

SESD014

CDC SOFTWARE ENGINEERING SERVICES

1

07/25/80

SES Release Process

Software Engineering Services
RELEASE PROCESS

written by E F Tushar
c January, 1979

revisions made:
c April, 1979
c June, 1979
c June, 1980

Design consultation by

R D Palm
c January, 1979

J J Krautbauer
c January, 1979
c June, 1979
c June, 1980

COMPANY PRIVATE

SES Release Process

1.0 INTRODUCTION

1.0 INTRODUCTION

In contrast to corporate implications of "releases", an SES Release is a process by which SES Tools are made "available" for company internal use with support. In the presentation that follows, any reference to "release" should be interpreted in this manner. Definitions of other terms used throughout this document are tabulated in Appendix A.

The objectives of the SES Release Process are:

- > To provide a timely availability of tools with a built in mechanism that allows for rapid repair of deficiencies.
- > To insure that uniform and dependable procedures are used to facilitate tool releases.
- > To tabulate the Tool Catalog file history in order to identify and recall any release version of these tools and to insure the integrity of the tools made available subject to the COMSOURCE Archive Policies.

The intent of this Release Procedures Document is to incorporate the purposes of the Release (stated above) into an organized program that insures consistency in the building and releasing of SES Tools.

This document is divided into four main parts. The first part deals with the general release process as a whole. The second and third parts discuss the role of the Release Manager and the Tool Submitter, respectively. The last part addresses various aspects of the release: schedules, catalog management, and terminology.

07/25/80

SES Release Process
.....2.0 THE RELEASE PROCESSES IN GENERAL
.....2.0 THE RELEASE PROCESSES IN GENERAL2.1 BASIC RESPONSIBILITIES

A successful release of a group of tools is dependent upon a systematic and organized execution of the release process according to established guidelines. The release process guidelines have been defined to facilitate the tasks of the Release Manager in co-ordinating the release. A conscious effort should be made by all persons involved with the release to adhere to these guidelines.

The fundamental responsibilities of the Release Manager are to co-ordinate the Scheduled Release process and manage the Tool Catalogs. In co-ordinating the release process, the Release Manager will attempt to synchronize within the pre-defined schedule the availability of materials, the execution of the verification build, and the final release of the new/changed tools involved. The section titled "The Scheduled Release Timetable" provides a breakdown of these release process tasks. The management of the Tool Catalogs is discussed in the section named "Tool Catalog Management".

The main role of the Submitter is to construct materials (i.e. tool PLs, build procs, and documents) for a tool's release according to the guidelines that this document attempts to establish and to submit these materials according to the defined timetable for the release.

2.2 THE RELEASE MANAGER AND TOOL SUBMITTER INTERFACE

The mode of interaction between the Submitter and the Release Manager will, for the most part, consist of the exchange of memos, notes, listings, and documentation. Memos and notes will usually be sent to indicate either the availability of tool materials for the Release Manager or the status of

07/25/80

SES Release Process

2.0 THE RELEASE PROCESSES IN GENERAL

2.2 THE RELEASE MANAGER AND TOOL SUBMITTER INTERFACE

certain release process tasks being performed by the Release Manager. All such memos and notes should be clear, concise, and complete to eliminate any need for additional communication on the matter at hand.

2.3 RELEASE TYPES

Releases are designated by number and are usually scheduled to occur at six months intervals. (For more specific dates refer to the SES PERT.) The interval within which the release activities are performed is referred to as the Release Period and designates that segment of time in which specific SES projects are scheduled for completion and in which the official release process of the developed tools takes place. This also provides a definite reference point by which the availability of those tools may be expected.

Within any given six month duration, there are only three types of releases that are conducted by the Release Manager:

Scheduled Release

Pre-Release

Critical PSR Release

A Scheduled Release is that type of release which is performed during a designated Release Period. Since any number of tools are slated for release at this time, the release process follows a definite timetable covering approximately one man-month. The term "scheduled" has two implications here: that the point in time of a release has been pre-determined, and that the activities leading to this point have also been specifically laid out on a timetable. The intention of the scheduling is to facilitate the execution of each step in the release process for each tool submitted. Any tool that is scheduled for release must be in the pre-release at least ten days prior to Release Day (i.e. Day 10 or the Pre-Release Cut-Off Day).

A Pre-Release is designed to accommodate those situations in which a tool is made available in the interim between release intervals. The main reason for this type of release is to allow projects to release code with new features or minor bug

07/25/80

SES Release Process

2.0 THE RELEASE PROCESSES IN GENERAL

2.3 RELEASE TYPES

fixes on a more timely fashion. Pre-release information will be made available to SES Users who have expressed concern over the availability of specific tools. The general SES User Community will receive the information on these tools only when requested from the Release Manager, the Tool Submitter, or the Project Leader. Pre-Release Tools are not extensively tested. If problems are found, they should be reported directly to the Project Leader.

A Critical PSR Release occurs when a critical bug is found in a current release and a temporary fix file is accepted from the Tool Submitter which is placed in the SES Catalog. The Submitter must eventually follow through the Scheduled Release process in order to make the fix official.

The basic differences among the three release types is primarily a factor of when the tool is to be made available and for what reason. However, there are a few other components that further distinguish each type. The following table illustrates these differences:

BASIC DIFFERENCES IN THE
RELEASE TYPES

SCHEDULED RELEASE

Tool made available
after the formal
Release Period

Tool is built by the
Release Manager

TAB blurb required
and User Handbook
corrections expected

Access is default via
SES procedure calls

PRE-RELEASE

Tool made available
before the formal
Release Period

Tool is built by the
Tool Submitter or
Release Manager

TAB blurb required

Access is through a
procedure call from
the SSS PROCLIB file

CRITICAL PSR RELEASE

Tool fix made available
immediately

Tool is built by the
Tool Submitter

TAB blurb required

Access is default via
SES procedure calls
(SES proc altered)

07/25/80

SES Release Process

2.0 THE RELEASE PROCESSES IN GENERAL

2.4 RELEASE TASK AREAS

2.4 RELEASE TASK AREAS

The activities in the execution of the release of any tool are shared between the Release Manager and the Tool Submitter. In the release process, there are four identifiable areas of activity:

- > The acquisition of Tool documentation
- > The build and verification of these tools
- > Distribution of released tools
- > Support for released tools

2.4.1 DOCUMENTATION REQUIREMENTS

In the initiation of a release, the first and foremost requirement is the submission of documentation that properly describes the new tools and/or changes to existing tools. Common to all three release types is a document offering a summary or "blurb" of the Tool's condition or situation. Each synopsis will simply consist of TXTCODE source containing a title line and a paragraph that describes the tool's function and/or its effected changes. The collection of all of these individual synopses under the title line

TOOL AVAILABILITY BULLETIN

and arranged by Tool name in alphabetical order will then compose the general construction of the Tool Availability Bulletin (or TAB).

All such "blurbs" should be sent to the Release Manager before the Pre-Release Cut-Off Day as the TAB will be made available on-line and also sent out to the general SES Community at that time.

