	Status Item	1805		
	Boolean Item	1810		
	Dual Item	1815		
3.2.2.4	TABLE Name R int-1 S $			
	Hollerith Item	1820	1900	
	Transmission Code Item	1825	1905	
	Integer Item	1830	1910	1915

Classification Number		DDI-NO					
	Fixed Point Item	1835	1840	1920	1925		
	Floating Point Item	1845	1930				
	Status Item	1850	1935				
	Boolean Item	1855	1940				
	Dual Item	1860	1945				
	Summary of some modules using Ordinary Table Items.						
	Hollerith Item	1505	1550	1525	1570	1590	1575
	Transmission Code Item	1505	1550	1525	1570	1595	1580
	Integer Item	1500	1545	1530	1560	1605	1610
		1615	1620	1625	1635		
	Fixed Point Item	1500	1545	1530	1560	1515	
	Floating Point Item	1500	1545	1530	1560	1595	1515
		1580					
	Status Item	1510	1555	1520	1565	1600	1585
		1630					
	Boolean Item	1510	1555	1520	1565	1590	1575
	Dual Item	1510	1555	1520	1565	1600	1585

3.3 Defined Tables

3.3.1 Defined Entry Table Definition

Define Defined Entry Tables of the following types. Use all table descriptors at least once.

Classification Number									DDI-NO	
3.3.1.1	TABLE	Name	V	int-1	-	-	int-2	$	2500	
3.3.1.2	TABLE	Name	V	int-1	P	-	int-2	$	2505	2535
3.3.1.3	TABLE	Name	V	int-1	S	-	int-2	$	2510	2540

Classification Number									DDI-NO
3.3.1.4	TABLE	Name	R	int-1	-	-	int-2	$	2515
3.3.1.5	TABLE	Name	R	int-1	S	-	int-2	$	2520
3.3.1.6	TABLE	Name	R	int-1	P	-	int-2	$	2525
3.3.1.7	TABLE	Name	R	int-1	-	-	int-2	$	2530

3.3.2 Defined Entry Table Items

Define Preset and Reference all item types within the following Defined Entry tables. Use all possible item types at least once.

Classification Number	Classification Name							DDI-NO		
3.3.2.1	TABLE	Name	V	int-1	-	-	int-2	$		
	Integer Item								2500	2535 2540
3.3.2.2	TABLE	Name	V	int-1	P	-	int-2	$		
	Fixed Items								2505	
3.3.2.3	TABLE	Name	V	int-1	S	-	int-2	$		
	Hollerith Items								2510	
3.3.2.4	TABLE	Name	R	int-1	-	-	int-2	$		
	Transmission Item								2515	
3.3.2.5	TABLE	Name	R	int-1	S	-	int-2	$		
	Boolean Item								2520	
	Status Item								2520	
	Integer Item								2520	
3.3.2.6	TABLE	Name	R	int-1	P	-	int-2	$		
	Floating Item								2525	
3.3.2.7	TABLE	Name	R	int-1	-	-	int-2	$		
	Dual Item								2530	

Summary of some modules using Defined Entry Table Items.

Classification Number		DDI-NO
	Entry Table Items	
	Hollerith Item	2510
	Transmission Code Item	2515
	Integer Item	2500 2520 2535 2540
	Fixed Point Item	2505
	Floating Point Item	2525
	Status Item	2520
	Boolean Item	2520
	Dual Item	2530

3.3.3 Defined Entry Table Strings

Define all types of Defined Entry Table Strings within the following Defined Entry Table types.

