

TO: MSPM Distribution List  
FROM: A. Evans  
SUBJECT: Sections BE.1.01 and BE.1.02  
DATE: November 16, 1967

Attached are revisions of sections BE.1.01 and BE.1.02 of MSPM. These revisions reflect the following changes in the EPL command on CTSS:

- An intermediate pass, called pass 1.5, has been inserted between pass 1 and pass 2.
- A BRIEF option has been provided, permitting the user to suppress the verbose on-line output from EPL.

Note that pass 1.5 must be used whenever pass 1 is used -- no option is provided to circumvent this restriction. Be warned that the restriction is required, since pass 2 will not in all cases be able to operate successfully if pass 1.5 is not used.

Published: 11/04/67  
Supersedes: BN.3.01, 07/05/66)  
BE.1.01, 01/04/67

### Identification

Use of the EPL Compiler in CTSS  
A. Evans

### Usage

The command

R EPL NAME -options-

will cause the (ASCII) file NAME EPL to be compiled, producing the following three (ASCII) files:

NAME SOURCE A line-numbered listing of the source file, followed by a cross-referenced symbol table.

NAME MACROS Output of Pass 1 of the compiler and, therefore, input to Pass 2.

NAME EPLBSA The output of Pass 2 of the compiler, ready for assembly by EPLBSA.

The SOURCE and MACROS files will normally be left in temporary mode, although the user may direct otherwise. (See the discussion of options below.)

Many options are available to the EPL user -- they are discussed in detail below.

If no compilation is to be done, "\*" may be used instead of NAME. The run may still be used for updating options, determining compiler creation dates, etc.

If the first character of the file NAME EPL is the percent character, the program EPLMAC will be called before Pass 1 to process "include" macros. The file NAME EPLMAC which it produces is normally created in temporary mode, so it is deleted automatically after it has been read by pass 1. EPLMAC is described in Section BE.1.02.

Files with the following names may be produced as part of the EPL run, but they are deleted by the end of the run. Thus useful files with these names should not be in the user's directory.

EPL (TEMP)
NAME EPLMAC
NAME BSA
NAME BCD
NAME BSX

### Error Diagnostics

Syntactic errors in the user's EPL program may be detected in either Pass 1 or Pass 2, and they are indicated by error messages printed on the user's console. Pass 2 will be run even if errors are detected in Pass 1. EPL will comment if illegal option parameters are used, and will then refuse to update the profile.

### EPL Information On-Line

Because EPL is still under development, it has been found useful to maintain files giving the current status of the system. Several such files exist, each fulfilling a different purpose. All of the files are ASCII files. They are described here and summarized at the end of this section.

EPL MESSAG in T288 CMFLO2 is a short file, never more than a few lines long, which may be printed as part of each EPL run. It will contain a "message of the day", of immediate concern to all EPL users. A user with an EPL profile (as explained below) will only see EPL MESSAG when it has changed, since one of the items in the profile is the creation date of the last EPL MESSAG printed for the user.

EPL STUFF, also in T288 CMFLO2, is a log file of EPL status. In general, a description will be added at the beginning of EPL STUFF whenever there is a compiler change. Users maintaining a profile will be informed by EPL each time EPL STUFF is updated. The first few lines of EPL STUFF will always contain, in tabular form, a summary of the version number and level number for each pass in existence. (See the section below on "Compiler Versions", for a discussion of these terms.)

EPL BUG in T288 CMFLO2 contains a list of bugs and restrictions most recently encountered. The file contains only current problems and is usually quite short.

Two files of less concern to the average user are EPL1 LOG and EPL2 LOG, both in T288 CMFL02. These files contain a very brief record of what has changed in each new version of a pass of the compiler. These logs are updated by the compiler maintenance staff and are written in a rather concise form. EPL STUFF will usually contain the same information in more readable language.