In the particular case of a Scheduled Release, material for the SES User's Handbook should be received before other build related materials are made available. Usually, this documentation is included on a deck in the Tool's (main) PL. Other documents which may also be part of that PL and that can be submitted at this time are: the Tool's External Reference Specification (ERS) and the Tool Design Document --if available.

SES Release Process

2.0 THE RELEASE PROCESSES IN GENERAL

2.4.1 DOCUMENTATION REQUIREMENTS

The selection and the printing of listings is primarily the responsibility of the Submitter. Under normal conditions, any necessary listing for a tool should be generated during the execution of the build procedure or given to the Release Manager before the occurrence of the tool's release.

The following is a list of required documents --some of which may be generated by the Submitter separate from the build procedures that are expected during the course of the scheduled release process:

<u>Document</u>	<u>Made Available</u>	<u>For</u>
External Reference Specification (ERS)	<u>before</u> the Tool is released	TOOLDOC
SES User Handbook corrections or additions	<u>before</u> the build process for the Tool is started	---
Design Document (if applicable)	<u>before</u> the Tool is released	TOOLDOC
Compiled Tool Source with cross-reference tables and expanded common decks	<u>during</u> the build process	Racks
Load Map	<u>during</u> the build process	Racks

All such documentation received from the Submitter will be kept in a listing binder for common access by the general SES Community. Each tool will be kept in a separate binder and placed in an area designated as the "SES Library Area". Listings for each tool will replace old listings as they are accumulated by the Release Manager.

07/25/80

SES Release Process

2.0 THE RELEASE PROCESSES IN GENERAL

2.4.2 BUILD AND VERIFICATION

2.4.2 BUILD AND VERIFICATION

The heart of a the Scheduled Release process is the verification build of the tools being released. The series of events that occur here is briefly outlined below:

- > The Tool Submitter notifies the Release Manager of the readiness of the Tool for build.
- > The Release Manager (or his designate) acquires a copy of the Tool PL and performs the build.
- > When the build has been completed, the Submitter is notified, and the resulting build files are made available to him for verification.
- > Depending upon the outcome of the tests, the Release Manager will either accept the Tool for official release, or freeze it, allow the Submitter to make corrective changes, and then repeat the build and verification sequence.

The Pre-Release and Critical PSR Release do not involve as much participation on the part of the Release Manager. The Submitter will perform the tool build and verification from his own catalog. When satisfied with the results, he then presents the Tool to the Release Manager for "pre-release".

2.4.3 TOOL AVAILABILITY

Tool availability occurs in two steps. In the first step, all tools that are scheduled for release are required to be in pre-release a minimum of ten working days prior to that release. This point in time is referred to as the Pre-Release Cut-Off Day (Day 10 of the release schedule). In the second step, the tools are formally released. This occurs on the last day of the Release Period referred to as the Release Day (Day 0 of the release schedule).

Although there are many tasks that must be completed before the formal release of a tool, the single event which constitutes the actual release is the alteration of the SES PROCLIB file such that any procs associated with a released tool have been updated to reference the new version of the

07/25/80

SES Release Process

2.0 THE RELEASE PROCESSES IN GENERAL

2.4.3 TOOL AVAILABILITY

tool. Refer to the section titled "The Scheduled Release Timetable" for a further summary of the other tasks that precede this event.

However, in the pre-release of a tool, the procs associated with that tool are updated or added to the SSS PROCLIB file and referenced by a call in the form

SES,SSS.procname

It should be noted that in any of the three release types, all files associated with the execution of the tool are placed in the SES Catalog regardless of the location of the tool's procs (i.e. SES PROCLIB or SSS PROCLIB).

2.4.4 DISTRIBUTION

The purpose of the distribution is to provide non-local sites with tools in usable form. This implies the transmission of only the binary files of the tools and their supporting procedures. No information or files needed for building the tools is sent.

Tool distribution in conjunction with the Scheduled Release will be performed during the Post-Release Phase activities. The timetable for installation at remote sites will depend upon site conditions but hopefully would occur as soon as possible after the completion of the local release.

The Release Manager is responsible for transmission of copies of the new binaries to the remote sites. Tool installation at these sites is the responsibility of the local SES representative at each site.

2.4.5 TOOL SUPPORT

Only the current version of any tool will be considered for support. Tool support is implied by the category in which the tool is classified: Category I, Category II, or Category III.

Category I of support is provided when PSRs, RSMs, and SIEs written against a tool are regularly reviewed and action is taken to process each item according to the availability of resources and criticality. This is normally a tool with

SES Release Process

2.0 THE RELEASE PROCESSES IN GENERAL

2.4.5 TOOL SUPPORT

ongoing development activity.

Category II support involves periodic review of the PSRs, RSMs, and SIEs written against a tool will prompt action to process these items based on criticality and impact on the User Community. This is normally a tool with no current development activity. Tools in this category may be used for some special purpose but should be avoided in the normal build process if possible.

Category III implies no support. No reported problems are reviewed or acted upon. This is normally a tool with no further development planned and should not be used in any phase of a build process. Tools in this category have usually been replaced by new tools that provide the same basic capabilities.

The concept of tool support with respect to the Release Manager simply constitutes the management of the Tool Catalogs and the Tool Catalog Notebook as described in the section titled "Tool Catalog Management".

07/25/80

SES Release Process3.0 THE BUILD PHASE AND RELATED ITEMS3.0 THE BUILD PHASE AND RELATED ITEMS

To begin any build of a tool, the Release Manager must be notified by the Tool Submitter indicating the availability of the tool files for build and eventual release. This note can be sent to the Release Manager through SES.MAIL. Basic information that should be relayed is the formal name of the tool and the Submitter's name and phone extension. Subsequent items that should then be addressed are:

- > The Tool PL(s) -- particularly the PL containing the Tool Build Document (if applicable to the tool), and/or the Tool's build proc(s).
- > Any special instructions needed to generate the Tool Build Document, and/or to acquire the Tool Build Proc.
- > The user's catalog in which the Tool PL is stored.
- > To which user catalog the files resulting from the build should be made available for verification.
- > Who is to be notified when the files are ready for verification.
- > Any additional comments concerning the accessing of the PL contents, execution of the build, or in sending a return notification.

3.1 THE TOOL BUILD DOCUMENT

Because of the complexity of a tool's composition, a Tool Build Document (TBD) may be necessary to outline or direct any potential builder of that tool in the steps required to properly construct that tool. The TBD is usually a TEXTCODED document of information and instructions usually kept in source form on a deck --usually named BLDDOC-- in the tool's main sourc PL. Its contents should clearly lay out the build process for the tool. Some of the basic information provided by the TBD (if available) is:

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELATED ITEMS

3.1 THE TOOL BUILD DOCUMENT

- > the formal name of the tool,
- > specific file requirements,
- > specific 'result' files of the build process,
- > reference to build procs that are available,
- > special external file/tool/product dependencies, and
- > listings to be generated.