Classification Number		DDI-NO
3.3.3.1	TABLE Name V int-1 - M int-2 $	
3.3.3.1.1	STRING Name floating item int-3 int-4 N int-5 int-6 $	3600
3.3.3.1.2	STRING Name integer item int-3 int-4 N int-5 int-6 $	3605
3.3.3.2	TABLE Name V int-1 S N int-2 $	
3.3.3.2.1	STRING Name hollerith item int-3 int-4 N int-5 int-6 $	3610
3.3.3.2.2	STRING Name status item int-3 int-4 N int-5 int-6 $	3615
3.3.3.3	TABLE Name V int-1 P D int-2 $	
3.3.3.3.1	STRING Name transmission code item int-3 int-4 D int-5 int-6 $	3620
3.3.3.3.2	STRING Name boolean item int-3 int-4 D int-5 int-6 $	3625

Classification Number		DDI-NO		
3.3.3.4	TABLE Name R int-1 - - int-2 $			
3.3.3.4.1	STRING Name fixed item int-3 int-4 - int-5 int-6 $	3630		
3.3.3.4.2	STRING Name dual item int-3 int-4 - int-5 int-6 $	3635		
3.3.3.5	TABLE Name R int-1 S M int-2 $			
3.3.3.5.1	STRING Name transmission code item int-3 int-4 M int-5 int-6 $	3640		
3.3.3.5.2	STRING Name boolean name int-3 int-4 M int-5 int-6 $	3645		
3.3.3.6	TABLE Name R int-1 P - int-2 $			
3.3.3.6.1	STRING Name integer item int-3 int-4 - int-5 int-6 $	3650		
3.3.3.6.2	STRING Name hollerith item int-3 int-4 - int-5 int-6 $	3655		
3.3.3.7	TABLE Name V int-1 P - int-2 $			
3.3.3.7.1	STRING Name integer item int-3 int-4 D int-5 int-6 $	3660		
3.3.3.7.2	STRING Name integer item int-3 int-4 D int-5 int-6 $	3660		
	Summary of item types used in String definitions.			
	Hollerith Items	3610	3655	
	Transmission Code Item	3620	3640	
	Integer Item	3605	3650	3660
	Fixed Point Item	3630		
	Floating Point Item	3600		
	Status Item	3615		
	Boolean Item	3625	3645	
	Dual Item	3635		

3.4 Arrays

3.4.1 Array Definitions

Define Arrays of the following types.

Classification Number	Classification	DDI-NO
3.4.1.1	ARRAY int-1 item description $	3535 3545
3.4.1.2	ARRAY int-1 int-2 item description $	3500 3505 3515 3525 3530 3545
3.4.1.3	ARRAY int-1 int-2 int-3 item description $	3510 3520 3540 3545

3.4.2 Array Items

Define Preset and reference all item types within the following Array types.

Classification Number	Classification	DDI-NO
3.4.2.1	ARRAY int-1 integer item $	3545
3.4.2.2	ARRAY int-1 int-2 integer item $	3500 3545
3.4.2.3	ARRAY int-1 int-2 int-3 integer item $	3545
3.4.2.4	ARRAY int-1 int-2 floating item $	3505
3.4.2.5	ARRAY int-1 int-2 int-3 status item $	3510
3.4.2.6	ARRAY int-1 int-2 hollerith item $	3515
3.4.2.7	ARRAY int-1 hollerith item $	3520
3.4.2.8	ARRAY int-1 int-2 int-3 boolean item $	3525
3.4.2.9	ARRAY int-1 int-2 dual item $	3530
3.4.2.10	ARRAY int-1 int-2 fixed item $	3535
3.4.2.11	ARRAY int-1 int-2 int-3 transmission code item	3540

3.5 Switches

Validate switch usage by defining a switch and then referencing it.

Classification Number		DDI-NO		
3.5.1	ITEM SWITCH Validate the usage of item switches using various item types and sequence designators.			
3.5.1.1	Integer Item Switch, Statement-Name	3900	3910	3940
3.5.1.2	Fixed Item Switch, Statement-Name	3905		
3.5.1.3	Hollerith Item Switch, Statement-Name	3910		
3.5.1.4	Floating Point Item Switch, Statement-Name	3915		
3.5.1.5	Transmission Item Switch, Statement-Name	3920		
3.5.1.6	Dual Item Switch, Statement-Name	3925		
3.5.1.7	Status Item Switch, Statement-Name	3930		
3.5.1.8	Boolean Item Switch, Statement-Name	3935		
3.5.2	INDEX SWITCH			
3.5.2.1	Index Switch, Statement-Name	3940		
3.5.2.2	Index Switch, Statement-Name	3945		
3.5.2.3	Index Switch, Index Switch	3950		
3.5.2.4	Index Switch, Statement-Name	3955		
3.5.2.5	Index Switch, Statement-Name	3960		
3.5.2.6	Index Switch, Close-Name	3965		

4. SPECIAL DATA REFERENCING

4.1 BIT

Validate BIT referencing for the following variable types and one or two component indices.