### Compiler Versions

Because of both the relatively rapid changes in EPL and the desire on the part of its users for as much stability as possible, there are frequently several versions of each pass of the compiler in existence at any time. The three passes of the "official" version are always EPL1 SAVED, EPL15 SAVED, and EPL2 SAVED. A pass called NEPL<sub>i</sub>, (where i is "1", "15" or "2") will usually contain the next version of the compiler, which is to be installed as EPL<sub>i</sub> as soon as its reliability has been proved. Further, there will often be an XEPL<sub>i</sub> -- an experimental version which is not yet reliable enough to become a NEPL<sub>i</sub>. In general, users should refrain from using an XEPL<sub>i</sub> unless they are quite sure that they know what features (and problems) it has. After the installation of a NEPL<sub>i</sub> as an EPL<sub>i</sub>, the previous EPL<sub>i</sub> will be available for a few days as OEPL<sub>i</sub>, in case of mishap. Finally, there may on occasion be special experimental versions identified by any letter at all.

Because of this proliferation of compiler versions, the EPL command provides options so that the user may select which version of each pass that he wants. For example, to use EPL1 OEPL15 and XEPL2, he would include in his call to EPL the arguments

USE1            USEX2            USE015

To use NEPL1, NEPL15 and NEPL2,

USEN3            USEN15

would work. ("3" in this context means both "1" and "2".)

Since the compiler is still in a final stage of debugging, it is helpful to the compiler maintainers to be able to associate with the output from each run the identities of the versions of the compiler which were used. Thus every version of each pass is assigned two identifying numbers: a version number and a level number. Each time

a pass of the compiler is recompiled, it is assigned the next sequential version number for that pass, and level zero. If an octal patch is made, the level number is increased by one. Thus the version and level numbers identify uniquely a pass, regardless of what name it has or what is its file system creation date.

EPL takes several steps to help identify compiler versions. The console output from an EPL run includes the names of the versions selected, along with their creation date and author. Also, each pass prints at the beginning of its operation two lines: the first line of its input file, and the version number and level number. Finally, the version number and level number are included as the first line of each output file produced. One effect of this is that each EPLBSA file created by EPL contains, in its first three lines, comments identifying the compiler which created it.

### The "EPL Profile"

As discussed in detail in the next two sections, the user has a wide choice of options available in his use of EPL. Since a particular user may want to use the same set of options repeatedly, he may find it useful to have EPL maintain for him an EPL profile which contains a record of options selected. He may still supply option arguments for a run, and these will modify those selected in the profile, rather than the default options. The argument UPDATE may be used to indicate that the profile is to be updated to contain the entire set of options in force for the current run. FRESH may be used to start over with all default options, and STATUS may be used to get a listing of the options current. These arguments are explained in more detail below.

The user with a profile obtains two other advantages. Normally, the file EPL MESSAG is printed as part of each EPL run. The user with a profile will see this message only when it has changed, since one of the items in the profile is the CTSS file system creation date of the last EPL MESSAG printed. Also in the profile is the creation date of EPL STUFF, and a message is printed whenever this date changes. (The file itself is not printed by EPL -- it is up to the user to do that.)

The EPL profile is actually part of a file called USER PROFIL. This file is used by the CTSS command ".", as described in Section AJ.2.01 of the CTSS Programmer's Guide. Users who wish to have an EPL profile must first use "." to create a USER PROFIL. The technique for doing this is described below, under "Making the Links".

The items in the profile are now listed. Note particularly that there are no items corresponding to ONOFF, MESSAG, STATUS, FRESH, UPDATE, etc., which only apply on a per-invocation basis.

1. File system creation dates for EPL STUFF and EPL MESSAG.
2. Listing control -- a switch indicating whether or not RQASCI should be invoked as part of the run.
3. Five switches for permanent EPL, EPLMAC, SOURCE, MACROS and EPLBSA.
4. Five switches for RQASCI control of EPL, EPLMAC, SOURCE, MACROS and EPLBSA.
5. Three words, containing the names of pass 1, pass 1.5 and pass 2 to be used.
6. A BRIEF switch controlling the verbosity of the runcom's printing.
7. Several other miscellaneous switches.