The main purpose of this document is to provide the Release Manager with a single source that will explain all he needs to know about the building of the tool and the organizing of its release. For less complex tools, a build proc may be the only "documentation" of the tool's build process.

In either case, a tool's build proc is usually kept on a deck --usually named BUILD-- in the main source PL of the tool. When several such build procs are necessary, they are usually named BLDxxxx where xxxx is used to distinguish each proc in the build process. All build procs for every tool are expected to have HELP information built into it to indicate what tool or part of a tool is being built and, more importantly, what parameters are available with the proc, and subsequently, their default values.

3.2 BUILD PROCEDURE GUIDELINES

The build procedure consists of a collection of control statements that will set the execution of the "build" of the tool into motion. Such procedures should be constructed in the manner of a structured program where a main procedure directs the control of other sub-procedures, files, and tools that support the overall build. The access fabrication of any procedure for the build is subject to several general requirements concerning its execute line, naming conventions for local files, and its relationship to a batch environment.

Build procedures should use SES procs where possible and should be designed to run either as a batch job or interactively. Also, these procedures should be designed to accommodate use by the Tool Submitter in his daily operations.

As many builds will be performed under a single catalog and because many files will be generated, the following naming convention should be used to label permanent files that will

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELATED ITEMS

3.2 BUILD PROCEDURE GUIDELINES

be of particular concern to the Release Manager --for example, compiled listings of tool source. The purpose of this naming convention is to facilitate the task of file identification and dispersement with respect to the tool to which it is associated. The convention is as follows--

All files created in the build process pertinent to the tasks of the Release Manager will be named in the form:

xyyyyyy where xx is the two-character tool identifier and yyyyy is supplied by the Submitter for file name uniqueness within the build procedure.

The names of the files that will later be transferred to the Tool Catalogs must be passed as parameters to the build procedure. File name substitution will be restricted to this classification of files and that the parameter list will be kept as limited as possible.

If a build procedure completes normally, it should create a local file containing a dump of the NOS Dayfile. The named assigned to this file should be of the form:

xyyyyOK

If a build procedure terminates abnormally, it should create and save a file containing a dump of the NOS Dayfile and should be of the form:

xyyyERR

Local files used as scratch files during the build should use

UNIQUE(NAME)

to generate a random, unique file name.

A typical build procedure should have the form of an SES procedure file, and can be divided into four basic parts --which are illustrated below:

SAMPLE_BUILD_PROCEDURE

1. To begin with, the build proc must have HELP Documentation built into it so that a user will have some immediate means of determining what the proc does and how it is initiated. Such documentation is

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELATED ITEMS

3.2 BUILD PROCEDURE GUIDELINES

constructed as follows:

BLDUT

```
\ IF MODE = HELP THEN
\  ROUT FA = PRIMOUT
```

This procedure will direct the build of the PL CROSS REFERENCE UTILITY.

The parameters are defined below:

PARAMETERS	DEFAULT	DESCRIPTION
utpl	SES1054	the source PL
utb1	SES50BF	.
utb2	SES50C0	.
utd1	SES50C1	.
utd2	.	.
utd3	.	.
pr	no print	print all listings

```
\  ROUTEND PRIMOUT
\  STOP
\ IFEND
```

2. The next section of the proc usually establishes parameter definitions, default values for the parameters, and unique names and labels to be used throughout the proc:

```
\  PARM KEY = 'utpl'   NVALS = 0..1  NAM
\  PARM KEY = 'utb1'   NVALS = 0..1  NAM
\  PARM KEY = 'utb2'   NVALS = 0..1  NAM
\  PARM KEY = 'utd1'   NVALS = 0..1  NAM
\  PARM KEY = 'utd2'   NVALS = 0..1  NAM
\  PARM KEY = 'utd3'   NVALS = 0..1  NAM
\  PARM KEY = 'pr'     NVALS = 0
```

\PARMEND

```
\  utpl      = SETVAL ('SES1054', notused, utpl)
\  utb1      = SETVAL ('SES50BF', notused, utb1)
\  utb2      = SETVAL ('SES50C0', notused, utb2)
\  utd1      = SETVAL ('SES50C1', notused, utd1)
\  utd2      .
\  utd3      .
```

```
\  compile = UNIQUE(NAME)
```

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELATED ITEMS

3.2 BUILD PROCEDURE GUIDELINES

3. The body of the proc consists of the gencomps, compiles, and loads used to generate object code for the tool being built. In this portion of the proc other items such as printing of listings and maps, saving specified files, etc. are also addressed. The layout and grouping of commands is left to the discretion of the proc writer. A proc body is illustrated as follows:

```

SESMSG. *** BEGIN THE BUILD
ACQUIRE(pl=&utpl&)
ses.gencomp m=readsrc b=pl ab=((pascmn,ses)) cf=&c
ses.pascalx i=&compile& l=utl1st b=lgo cc
ses.link170 pasclib f=lgo b=&utb1& l=utl1map
ses.rewrite i=&utb1& o=&utb1& CT=s M=e
$RETURN(&compile&,lgo)
\ IF DEFP(PR) THEN
ses.print (utl1st,utl1map)
\ IFEND
$RETURN(utl1st,utl1map,&utb1&)
SESMSG. ** FIRST BINARY COMPLETED

ses.gencomp m=format b=pl
ses.pascalx i=compile l=utl21st
ses.link170 pasclib f=lgo b=&utb2& l=utl2map
ses.rewrite i=&utb2& o=&utb2& CT=s M=e
$RETURN(compile,lgo)
\ IF DEFP(PR) THEN
ses.print (utl21st,utl2map)
\ IFEND
$RETURN(utl21st,utl2map,&utb2&)
SESMSG. ** SECOND BINARY COMPLETED

ses.gencomp m=sesdir1 b=pl cf=&utd1&
ses.rewrite i=&utd1& o=&utd1& CT=s M=r
ses.gencomp i=sesdir2 b=pl cf=&utd2&
ses.rewrite i=&utd2& o=&utd2& CT=s M=r
ses.gencomp m=sesdir3 b=pl g=&utd3&
ses.rewrite i=&utd3& o=&utd3& CT=s M=r
$RETURN(&utd1&,&utd2&,&utd3&)
SESMSG. ** COMPLETED DIRECTORY FILES

```

Notice also how SESMSG is used throughout to continuously relay the flow of events to the terminal user as each set or portion of the command flow is executed. The use of this feature is particularly beneficial to the tool builder (i.e. the Release Manager) who is monitoring the

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS
 3.2 BUILD PROCEDURE GUIDELINES

build's progress.