Classification Number		DDI-NO
4.1.1	BIT ($ two component index $)(integer item)	4000
4.1.2	BIT ($ one component index $)(fixed point item)	4005
4.1.3	BIT ($ two component index $)(status item)	4010
4.1.4	BIT (# one component index $)(boolean item)	4015

4.2 BYTE

Validate BYTE referencing for the following variable types and one or two component indices.

Classification Number		DDI-NO
4.2.1	BYTE ($ two component index $)(transmission item)	4080
4.2.2	BYTE ($ two component index $)(table hollerith item)	4085
4.2.3	BYTE ($ one component index $)(hollerith item)	4090
4.2.4	BYTE ($ two component index $)(table transmission item)	4095

4.3 CHAR

This feature is machine dependent and will not be tested here.

4.4 ENTRY and ENT

Validate ENTRY referencing within the indicated table type and forms of the functional modifier.

Classification Number		DDI-NO
4.4.1	ENTRY	
4.4.1.1	Within an IF statement	4120

Classification Number		DDI-NO
4.4.1.2	Within an IF statement	4122
4.4.1.3	Within an Assignment statement	4124
4.4.1.4	Within an Exchange statement	4126
4.4.1.5	Within IFEITH and ORIF statements	4128

Use the following defined table type in subsequent tests.

TABLE Name R int-1 S - $

Use the following forms of the ENTRY and ENT modifiers.

1 ENTRY(table-name($ one component index $))
2 ENTRY(table-item-name($ one component index $))
3 ENT(table-name($ one component index $))
4 ENT(table-item-name($ one component index $))

4.4.2 ENT

Classification Number		DDI-NO
4.4.2.1	Within an IF statement	4130
4.4.2.2	Within an IF statement	4132
4.4.2.3	Within an Assignment statement	4134
4.4.2.4	Within an Exchange statement	4136
4.4.2.5	Within IFEITH and ORIF statements	4138

Use the following table types in subsequent tests.

4.4.3 ENTRY

Classification Number									DDI-NO
4.4.3.1	TABLE	Name	V	int-1	S	-	$		4140
4.4.3.2	TABLE	Name	V	int-1	P	-	$		4142
4.4.3.3	TABLE	Name	V	int-1	-	-	$		4144
4.4.3.4	TABLE	Name	R	int-1	S	-	$		4146
4.4.3.5	TABLE	Name	R	int-1	P	-	$		4148
4.4.3.6	TABLE	Name	R	int-1	-	-	$		4150

4.4.4 ENT

Classification Number									DDI-NO
4.4.4.1	TABLE	Name	V	int-1	S	-	$		4152
4.4.4.2	TABLE	Name	V	int-1	P	-	$		4154
4.4.4.3	TABLE	Name	V	int-1	-	-	$		4156
4.4.4.4	TABLE	Name	R	int-1	S	-	$		4158
4.4.4.5	TABLE	Name	R	int-1	P	-	$		4160
4.4.4.6	TABLE	Name	R	int-1	-	-	$		4162

4.5 PRIME LOC

Validate referencing with 'LOC.

Classification Number		DDI-NO		
4.5.1	'LOC (program-name)	4175	4196	4199
4.5.2	'LOC (simple item-name)	4178		
4.5.3	'LOC (table-name)	4181	4193	
4.5.4	'LOC (table-item-name)	4181	4193	

Classification Number		DDI-NO
4.5.5	'LOC (array-item-name)	4184
4.5.6	'LOC (string item-name)	4187
4.5.7	'LOC (statement-name)	4190 4193

4.6 MANT

This feature is machine dependent and will not be tested here.

4.7 NENT

Validate NENT referencing for the following table types.