### Overview of Options

The user of EPL has available to him a wide choice of options in the use of the compiler. He may select options for a particular run by including option parameters in his call for EPL, or he may maintain in his file directory a profile of options to be used for all compilations. If he makes the latter choice, he may on a particular run override some or all of the options in his profile. Further, the user may update his permanent options as he chooses.

There are six categories of options available:

- . There are compiler version selectors. Options are available so that the user may select which version of each pass that he wants. If the user selects a NEPLi or OEPLi which is not present in his directory, the corresponding EPLi will be used instead.
- . There is listing control. EPL will, if desired, call RQASCI to request the printing of files relevant to the run. There are switches settable by options to indicate which files are to be printed.

- . There is mode control. EPL will usually leave certain created files in temporary mode, but the user may specify permanent mode for any or all of them.
- . There is a BRIEF switch. Normally the EPL command is fairly verbose in its online printing. The user may stop this verbosity by selecting BRIEF.
- . There are miscellaneous options. Either Pass 1 and Pass 1.5, or Pass 2 may be omitted from a run, the result of previous EPLMAC may be used as input to Pass 1, etc.
- . There is option maintenance. The user maintaining a permanent profile of options has the ability to control its updating or to override it.

In general, setting an option involves setting a switch ON or OFF. The option name used by the user as an argument to EPL designates which switch to set, and the setting the switch is to receive is the current value of the switch ONOFF. This latter switch is controlled by the arguments "ON" and "OFF".

#### Details about the Options

The options will now be described in detail. Reference will frequently be made to the internal switch ONOFF, whose value is used to set option switches. ONOFF is initially ON, but may be set OFF by the "OFF" option and ON by the "ON" option. Its "current" value is used, as the arguments are processed from left to right.

The following abbreviations are used in this description:

- x Any letter or digit.
- d Either "1", "2", "3" or "15". d indicates a pass of the compiler, and a d of three indicates both pass 1 and pass 2. "15" indicates pass 1.5.
- nam Any of EPL, EPM, SRC, MAC or EPB. These stand respectively for EPL, EPLMAC, SOURCE, MACROS and EPLBSA.

- USExd Use compiler version xEPLd.
- USEd Use compiler version EPLd. Default.
- RQASCI Call RQASCI to print some or all of NAME EPL, NAME EPLMAC, NAME SOURCE, NAME MACROS and NAME EPLBSA. The options RQEPL, RQEPM, RQSRC, ROMAC and RQEPB control which files are to be printed. The default files to print if this option is used are SOURCE, MACROS and EPLBSA.
- NOLIST Do not call RQASCI. Default.
- MESSAG Print EPL MESSAG as part of this run.
- STATUS After reading all the arguments, print the entire set of options which will be used for this run. If UPDATE is used, these are the options to be put into the profile.
- UPDATE Update the EPL profile so that it contains the entire set of options which will be used for this run.
- FRESH Set all options to default values. The argument list is then rescanned, from the beginning.
- ON Set the switch ONOFF to ON. The switch is initialized to ON.
- OFF Set ONOFF to OFF.
- BRIEF Set to ONOFF a switch controlling the verbosity of on-line output from the EPL Runcom. If BRIEF is ON, only error messages and compiler versions will be printed. Default is OFF indicating verbose output.
- PASS1 Set the execution switch for Pass 1 and Pass 1.5 to ONOFF. Passes 1 and 1.5 are included as part of the run only if this switch is ON. Default is ON.
- PASS2 Set execution switch for Pass 2 to ONOFF. Pass 2 is included as part of the run only if this switch is ON. Default is ON.