4. The last part of the proc usually concludes the build by returning all unnecessary files from the working space and then constructs a copy of the job's dayfile. Usually, when a proc is run from a terminal, the dayfile is retained only as a local file. If an error occurs or the build proc is executed in batch mode, the dayfile is retained as a permanent file. The last portion of the proc may look like this:

```

SESMSG. >>>> BLDUT COMPLETED
ses.dayfile find='BEGIN THE BUILD' n=999 o=ut99ok
EXIT.
SESMSG. <<<< BLDUT ERRORS ENCOUNTERED
ses.dayfile find='BEGIN THE BUILD' n=999 o=utdayf
ses.rewrite i=utdayf o=ut99err
$RETURN(utdayf)

```

Additional examples are included in the sub-section titled: "Examples" to further clarify build proc construction.

3.3 INITIATION OF THE BUILD PROCESS

In general, the Release Manager will perform a build according to the directions presented in the build documentation from the Tool Submitter.

Where the actual execution of the build may be very simple, the bulk of the work for the Release Manager involves several important preparatory and follow-up activities. These activities are keyed off of the Tool Build Document (TBD) and/or the HELP-Mode information from the Tool's build proc(s).

To print the TBD, the Release Manager must process the document through the text formatter and print it --either locally or at a printer. With this document certain build-related information can be drawn out before the build is performed. The Release Manager should be able to identify:

- > Tool PL(s) required for the build,
- > Named File and Numbered File assignments, and
- > special file/product/tool dependencies of the build procedure.

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS

3.3 INITIATION OF THE BUILD PROCESS

Upon completion of the build the Release Manager should be able to identify from this document:

- > local files resulting from the build,
- > files that are to be saved for verification,
- > files to be printed out from the build procedure, and
- > how these files are to be saved.

The HELP information provided in the build proc can be printed at the terminal during build time by stating:

```
ses,help.procname
```

This information should be able to explain to the builder (Release Manager) all parameter options for the proc, their consequences (especially when the keyword name is not implicitly clear), and their default values when applicable.

Based on the information from the TBD (if available), any special instructions on the Submitter's notice, and/or the HELP information in the Tool's build proc, the build environment can be established and the build execution can begin. In brief, the series of tasks involved would be as follows:

- > Identify the file requirements of the build procedure. (HELP information)
- > See to it that these files are present or will be accessible during the build execution.
- > Execute the build procedure(s).
- > Identify and save the resulting tool files.
- > Gather and bind the tool listings printed out.
- > Notify the Submitter of the build completion.

When the build execution has completed, all files from the build are retained until tool verification indicates a successful build. If the verification is positive, the files may be purged from the build catalog.

The build procedure execution should always produce a listing(s) --which may be optionally printed with the specification of a 'print' keyword. As is applicable, these listings should be "name coded" according to the established conventions in order to distinguish the different phases or parts of the tool build-output. (See the sub-section entitled "Build Procedure Outline" for an explanation of this naming convention.) All listings are to be retrieved from the printer and separated according to tool. Listings should not be stored as permanent files once printed. Each set of listings for the tool is then bound and placed in the common library area at the central site.

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELATED ITEMS

3.3 INITIATION OF THE BUILD PROCESS

At this point, the Submitter is notified that the build is complete and that the tool files are available to him for verification. Information that is relayed to the Submitter usually indicates:

- > the formal name of the tool,
- > the list of tool files resulting from the build that are presently available in the build catalog for verification, and
- > additional comments or informative notes indicating deviations from the specified build proceedings.

It is expected that all files in the build catalog will have semi-private and read-only/execute-only permissions. No general write permissions will be allowed.

3.4 VERIFICATION

The first and foremost purpose of the Release Manager's build in the release process is to ascertain that the construction of the tool was self-contained --that is, the build process used only materials provided by the Tool Submitter and/or from standard/supported products available.

Verification of the results of a tool's build should determine whether or not the tool that was produced is the same as the corresponding tool in pre-release. The verification is usually made in the following two forms: file contents comparison and/or tool testing. File contents verification involves the comparison of contents of the newly built tool file(s) and each counter-part in the SES Catalog. the NDS command VERIFY is usually used to accomplish this. Tool testing involves the actual exercising of the tool from the files that were just built.

The responsibility of file comparison and tool testing lies with the Submitter. The testing of the tool itself should have been done prior to submitting the tool for release. It is expected that specific tests will be developed by the Submitter for each tool. No particular restrictions are made on test composition and construction by this document or the release process. It is left to the Tool Submitter to determine the degree and the extent to which he wishes to test the tool since only he is accountable for tool performance after the official release occurs. However, testing should cover all pertinent files resulting from that build.

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELATED ITEMS
 3.4 VERIFICATION

The Submitter will inform the Release Manager whether or not the tool was built properly by indicating the condition of the tool files resulting from the build, and will make a recommendation for abort, postponement, or a go-ahead of the tool's release. Based on this report, the Release Manager then decides whether to allow the release of that tool or to "recycle" the tool into pre-release.

3.5 EXAMPLES OF BUILD PROCEDURES

The following examples are provided to exemplify the build procedure guidelines discussed in the previous sub-sections. A brief summary precedes each example to describe the situation being depicted and/or to point out items of interest.

3.5.1 SAMPLE PROCEDURE --- ACQUIRE

This is an example of the build proc that will compile and link the ACQUIRE Program. Note in particular how the HELP Mode information is laid out. Also, how the default values for the parameters are set using SETVAL, the use of 'UNIQUE' to generate unique names for scratch files (and labels), and the use of SESMSG to indicate the status of the build throughout the proc. The body of the proc contains a mixture of 'SES calls' and KCL Commands. The use of the 'R-registers' is employed to aid the proc in determining error conditions otherwise transparent to the proc. The use of this feature is left to the discretion of the proc writer and is not a necessity when constructing a build proc.

BLDACQ

```
\ IF MODE = HELP THEN
\   ROUT FA = PRIMOUT
```

This BUILD procedure compiles and links the ACQUIRE program

PARAMETER	DEFAULT	ALLOWABLE VALUE(S)
pl	SES1080	filename (of source PL)

COMPANY PRIVATE

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS

3.5.1 SAMPLE PROCEDURE --- ACQUIRE

plun	SESAUX	username (owner of pl)
bin	SES5140	filename (for binary)
binun	current user	username (owner of bin)
list	LISTING	filename (for listing)
pr	no print	keyword
dayf	no dayfile	keyword