Classification Number		DDI-NO
4.7.1	NENT (Ordinary table-name)	
4.7.1.1	TABLE name R int-1 S - $	4200
4.7.1.2	TABLE name R int-1 P - $	4205
4.7.1.3	TABLE name V int-1 S - $	4210
4.7.1.4	TABLE name V int-1 P - $	4215
4.7.1.5	TABLE name V int-1 - - $	4220
4.7.1.6	TABLE name V int-1 - - $	4225
4.7.2	NENT (Defined Entry table-name)	
4.7.2.1	TABLE name V int-1 S int-2 $	4230
4.7.2.2	TABLE name V int-1 P int-2 $	4235

4.8 NWDSEN

Validate NWDSEN referencing for the following table types.

Classification Number		DDI-NO
4.8.1	TABLE name V int-1 S - $	4300
4.8.2	TABLE name V int-1 P - $	4305
4.8.3	TABLE name R int-1 S - $	4310
4.8.4	TABLE name V int-1 - - $	4315

4.9 ODD

Validate referencing with ODD.

Classification Number		DDI-NO
4.9.1	ODD (integer variable)	4340
4.9.2	ODD (fixed variable)	4345
4.9.3	ODD (loop variable)	4350

4.10 POS

This is an I/O concept and will not be tested.

4.11 ALL

Validate the use of ALL in a FOR loop statement.

Classification Number		DDI-NO
4.11.1	ALL	4390

5. PROGRAM ORGANIZATION

5.1 PRIME PROGRAM

Test the ability of the compiler to origin programs correctly.

Classification Number		DDI-NO	
5.1.1	'PROGRAM name octal constant	4196	4175
5.1.2	'PROGRAM name decimal constant	4199	4175

5.2 Procedures

5.2.1 User Defined Procedures

Check the usage of user defined procedures.

Classification Number		DDI-NO	
5.2.1.1	PROC name $	4450	4465
5.2.1.2	PROC name (input-parameter list) $		
	Input Parameter List - Variable Reference	4470	
5.2.1.3	PROC name (= output parameter list) $		
	Output Parameter List - Table Reference	4475	
5.2.1.4	PROC name (input parameter list = output parameter list) $		
	Input Parameter List - Variable Reference	4455	4485
	Input Parameter List - Array References	4490	
	Input Parameter List - Table References	4480	
	Output Parameter List - Formula References	4495	
	Output Parameter List - Variable References	4460	4485

Classification Number		DDI-NO
	Output Parameter List - Array Reference	4480
	Output Parameter List - Table Reference	4490

5.2.2 Language Defined Procedure

Check the usage of the language defined procedure, REMQUO.

Classification Number		DDI-NO
5.2.2.1	REMQUO (integer item-1 , integer item-2)	4770
5.2.2.2	REMQUO (integer item-1 , integer item-2)	4775
5.2.2.3	REMQUO (integer item-1 , integer item-2)	4780
5.2.2.4	REMQUO (integer constant-1, integer-constant-2)	4785

5.3 Functions

5.3.1 User Defined Functions - PROC

Check the usage of user defined functions.

Classification Number		DDI-NO
5.3.1.1	PROC name $	
	Function Type	
	Boolean	4805
	Integer	4810
	Hollerith	4820
	Dual	4845

Classification Number		DDI-NO			
5.3.1.2	PROC name (input parameter list) $				
	Function Type				
	Boolean	4800	4830	4840	
	Integer	4810	4815		
	Transmission	4825			
	Floating	4830			
	Fixed	4835			
	Input Parameter List				
	Boolean Formula	4800			
	Integer	4810	4830	4835	4840
	Table	4825			
	Array	4825			
	Floating	4830			

5.3.2 Language Defined Functions, ABS and REM

Check the usage of the language defined functions ABS and REM.

Classification Number		DDI-NO
5.3.2.1	ABS (integer item)	5100
5.3.2.2	integer = ABS (integer item)	5100
5.3.2.3	REM (integer item-1 , integer item-2)	5105
5.3.2.3.1	REM (integer item-1 , integer item-2)	5110
5.3.2.3.2	REM (integer item-1 , integer item-2)	5115
5.3.2.3.3	REM (integer constant-1, integer constant-2)	5120

5.3.3 User Defined Function - CLOSE

Check the usage of the user defined function, CLOSE.