- RQnam Set to ONOFF a switch controlling whether or not file "NAME nam" is to be included in any call to RQASCI. Default is ON for SOURCE, MACROS and EPLBSA; and OFF for EPL and EPLMAC.
- PRMnam Set to ONOFF a switch controlling whether or not file "NAME nam" is to be left in permanent mode at the end of the run. If the switch for a particular file is OFF, its mode will be changed to temporary at the end of the run. Default is ON for EPL and EPLBSA; and OFF for EPLMAC, SOURCE and MACROS. Note that setting this switch OFF for EPL or EPLBSA is almost surely an error, resulting in the possible loss of the user's input file or the useful output created.
- KEPEPM Set to ONOFF a switch indicating whether or not NAME EPLMAC is to be created in permanent mode. Default is OFF, indicating that this file is to be created by EPLMAC in temporary mode and thus will be deleted by CTSS when it has been read by pass 1.
- EPLMAC Set to ONOFF a switch indicating that input to pass 1 is to be NAME EPLMAC instead of NAME EPL. Default is OFF.

#### Further Comments about Options

If conflicting options are used, the last one read will govern. In general, few checks are made for consistency of options. For example, the sequence

```
... ON RQPEM OFF KEPEPM
```

is useless, but EPL will not comment on it. As mentioned above, setting PRMEPL to OFF is quite dangerous, since it may result in the loss of the user's input file.

EPL does do two things for the user, though. If the file used as input to the compiler has any mode other than zero, EPL will make no attempt to change it, regardless of the user's request. Also, if NAME EPL is in temporary mode, EPL will not check to see if its first character is percent but will invoke EPLMAC automatically.

EPL will usually fail in an unpleasant way if it is unable to delete an existing file whose name it needs. For example, EPL will fail if the user has an existing NAME EPLBSA in his directory in protected mode, or if he has a link named NAME EPLBSA.

### Examples

To compile file THING EPL with options A and B set ON and with options C and D set OFF, the call

```
R EPL THING ON A B OFF C D
```

could be used. The ON is not needed, since ONOFF is set to ON initially.

To use NEPL1, EPL15 and OEPL2, calling RQASCI to print all five relevant files, making this the permanent options,

```
R EPL * USEN1 USE15 USE02 RQASCI ON RQEPL RQEPM UPDATE KEPEPM
```

might be used. The file name "\*" indicates that no compile is to be done. ON is again redundant, and the UPDATE could have appeared anywhere in the list. Indeed, all possible reorderings of the option list would have the same effect.

Having set the profile this way, THING can be compiled with these options by

```
R EPL THING
```

To compile ZORCH with standard options, but leaving the profile undisturbed,

```
R EPL ZORCH FRESH
```

might be used. Then to redo pass 2 with ZORCH MACROS just produced but with NEPL2,

```
R EPL ZORCH OFF PASS1 USEN2
```

might be used. The first compilation would have left ZORCH MACROS in temporary mode, so

```
R EPL * ON PRMMAC UPDATE
```

would adjust the profile so that subsequent runs would leave the MACROS file in permanent mode.

### Method

The program EPL SAVED actually initiates a sort of pre-programmed RUNCOM. The EPL compiler does its work in three passes, each pass being a complete core load. Further, each pass produces as output the files NAME BCD and NAME BSA, the first file being a listing file and the second being the useful output. In the simplest case, EPL compilations could be done with a RUNCOM such as the following:

```
CHAIN NAME
R EPL1 NAME EPL
RENAME NAME BCD NAME SOURCE
RENAME NAME BSA NAME BSX
R EPL15 NAME BSX
R JJJ NAME
DELETE NAME BSX NAME BSA NAME BCD
R EPL2 NAME MACROS
RENAME NAME BSA NAME EPLBSA
DELETE NAME BCD
```

(JJJ may be assumed to be a program which adjoins the files NAME BSA and NAME BSX to create the file NAME MACROS. No such program actually exists.)

Actual compilations are a great deal more complex for many reasons, some of which are these:

- . EPLMAC may be wanted to process "include" macros.
- . A version of the compiler other than the current one may be wanted.
- . The SOURCE and MACROS files are (usually) to be left in temporary mode, so a CHMODE is needed.
- . A call to RQASCI to print relevant files may be wanted as part of the run.
- . The file EPL MESSAG is to be printed as part of each run.