In addition, all of the standard SES batch job parameters are available including the dayfile parameter (default is to run the proc in LOCAL mode). If the build is successful, the dayfile will be left on a local file (default name is DAYFILE) If the build doesn't work or the proc is run as a batch job, the dayfile will be saved in a permanent file in the current user's catalog.

```

\   ROUTEND PRIMOUT
\   STOP
\   IFEND

\   PARM   KEY = 'pl'       NVALS = 1   NAM
\   PARM   KEY = 'plun'    NVALS = 1   STR
\   PARM   KEY = 'bin'     NVALS = 1   NAM
\   PARM   KEY = 'binun'   NVALS = 1   STR
\   PARM   KEY = 'list'   NVALS = 1   NAM
\   PARM   KEY = 'pr'     NVALS = 0
\   INCLUDE 'JOBPARM' L=UNIQUE(NAME) LPFN=SESLNAM UN=SESUNAM
\   PARMEND

\   INCLUDE 'JOBHDR1' L=UNIQUE(NAME) LPFN=SESLNAM UN=SESUNAM
\   INCLUDE 'JOBHDR2' L=UNIQUE(NAME) LPFN=SESLNAM UN=SESUNAM

\   pl      = SETVAL('SES1080', notused, pl)
\   plun    = SETVAL('SESAUX', notused, plun)
\   bin     = SETVAL('SES5140', notused, bin)
\   binun   = SETVAL(USER, notused, binun)
\   list    = SETVAL('LISTING', notused, list)

\   compile = UNIQUE(NAME)
\   lgo      = UNIQUE(NAME)
\   lgob     = UNIQUE(NAME)

\   error   = UNIQUE(LABEL)
\   groovy  = UNIQUE(LABEL)
\   done    = UNIQUE(LABEL)
\   solong  = UNIQUE(LABEL)

SESMMSG.*   BUILDING ACQUIRE
    
```

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS

3.5.1 SAMPLE PROCEDURE --- ACQUIRE

```

SES.GENCOMP, M=ACQUIRE, CF=&compile&, B=&pl&, UN=&plun&, ..
      NOMSG, STATUS=EF.
$IF(EF.NE.0)$GOTO(&error&)
SES.COMPASS, I=&compile&, L=&list&, B=&lgo&, NOMSG
$IF(EF.NE.0)$GOTO(&error&)
SES.GENCOMP, M=MACQUIR, CF=&compile&, B=&pl&, UN=&plun&, ..
      PASCCMN, NOMSG, STATUS=EF.
$IF(EF.NE.0)$GOTO(&error&)
SES.PASCALX, I=&compile&, L=&list&, LO=R, B=&lgo&, NOMSG
$IF(EF.NE.0)$GOTO(&error&)
SES.LINK170, F=&lgo&, B=&lgo&, PASCLIB, L=&list&, LO, ..
      EP='ACQUIRE,$RFL=$,$SDM=$'.
$IF(EF.NE.0)$GOTO(&error&)
SES.REWRITE, I=&lgo&, O=&bin&, UN=&binun&, ..
      FAILED='&error&', LOOPEND='&error&'.
$SET(R1=0)
$GOTO(&groovy&)

EXIT.
&error&,*
SESMSG.-      ERROR(S) IN BUILD OF ACQUIRE
$SET(R1=1)

&groovy&,*
\ IF DEFK(pr) THEN
$IF(FILE(&list&,.NOT.AS))$GOTO(&done&)
SES.PRINT, F=&list&, NOSHIFT, ID='ACQUIRE'
\ IFEND

&done&,*
$RETURN(&compile&,&lgo&,&lgo&,&bin&)

\ IF DEFK(dayf) THEN
\   IF jobmode = 'LOCAL' THEN
SES.DAYFILE, N=9999, FIND='BUILDING ACQUIRE', O=&dayfile&
EDT(&dayfile&,0)D.S;*.S;-1.D
$IF(R1.EQ.0)$GOTO(&solong&)
\   ELSE
$DAYFILE(&dayfile&)
\   IFEND
$RENAME(&compile&=&dayfile&)
SES.REWRITE, I=&compile&, O=&dayfile&
$RETURN(&compile&)
\ IFEND

&solong&,*
$SET(EF=R1)
$SET(R1=&R1&)

```

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS

3.5.1 SAMPLE PROCEDURE --- ACQUIRE

* END BUILD ACQUIRE

3.5.2 SAMPLE PROCEDURE --- UTLITIY LIBRARY

BLDULIB

\ IF MODE = HELP THEN
 \ ROUT FA = PRIMOUT

This BUILD procedure compiles/assembles all of the components of the SES Utility Library for PASCAL-X CC programs and puts themn a User LIBrary ready for use with the CYBER Loader.

PARAMETER	DEFAULT	ALLOWABLE VALUE(S)
pl	SES109F	filename (of source PL)
plun	SESAUX	username (owner of pl)
ulib	SESULIB	filename (for library)
ulibun	current user	username (owner of ulib)
list	LISTING	filename (for listing)
pr	no print	keyword
dayf	no dayfile	keyword

In addition, all of the standard SES batch job parameters are available including the dayfile parameter (default is to run the proc in LOCAL mode). If the build is successful, the dayfile will be left on a local file (default name is DAYFILE). If the build doesn't work or the proc is run as a batch job, the dayfile will be saved in a permanent file in the current user's catalog.

\ ROUTEND PRIMOUT
 \ STOP
 \ IFEND

\ PARM KEY = 'pl' NVALS = 1 NAM
 \ PARM KEY = 'plun' NVALS = 1 STR
 \ PARM KEY = 'ulib' NVALS = 1 NAM
 \ PARM KEY = 'ulibun' NVALS = 1 STR
 \ PARM KEY = 'list' NVALS = 1 NAM
 \ PARM KEY = 'pr' NVALS = 0
 \ PARM KEY = 'dayf' NVALS = 0
 \ INCLUDE 'JOBPARM' L=UNIQUE(NAME) LPFN=SESLNAM UN=SESUNAM
 \ PARMEND

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS

3.5.2 SAMPLE PROCEDURE --- UTLITIY LIBRARY

```

\ INCLUDE 'JOBHDR1' L=UNIQUE(NAME) LPFN=SESLNAM UN=SESUNAM
\ INCLUDE 'JOBHDR2' L=UNIQUE(NAME) LPFN=SESLNAM UN=SESUNAM

\ pl      = SETVAL('SES109F', notused, pl)
\ plun    = SETVAL('SESAUX', notused, plun)
\ ulib    = SETVAL('SESULIB', notused, ulib)
\ ulibun  = SETVAL(USER, notused, ulibun)
\ list    = SETVAL('LISTING', notused, list)

\ compile = UNIQUE(NAME)
\ lgo     = UNIQUE(NAME)
\ templib = UNIQUE(NAME)
\ error   = UNIQUE(LABEL)
\ groovy  = UNIQUE(LABEL)
\ done    = UNIQUE(LABEL)
\ solong  = UNIQUE(LABEL)