Classification Number		DDI-NO
5.3.3.1	CLOSE	5160

5.4 RETURN

Check the usage of the RETURN feature.

Classification Number		DDI-NO
5.4.1	RETURN from a procedure	5180
5.4.2	RETURN from a used defined function	5185
5.4.3	RETURN from a close	5190

6. PROGRAM CONTROL

6.1 GOTO

Check the usage of the GOTO statement. Since most modules use the GOTO statement, no special modules will be devoted to testing this feature. Instead, references to modules using this feature will be given.

Classification Number		DDI-NO
6.1.1	GOTO statement-name $	4162
6.1.2	GOTO close-name $	5160
6.1.3	GOTO item-switch name $	3900
6.1.4	GOTO index-switch name $	3945

6.2 STOP

Because of a possible conflict with the operating system, this feature will not be tested.

6.3 IF Clause

Check the usage of the IF clause.

Classification Number		DDI-NO
6.3.1	IF statement followed by simple statement. This version of the IF is used in various modules. No special module will be devoted to its testing.	
6.3.2	IF statement followed by compound statement	5280

6.4 IFEITH, ORIF

Check the usage of the IFEITH and ORIF for various item types.

Classification Number		DDI-NO
6.4.1	Boolean Items	5310
6.4.2	Integer Items	1445
6.4.3	Floating Items	1450
6.4.4	Status Items	1455
6.4.5	Transmission Items	1460
6.4.6	Mixed Data Items	5805

6.5 FOR Loops

Check the usage of the various forms of the FOR loop.

Classification Number		DDI-NO
6.5.1	Single FOR loops - Constant Factors	
6.5.1.1	One Factor FOR - Incrementing	5340
6.5.1.2	Two Factor FOR - Incrementing	5342
6.5.1.3	Three Factor FOR - Incrementing	5344
6.5.1.4	Three Factor FOR - Incrementing	5346
6.5.1.5	One Factor FOR - Decrementing	5348
6.5.1.6	Two Factor FOR - Decrementing	5350
6.5.1.7	Three Factor FOR - Decrementing	5352
6.5.1.8	Three Factor FOR - Decrementing	5354
6.5.2	Nested or Multiple FOR loops - Constant Factors	
6.5.2.1	One Factor FOR within Three Factor FOR	5356
6.5.2.2	Two Factor FOR within Three Factor FOR	5358
6.5.2.3	Two Factor FOR and Three Factor FOR	5360
6.5.2.4	Two Factor FOR and Three Factor FOR	5362
6.5.2.5	Three Factor FOR with Three Factor FOR	5364
6.5.3	FOR loops - Non Constant Factors	
6.5.3.1	Three Factor FOR - Variable First Factor	5366
6.5.3.2	Three Factor FOR - Variable Second Factor	5368
6.5.3.3	Three Factor FOR - Formula First Factor	5370
6.5.3.4	Three Factor FOR - Formula Second Factor	5372

6.6 Loop Control

Check the operation of the TEST statement under the following FOR loop conditions.

Classification Number		DDI-NO
6.6.1	Three Factor FOR	5390
6.6.2	Two Factor FOR	5392
6.6.3	Loop Variable within Loop Variable	5394
6.6.4	Loop Variable within Loop Variable	5396