For these reasons and others, and because of the inefficiencies inherent in the use of RUNCOM in CTSS, the effect of the RUNCOM is simulated by programming. The program EPL SAVED, which is the program the user calls, starts off by saving the contents of the command buffers, along with switches indicating the options selected for this run, in the file EPL (TEMP). The file EPL MESSAG is then printed, unless the user has an EPL profile; in which case it is printed only if it has been updated since the user's last run. EPL then sets up some or all of the following commands in the command buffers, depending on the options selected by the user:

```
R EPL1 name EPL
R EPLX (ALFA)
R EPL15 name BSX
R EPLX (BETA)
```

Here "name" is the first name of the file being compiled. If EPLMAC is to be part of the run, the first line is replaced by:

```
R EPLMAC name T
R EPL1 name EPLMAC
```

EPLMAC is the program which processes the "include" macro. It leaves its output in NAME EPLMAC. (For details of EPLMAC, see BE.1.02.)

EPL1 is Pass 1 of the EPL compiler, and it will produce NAME BCD and NAME BSA. The first is a listing file and the second is the Pass 1 output.

EPLX, when called with parameter (ALFA), renames NAME BCD to NAME SOURCE and NAME BSA to NAME BSX. It then continues the chain.

EPL15 is pass 1.5 of the compiler, also producing NAME BCD (which is useless) and NAME BSA, its useful output.

EPLX, when called with parameter (BETA), combines NAME BSA and NAME BSX (in the ASCII sense) to produce NAME MACROS. It then loads the command buffers with

```
R EPL2 NAME MACROS
R EPLX (GAMA)
```

and continues the chain.

EPL2 is pass 2 of the compiler, producing (useless) NAME BCD and useful output NAME BSA.

EPLX, when called with parameter (GAMA), renames NAME BSA to NAME EPLBSA and deletes NAME BCD. It then performs tasks requested by the user, such as changing the mode of designated files to temporary, etc. It reloads the command buffers from EPL (TEMP), which it then deletes. It terminates by calling RQASCII (if requested) and continuing whatever chain was in effect when EPL was first invoked.

### Creating the Links

The user must link to EPL SAVED in T234 CMFLO1, and to the version(s) of the compiler he wants in T288 CMFLO2. (In general, he will want only EPL1 SAVED, EPL15 SAVED and EPL2 SAVED.) The first time EPL is run, it will make links to EPL SAVED and to EPLMAC SAVED in T234 CMFLO1 and to EPL MESSAG and EPL STUFF IN T288 CMFLO2.

To create a USER PROFIL, the command "." must be used. (Users of "." will already have a USER PROFIL.) All the user need do is type the single line ". D /".

Summary of Relevant Files

Following is a list of the files needed by the user to use EPL. The directory in which the file may be found is given, along with a brief description.

## On-Line Documentation

EPL MESSAG	T288 CMFLO2	The EPL "message of the day", printed as part of each EPL run.
EPL STUFF	T288 CMFLO2	Details about new passes.
EPL1 LOG	T288 CMFLO2	Concise log of pass 1 changes.
EPL2 LOG	T288 CMFLO2	Concise log of pass 2 changes.
EPL BUG	T288 CFFLO2	Current "bug" list.

## The EPL Command

EPL SAVED	T234 CMFLO1	The first part of the EPL command; the only part called by the user.
EPLX SAVED	T234 CMFLO1	The remaining parts of the EPL command.
EPLMAC SAVED	T234 CMFLO1	The EPLMAC command, to process "include" macros.

## The Compiler Itself

EPL1 SAVED	T288 CMFLO2	Pass 1, the current version.
EPL15 SAVED	T288 CMFLO2	Pass 1.5, the current version.
EPL2 SAVED	T288 CMFLO2	Pass 2, the current version.

Versions of the compiler other than the current version will usually also be in T288 CMFLO2.