```

SESMG.* BUILDING SES UTILITY LIBRARY

```

SES.GENCOMP, M=(BLNKSTR..CVASC64, ZBIICLO..ZBIIWEO, ZDIICLO..ZDIISTF, ..
               ZIOICLO..ZIOIQRY, ZLGICLO..ZLGIWEO, ZPRIOPN..ZPRIOUT), ..
               CF=&compile&, B=&pl&, UN=&plun&, ..
               NOMSG, STATUS=EF.
$IF(EF.NE.0)$GOTO(&error&)
SES.PRINTID, ID='SESULIB/PXIO/COMPASS//)DATE', O=&list&
SES.COMPASS, I=&compile&, L=&list&, B=&lgo&, NOMSG
$IF(EF.NE.0)$GOTO(&error&)
SES.GENCOMP, M=(ZCLIIAP), ..
               CF=&compile&, B=&pl&, UN=&plun&, ..
               NOMSG, STATUS=EF.
$IF(EF.NE.0)$GOTO(&error&)
SES.PRINTID, ID='SESULIB/SCL/COMPASS//)DATE', O=&list&
SES.COMPASS, I=&compile&, L=&list&, B=&lgo&, NOMSG
$IF(EF.NE.0)$GOTO(&error&)
SES.GENCOMP, M=(ZCLMANT..ZCLMXFT, ZOSMEND..ZOSMSSA, ZPMMDAT..ZPMMTIM), ..
               CF=&compile&, B=&pl&, UN=&plun&, ..
               NOMSG, STATUS=EF.
$IF(EF.NE.0)$GOTO(&error&)
SES.PRINTID, ID='SESULIB/SCL/PASCAL-X//)DATE', O=&list&
SES.PASCALX, CC, I=&compile&, L=&list&, B=&lgo&, NOMSG
$IF(EF.NE.0)$GOTO(&error&)
SES.GENCOMP, M=(ZN7ICIO..ZN7IWNB, ZUTIABT..ZUTISFN), ..
               CF=&compile&, B=&pl&, UN=&plun&, AB=((DPL, LIBRARY)), ..
               NOMSG, STATUS=EF.
$IF(EF.NE.0)$GOTO(&error&)
SES.PRINTID, ID='SESULIB/MISC./COMPASS//)DATE', O=&list&
SES.COMPASS, I=&compile&, L=&list&, B=&lgo&, NOMSG

```

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS

3.5.2 SAMPLE PROCEDURE --- UTLITIY LIBRARY

```

$IF(EF.NE.0)$GOTO(&error&)
SES.GENCDMP, M=(ZN7MAQR..ZN7MRDR, ZUTMAQR..ZUTMW2D), ..
      CF=&compile&, B=&pl&, UN=&plun&, ..
      NOMSG, STATUS=EF.
$IF(EF.NE.0)$GOTO(&error&)
SES.PRINTID, ID='SESULIB/MISC./PASCAL-X//)DATE', O=&list&
SES.PASCALX, CC, I=&compile&, L=&list&, B=&lgo&, NOMSG
$IF(EF.NE.0)$GOTO(&error&)
$LIBGEN(F=&lgo&,P=&templib&,N=SESULIB,NX=1)
SES.REWRITE, I=&templib&, O=&ulib&, UN=&ulibun&, ..
      FAILED='&error&', LOOPEND='&error&'.

$SET(R1=0)
$GOTO(&groovy&)

EXIT.
&error&,*
SEMSG.-      ERROR(S) IN BUILD OF SES UTILITY LIBRARY
$SET(R1=1)

&groovy&,*
\ IF DEFK(pr) THEN
$IF(FILE(&list&,.NOT.AS))$GOTO(&done&)
SES.PRINT, F=&list&, NOSHIFT, ID='SES/UTILITY/LIBRARY//)DATE'
\ IFEND

&done&,*
$RETURN(&compile&,&lgo&,&templib&,&ulib&)

\ IF DEFK(dayf) THEN
\   IF jobmode = 'LOCAL' THEN
SES.DAYFILE, N=9999, FIND='BUILDING SES UTIL', O=&dayfile&
EDT(&dayfile&,0)D.S;*.S;-1.D
$IF(R1.EQ.0)$GOTO(&solong&)
\   ELSE
$DAYFILE(&dayfile&)
\   IFEND
$RENAME(&compile&=&dayfile&)
SES.REWRITE, I=&compile&, O=&dayfile&
$RETURN(&compile&)
\ IFEND

&solong&,*
$SET(EF=R1)
$SET(R1=&R1&)
*      END BUILD      SES UTILITY LIBRARY

```


07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS

3.5.3 SAMPLE PROCEDURE --- VE LINKER

3.5.3 SAMPLE PROCEDURE --- VE LINKER

BLDLD

```
\ IF MODE = HELP THEN
\   ROUT FA = PRIMOUT
```

This procedure compiles the modules for the VE Linker.
It's parameters are shown below:

PARAMETER	DEFAULT	DEFINITION
ldpl	SES106C	source PL
ldol	SES5118	object library containing relocatables
ldov	SES5117	relocatable containing the overlay bin
pr	no print	when specified, will print all listing

```
\   ROUTEND PRIMOUT
\   STOP
\ IFEND
```

```
\ PARM KEY = 'ldpl' NVALS = 0..1 NAM
\ PARM KEY = 'ldol' NVALS = 0..1 NAM
\ PARM KEY = 'ldov' NVALS = 0..1 NAM
\ PARM KEY = 'pr' NVALS = 0 NAM
\ PARM KEY = 'dayf' NVALS = 0 NAM
\ PARMEND
```

```
\ ldpl = SETVAL ('SES106C', notused, ldpl)
\ ldol = SETVAL ('SES5118', notused, ldol)
\ ldov = SETVAL ('ses5117', notused, ldov)
```

```
SETTL,7777.
RETURN,LISTING,LGO.
RETURN,LDERROR.
RFL,20000.
```

```
** COMPILE THE LINKER AND ITS OVERLAY SCHEME
SES.GENCOMP ZLDM000 B=&ldpl& AB=((SESPLXX,SES))
SES.ISWL LARGE LO=R L=LINKLST
SES.GENCOMP ZLDM100 B=&ldpl& AB=((SESPLXX,SES))
SES.ISWL LO=R L=LINKLST
SES.GENCOMP ZLDM200 B=&ldpl& AB=((SESPLXX,SES))
SES.ISWL LO=R L=LINKLST
SES.GENCOMP ZLDM300 B=&ldpl& AB=((SESPLXX,SES))
SES.ISWL LO=R L=LINKLST
```

07/25/80

SES Release Process

3.0 THE BUILD PHASE AND RELEATED ITEMS

3.5.3 SAMPLE PROCEDURE --- VE LINKER

```
SES.GENCOMP ZLDMSSC B=&ldpI& AB=((SESPLXX,SES))
SES.ISWL LO=R L=LINKLST
```

```
\ IF DEFP(pr) THEN
SES.PRINT LINKLST
\ IFEND
```

```
SES.REPULIB G=LGD B=&ldoI& NX
RETURN,LGO,LINKLST.
SES.GENCOMP ZLDILNK B=&ldpI& AB=((SESPLXX,SES))
SES.COMPASS L=LINKLST
SES.GENCOMP ZLDMDUM B=&ldpI& AB=((SESPLXX,SES))
SES.ISWL LO=R L=LINKLST
SES.GENCOMP ZLDIOVL B=&ldpI& AB=((SESPLXX,SES))
SES.COMPASS L=LINKLST
SES.REWRITE I=LGD O=&ldov&
\ IF DEFP(pr) THEN
SES.PRINT LINKLST
\ IFEND
* BLDLD COMPLETED
```

```
EXIT.
PACK,LINKLST.
SES.REWRITE I=LINKLST O=LDLIST
SES.DAYFILE FIND='SETTL,7777.' N=999 O=LDERROR.
SES.REWRITE I=LDERROR
* BLDLD ERRORS ENCOUNTERED
```

07/25/80

SES Release Process
-----4.0 MISCELLANEOUS NOTES
-----4.0 MISCELLANEOUS NOTES4.1 TOOL MATERIAL ORGANIZATION

A tool must be completely contained on a minimum number of PLs (hopefully one). The composition of "this PL" must consist of "all" the tool source code necessary for tool generation, "all" the build procedures needed to perform the build, the source necessary to produce the documents required, and test cases for tool verification.