7. PROCEDURE FORMATION

7.1 Numeric Expression

Classification Number		DDI-NO
7.1.1	Arithmetic Operations	
7.1.1.1	Integer Variables	5400
7.1.1.2	Fixed Point Variables	5405
7.1.1.3	Floating Point Variables	5410
7.1.2	Unary Operators	
7.1.3	Multiple Operator Expressions	
7.1.3.1	Integer Variables (+,*)	5425
7.1.3.2	Fixed Point Variables (+,*)	5430
7.1.3.3	Floating Point Variables (+,*)	5435
7.1.3.4	Integer Variables (*,/,+)	5440
7.1.3.5	Fixed Point Variables (*,/,+)	5445
7.1.3.6	Floating Point Variables (*,/,+)	5450
7.1.3.7	Integer Variables (+,-,*,/,**)	5455
7.1.3.8	Fixed Point Variables (+,-,*,/,**)	5460
7.1.3.9	Floating Point Variables (+,-,*,/,**)	5465
7.1.4	Precedence of Operations	
7.1.4.1	Bracketed Addition	5470
7.1.4.2	Integer Negation	5475
7.1.4.3	Negative Number Exponentation	5480
7.1.5	Dual Operations	
7.1.5.1	Integer Variables (+,-,*,/)	5485
7.1.5.2	Integer Variables (+,*,/)	5490
7.1.5.3	Fixed Point Variables (+,*,/)	5495
7.1.5.4	Floating Point Variables (+,*,/)	5500
7.1.5.5	Integer Variables (-,*,/,**)	5505
7.1.5.6	Fixed Point Variables (+,-,*,/,**)	5510
7.1.5.7	Floating Point Variables (+,-,*,/,**)	5515

Classification Number		DDI-NO
7.1.6	Precedence in Dual Expressions	
7.1.6.1	Bracketed Division	5520
7.1.6.2	Bracketed Division	5525
7.1.6.3	Bracketed Division	5530
7.1.6.4	Precedence of Multiplication	5535
7.1.6.5	Precedence of Negated Constant	5540
7.1.6.6	Precedence of Negated Constant	5545

7.2 Simple Comparisons

Check the use of comparisons under the following conditions.

Classification Number		DDI-NO
7.2.1	Relational Operators - Integer Variables	5800
7.2.2	Relational Operators - Fixed Variables	5805
7.2.3	Relational Operators - Floating Variables	5810
7.2.4	Relational Operators - Dual Variables	5815
7.2.5	Relational Operators - Integral Variables and Octal Constants	5820
7.2.6	Relational Operators - ENT and ENTRY	5825
7.2.7	Equality and Inequality - STATUS Variables	5830
7.2.8	Equality and Inequality - Hollerith Variables	5835
7.2.9	Equality and Inequality - Transmission Variables	5840

7.3 Chained Comparisons

Check the use of chained comparisons.

Classification Number		DDI-NO
7.3.1	Integer Variables	6000
7.3.2	Fixed Variables	6005
7.3.3	Floating Variables	6010
7.3.4	Dual Variables	6015
7.3.5	Hollerith Variables	6020
7.3.6	Transmission Variables	6025
7.3.7	Octal Constants	6030

7.4 Boolean Expressions

Check the usage of NOT, AND and OR in the development of Boolean expressions.

Classification Number		DDI-NO				
7.4.1	NOT	6125	6130	6135		
7.4.2	AND	5810	5815	6110	6115	6135
7.4.3	OR	5820	6100	6105	6120	6135

8. REPLACEMENT STATEMENTS

8.1 Assignment

Module 6200 contains the following Assignment statement variations:

1. Numeric Assignment
2. Dual
3. Literal
4. Boolean
5. Status
6. Entry

8.2 EXCHANGE

Check the use of the EXCHANGE statement in the following situations.

Classification Number		DDI-NO
8.2.1	Integer Variable = = Integer Variable	6400
8.2.2	Fixed Variable = = Fixed Variable	6405
8.2.3	Hollerith Variable = = Hollerith Variable	6410
8.2.4	Dual Variable = = Dual Variable	6415
8.2.5	Transmission Variable = = Transmission Variable	6420
8.2.6	Boolean Variable = = Boolean Variable	6425
8.2.7	Status Variable = = Status Variable	6430
8.2.8	Table Integer Variable = = Table Integer Variable	6435
8.2.9	Table Fixed Variable = = Table Fixed Variable	6440
8.2.10	Table Hollerith Variable = = Table Hollerith Variable	6445
8.2.11	Table Dual Variable = = Table Dual Variable	6450
8.2.12	Table Transmission Variable = = Table Transmission Variable	6455
8.2.13	Table Boolean Variable = = Table Boolean Variable	6460
8.2.14	Table Status Variable = = Table Status Variable	6465
8.2.15	Table ENTRY Variable = = Table ENTRY Variable	6470
8.2.16	Table ENT Variable = = Table ENT Variable	6475

9. COMPILER DIRECTING CONCEPTS

9.1 DEFINE

Check the usage of the DEFINE compiler directive. DDI-No. - 6500, Module - 6400

9.2 LIKE

Test the ability of compiler to define LIKE tables.