4.2 TOOL DEPENDENCIES

The dependence of one tool on others practically always turns out to be much more intricate and complicated than anyone at first supposes. However, it is possible to confine the dependencies to specific areas in the build procedures and therefore, make it much easier to insure that the proper versions of other required tools are used. A conscious effort should be made on the part of the build procedure writer to reduce the number of inter-tool dependencies to as few as possible.

4.3 CATALOG DEPENDENCIES

The problem of moving a tool from one catalog to another or of extracting required tools or files from the proper catalogs has long been a difficulty that has forced past release managers to constantly modify build procedures "on the fly". It is therefore expected that in all release procedures the need to acquire files--other than files that are submitted as part of the build process--will only be from the Tool Catalogs. In addition, such files should be supported tools or products.

07/25/80

SES Release Process

4.0 MISCELLANEOUS NOTES

4.3 CATALOG DEPENDENCIES
-----4.4 PSR CORRECTIONS

PSRs written against the product will be routed to the Tool Submitter via the PSR Coordinator. It will be up to the Submitter to inform the Release Manager of the situation. Arrangements for non-critical corrections should be made to coincide with a Scheduled Release.

Critical corrections are transmitted to the general SES User Community according to the following series of steps:

- 1 The creation of a substitute tool binary by the Submitter.
- 2 Give the Release Manager access to the new binary file.
- 3 The Release Manager will place the corrected version of tool in the SES Catalog.
- 4 The Tool Proc in SES PROCLIB will be adjusted to call the new version of the Tool.
- 5 The Tool Submitter will provide notification of this new version of the tool to the SES User Community through SES.INFO.

During the next Scheduled Release, the Submitter is expected to follow up with Scheduled Release procedures in order to properly document the change.

07/25/80

SES Release Process

5.0 THE SCHEDULED RELEASE TIMETABLE

5.0 THE SCHEDULED RELEASE TIMETABLE

It usually requires fifteen (15) working days to complete the Scheduled Release activities. Projects must submit materials and complete certain tasks by the days indicated in the schedule below --or sooner, if desired.

Although the exact schedule may vary within any given Release Period, the following layout indicates the general occurrence of events with respect to specific points in the release schedule activities. Day 10 marks the first working day of the Release Period. Day -5 is the last working day of that period.

before Day 10

The Submitter should have all of the essential coding complete and should be confident that the tool will execute properly.

The Submitter should have isolated and documented all the dependencies his tool has on other tools that are being co-released.

All changes for the SES User's Handbook documentation should be turned into the Release Manager at this time.

The Submitter will supply the Release Manager with a brief description of the new tool or change(s) from the old tool for inclusion in the Tool Availability Bulletin.

The Submitter should have completed the writing of a tool build document (if necessary), and the construction of build procedures.

The Submitter will inform the Release Manager of the availability of the tools for pre-release.

07/25/80

SES Release Process

5.0 THE SCHEDULED RELEASE TIMETABLE

Day 10

All tools that will be released will be set-up in the pre-release.

before Day 0

The Release Manager will make the TAB available to the SES User Community.

The Release Manager will make the new\updated ERS materials available to the SES User Community via TOOLDOC.

The Release Manager will have constructed and reviewed the SES User Handbook, presented it to DCS for distribution, and made it available to the SES User Community via TOOLDOC.

The Release Manager will begin verification builds of the tools and will notify the Submitter when the completed tool is ready to be verified. (This will be an on-going activity up until Day -5.)

Day 0

The Release Manager will give public access to the new tools.

before Day -5

The Submitter will have completed the verification that his tool was built correctly.

The Release Manager will complete the verification builds on all tools of the release.

The Release Manager will have created a set of tapes containing SES Tools and have sent them to the non-local sites.

It is important to note that in following any release schedule, it is expected that all designated tool materials

SES Release Process

5.0 THE SCHEDULED RELEASE TIMETABLE

will be available by the days specified in the schedule. However, it is acceptable to submit tool materials sooner than the schedule shows.

07/25/80

SES Release Process
*****6.0 TOOL CATALOG MANAGEMENT
*****6.0 TOOL CATALOG MANAGEMENT6.1 INTRODUCTION

The "Tool Catalog" is a collection of SES tools and files and actually consists of three catalogs: the principle catalog which is referred to as the SES Catalog, the SESAUX Catalog also referred to as the Auxillary Catalog, and the SSS Catalog which is the "pre-release" catalog.

The reason for the multiple catalogs is to provide the Release Manager with a means to control the archiving of tool files periodically done by COMSOURCE. The SES Catalog contains only those files necessary to execute the current and released versions of the tools. Since it is inconvenient to have files of this nature archived, the RETAIN proc is periodically executed to retain those files in the catalog. The SESAUX Catalog contains the remaining SES tool files --such as PLs and backup files or those files from previous SES releases. The SSS Catalog, by practice, holds only the PROCLIB containing the procedures for executing the Pre-Release versions of tools.

When newer versions of existng tools are released, tool files from the previous release and currently residing in the SES Catalog are transferred to the Auxillary Catalog. The tool files now in the Auxillary Catalog then become subject to the COMSOURCE archiving policies and thus, lends to the support of the release process objective which is to provide for reproducibility of any desired version of a tool.

6.2 CATALOG CONIENIS

The contents of the SES Catalog and the Auxillary Catalog are subdivided into two basic types --source files and binary files.

07/25/80

SES Release Process

6.0 TOOL CATALOG MANAGEMENT

6.2 CATALOG CONTENTS

Files in the Tool Catalogs are assigned unique names by the Release Manager to avoid naming problems and to provide a general identification of the file contents. The general form of the name is:

SESnnnn where nnnn represents the numbered portion of the name.

The unique names for submitted files allows the Submitter to properly identify and verify new files prior to release, and also ensures that "old" versions of tools can be retrieved from Archive.

Some files in the SES Catalog will have more meaningful names (e.g. PASCLIB) because they might be accessed explicitly by such names rather than through an SES Procs. The contents of these files are updated on Release Day. However, any specially named files are still assigned unique names for backup and retrieval capabilities of old versions.

6.3 BASIC RESPONSIBILITIES

Management of the Tool Catalogs as handled by the Release Manager entails the following responsibilities:

- > to assure that the latest version and/or level of any SES tool file is always available,
- > to make older versions and levels of any SES tool file available upon request--this usually implies the un-archiving of files,
- > to maintain an updated list of files in both catalogs,
- > to remove unnecessary files from either catalog,
- > to assure that proper access permission is established on all the files,
- > to assign numbered file backups to all Named Files,
- > to replace files that have "gone bad", and
- > to replace critical PSR corrected files.

6.4 TOOL CATALOG NOTEBOOK MANAGEMENT

The Tool Catalog Notebook is a textcoded record of all the numbered file assignments made to the Tool Catalogs.

COMPANY PRIVATE

07/25/80

SES Release Process

6.0 TOOL CATALOG MANAGEMENT

6.4 TOOL CATALOG NOTEBOOK MANAGEMENT

Management of this notebook simply involves the recording of the names assigned when files are requested by Tool Submitters. The file name assignments are made according to the conventions as described in the previous sub-section entitled "Tool Catalog Contents". Entries in this notebook are made to the existing hard copy by hand and then periodically updated in the on-line copy. Single copies of the notebook may be sent to remote SES sites when requested, but the notebook is not generally made available to SES Users. Only the Release Manager is allowed to enter information into the notebook and make file assignments from it. Users wishing to know or obtain file assignments have access to the notebook hard copy through the Release Manager. This notebook always resides in the Release Manager's Office and is never circulated or loaned out.

07/25/80

SES Release Process

A1.0 APPENDIX A TERMINOLOGY

A1.0 APPENDIX A TERMINOLOGY

Tool

A tool is a facility to aid in producing products with minimal expenditure of time, labor, and materials.

Product

A collection of files made available for company customer use (with support).

Software Tools

Software tools are defined as the software (or firmware) programs that aid in the development of software.

Software Engineering

A branch of engineering in the computer industry whose function is to plan the processes of manufacture, development, and integration of software with minimal expenditure of time, labor, and materials.

Software Engineering Services or
SES

A group of people providing software development tools for Advanced Systems Development.

SES Tool

A software tool developed by, supported by, and/or made available through SES.

Tool Identifier

A two-character identifier uniquely assigned to each SES tool for identification of files and for tracking PSR information.

SES Release Process

A1.0 APPENDIX A TERMINOLOGY

Tool PL

A program library containing the tool source code, build procedure, documentation, and test cases.

Tool Build Document or TBD

A deck on in the Tool PL named "BLDDOC" that usually consists of a single page of text identifying the tool and describing the requirements necessary to build it.

Release

The process by which SES Tools are made "available" for company internal use (with support).

Release Period

The time within which specific SES Projects are scheduled for completion and their products are made available to the general SES User Community. The Release Period consists of two parts: a Pre-Release Phase and a Post-Release Phase.

Pre-Release Phase

The time in which tools are made available before their scheduled release for other projects co-releasing tools.

Post-Release Phase

The time in which the Release Manager completes build activities and sends the released tools to the remote sites.

Pre-Release Cut-Off Day

The point in the Pre-Release Phase by which all tools scheduled to be released are placed in pre-release status. Also, referred to as "Day 10".

Release Day

The point in the release process at which new/revise tools are made available to the general SES User Community. Also, referred to as "Day 0".

07/25/80

SES Release Process

A1.0 APPENDIX A TERMINOLOGY

Aborted Release

A situation occurring before the official release in which submitted files have to be modified after tool verification discovers a problem with the tool.

SES Release Manager

The person who acts as the release co-ordinator and is in charge of maintaining the presence of SES Tools in the SES Catalog.

Submitter

The person responsible for the release and maintenance of an SES tool.

Tool Build Procedure

A sequence of control statements and/or control language statements that directs the file manipulations necessary to construct (build) a tool.

Tool Catalog(s)

The collective term used when referring to all of the SES Tool catalogs: SES, SSS, SESAUX.

SES Catalog

The collection of permanent files consisting of SES Tool binaries necessary to execute a tool. The "no-archive" bit is set for this catalog.

SESAUX Catalog

The collection of permanent files consisting of backup files and source PLs for SES Tools. This catalog is set to allow for archiving.

SSS Catalog

Holds only the PROCLIB file which contains the procedures necessary to execute Interim and PSR type versions of tools. This catalog is set to allow for archiving.

07/25/80

SES Release Process

A1.0 APPENDIX A TERMINOLOGY

SES Users

A group of people who use SES Tools.

Tool Availability Bulletin or
TAB

A bulletin identifying the new/changed tools of a release and briefly describes the new tool's operation and function or the user level changes made to an old tool.

Documentation Control System or
DCS

The agency used by SES for tool documentation distribution within CDC.

07/25/80

SES Release Process

B1.0 APPENDIX B BIBLIOGRAPHY

B1.0 APPENDIX B BIBLIOGRAPHY

- 1) The ALL Catalog Release Procedures
October, 1975 Revision A
from the NCR/CDC Advanced Systems Laboratory Joint
Venture
- 2) SES Operating Guide September, 1975
Section 1.0: Release Procedure.

Table of Contents

1.0	INTRODUCTION	1-1
2.0	THE RELEASE PROCESSES IN GENERAL	2-1
2.1	BASIC RESPONSIBILITIES	2-1
2.2	THE RELEASE MANAGER AND TOOL SUBMITTER INTERFACE	2-1
2.3	RELEASE TYPES	2-2
2.4	RELEASE TASK AREAS	2-4
2.4.1	DOCUMENTATION REQUIREMENTS	2-4
2.4.2	BUILD AND VERIFICATION	2-6
2.4.3	TOOL AVAILABILITY	2-6
2.4.4	DISTRIBUTION	2-7
2.4.5	TOOL SUPPORT	2-7
3.0	THE BUILD PHASE AND RELATED ITEMS	3-1
3.1	THE TOOL BUILD DOCUMENT	3-1
3.2	BUILD PROCEDURE GUIDELINES	3-2
3.3	INITIATION OF THE BUILD PROCESS	3-6
3.4	VERIFICATION	3-8
3.5	EXAMPLES OF BUILD PROCEDURES	3-9
3.5.1	SAMPLE PROCEDURE --- ACQUIRE	3-9
3.5.2	SAMPLE PROCEDURE --- UTILITY LIBRARY	3-12
3.5.3	SAMPLE PROCEDURE --- VE LINKER	3-15
4.0	MISCELLANEOUS NOTES	4-1
4.1	TOOL MATERIAL ORGANIZATION	4-1
4.2	TOOL DEPENDENCIES	4-1
4.3	CATALOG DEPENDENCIES	4-1
4.4	PSR CORRECTIONS	4-2
5.0	THE SCHEDULED RELEASE TIMETABLE	5-1
6.0	TOOL CATALOG MANAGEMENT	6-1
6.1	INTRODUCTION	6-1
6.2	CATALOG CONTENTS	6-1
6.3	BASIC RESPONSIBILITIES	6-2
6.4	TOOL CATALOG NOTEBOOK MANAGEMENT	6-2
A1.0	APPENDIX A TERMINOLOGY	A1-1
B1.0	APPENDIX B BIBLIOGRAPHY	B1-1