Classification Number		DDI-NO
9.2.1	LIKE - Ordinary Tables	6600 6610
9.2.2	LIKE - Defined Tables	6605

9.3 OVERLAY

Define and check that OVERLAY's perform as stated in AFM 100-24.

Classification Number		DDI-NO
9.3.1	OVERLAY IDS-1 = IDS-2 = , ... IDS-n $	
9.3.1.1	OVERLAY item-name-1 = item-name-2 $	6700
9.3.1.2	OVERLAY item-name = table-name $	6705
9.3.1.3	OVERLAY item-name = array-name $	6710
9.3.1.4	OVERLAY table-name = item-name-1, item-name-2 $	6715
9.3.1.5	OVERLAY table-name-1 = table-name-2 $	6720
9.3.1.6	OVERLAY table-name-1 = table-name-2 $	6725
9.3.1.7	OVERLAY array-name-1 = array-name-2 $	6730
9.3.1.8	OVERLAY array-name = table-name $	6735
9.3.1.9	OVERLAY table-name = array-name-1, array-name-2 $	6740
9.3.1.10	OVERLAY table-name-1 = table-name-2, table-name-3 $	6745
9.3.2	OVERLAY octal constant = IDS-1 = IDS-2 = ... IDS-n $	
9.3.2.1	OVERLAY ocyal constant = table-name = item-name	6750

Classification Number		DDI-NO
9.3.2.2	OVERLAY octal constant = item-name-1 = item-name-2 $	6755
9.3.2.3	OVERLAY octal constant = array-name = item-name $	6760
9.3.2.4	OVERLAY octal constant = table-name-1 = table-name-2 $	6765
9.3.2.5	OVERLAY octal constant = array-name-1 = array-name-2 $	6770
9.3.2.6	OVERLAY octal constant = table-name = array-name $	6775
9.3.3	OVERLAY number = IDS-1 = IDS-2 = ... IDS-n $	
9.3.3.1	OVERLAY number = table-name = item-name $	6780
9.3.3.2	OVERLAY number = item-name-1 = item-name-2 $	6785
9.3.3.3	OVERLAY number = array-name = item-name $	6790
9.3.3.4	OVERLAY number = table-name-1 = table-name -2 $	6795
9.3.3.5	OVERLAY number = array-name-1 = array-name -2 $	6800
9.3.3.6	OVERLAY number = array-name = table-name $	6805

9.4 MODE

This feature instructs the compiler to retain a data item according to a specified set of item descriptors. One of the assumptions made in the development of the Population File is that no check would be made of the form in which an item is stored by the system. A check of the MODE feature would require such a check. Consequently, this feature will not be tested.

10. INPUT/OUTPUT CONCEPTS

Because of the non-standard character of these features, no Input/Output tests will be performed.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
JOVIAL J-3 (J3) compiler validation						

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Directorate of Systems Design & Development
Hq Electronic Systems Division
L G Hanscom Field, Bedford, Mass. 01730

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
N/A

3. REPORT TITLE
USER'S MANUAL JOVIAL COMPILER VALIDATION SYSTEM

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
None

5. AUTHOR(S) *(First name, middle initial, last name)*
None

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1970	234	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
IN-HOUSE b. PROJECT NO. 6917	ESD-TR-70-278
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Hq Electronic Systems Division (AFSC) L. G. Hanscom Field, Bedford, Mass. 01730

13. ABSTRACT

This technical report consists of detailed specifications for the use of the JOVIAL Compiler Validation System (JCVS). The system is designed to measure the compliance of a specific JOVIAL J3 compiler against the language specifications in Air Force Manual 100-24, "Standard Computer Programming Language for Air Force Command and Control Systems". This report describes the card input formats, deck structures, tape requirements, test modules, and operator procedures required to use the system.

DD FORM 1 NOV 65 1473

Security Classification