



MCUEZPE05/D
JANUARY 1998

Freescale Semiconductor, Inc.

P&E ASSEMBLER CONVERTER USER'S MANUAL

© Copyright 1997 Motorola and HIWARE AG; All Rights Reserved

**For More Information On This Product,
Go to: www.freescale.com**

Important Notice to Users

While every effort has been made to ensure the accuracy of all information in this document, Motorola assumes no liability to any party for any loss or damage caused by errors or omissions or by statements of any kind in this document, its updates, supplements, or special editions, whether such errors are omissions or statements resulting from negligence, accident, or any other cause. Motorola further assumes no liability arising out of the application or use of any information, product, or system described herein; nor any liability for incidental or consequential damages arising from the use of this document. Motorola disclaims all warranties regarding the information contained herein, whether expressed, implied, or statutory, *including implied warranties of merchantability or fitness for a particular purpose*. Motorola makes no representation that the interconnection of products in the manner described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting or license to make, use or sell equipment constructed in accordance with this description.

Information contained in this document applies to REVersion (0) MCUEz.

The computer program contains material copyrighted by Motorola Inc., first published 1997, and may be used only under a license such as the License For Computer Programs (Article 14) contained in Motorola's Terms and Conditions of Sale, Rev. 1/79.

Trademarks

This document includes these trademarks:

MCUEz is a trademark of Motorola Inc.
EXORciser is a trademark of Motorola Inc.

The MCUEz development, emulation, and debugging application is based on HI-WAVE; a software technology developed by HIWARE. HI-WAVE is a registered trademark of HIWARE AG.

AIX, IBM, and PowerPC are trademarks of International Business Machines Corporation.
SPARC is a trademark of SPARC international, Inc.
Sun and SunOS are trademarks of Sun Microsystems, Inc.
UNIX is a trademark of Novell, Inc., in the United States and other countries, licensed exclusively through X/Open Company, Ltd.
X Window System is a trademark of Massachusetts Institute of Technology.





CONTENTS
CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION	1-1
1.2 USER INTERFACE	1-1
1.3 INPUT FILES	1-1
1.4 OUTPUT FILES	1-1

CHAPTER 2 USING THE ASSEMBLER CONVERTER

2.1 INTRODUCTION	2-1
2.2 CONVERTER ENVIRONMENT	2-1
2.3 PROJECT DIRECTORY	2-1
2.4 CONVERTING A P&E ASSEMBLER SOURCE FILE	2-1
2.5 SPECIFYING THE OUTPUT FILE NAME	2-2
2.6 INPUT FILE WITH AN ASM EXTENSION	2-3

CHAPTER 3 ENVIRONMENT

3.1 INTRODUCTION	3-1
3.2 ENVIRONMENT VARIABLE DETAILS	3-1
3.2.1 DEFAULTDIR	3-1
3.2.2 ENVIRONMENT	3-2
3.2.3 GENPATH	3-2
3.3 DEFINING ENVIRONMENT VARIABLES	3-2
3.3.1 Methods Of Defining Environment Parameters	3-3
3.3.2 Syntax For Parameter Definition	3-3
3.3.3 Specifying Paths In Parameter Definition	3-3
3.3.4 Line Continuation In Parameter Definition	3-4

CHAPTER 4 COMMAND-LINE OPTIONS

4.1 INTRODUCTION	4-1
4.2 COMMAND-LINE OPTION DETAILS	4-1
4.2.1 -H	4-1
4.2.2 -V	4-2
4.2.3 -O	4-2

CHAPTER 5 AMBIGUITIES IN P&E ASSEMBLER SYNTAX

5.1 INTRODUCTION	5-1
5.2 DASHES IN LABELS.....	5-1
5.3 PERCENT CHARACTERS IN MACROS.....	5-1
5.4 LABELS IN MACROS.....	5-2

CHAPTER 6 MODIFICATIONS DURING CONVERSION

6.1 INTRODUCTION	6-1
6.2 CONSTANTS	6-1
6.2.1 Q Suffix	6-1
6.2.2 T Suffix	6-1
6.3 DASHES IN LABELS.....	6-1
6.4 DEFAULT BASE	6-3
6.5 DIRECTIVES IN THE FIRST COLUMN	6-3
6.6 EXPRESSIONS	6-3
6.7 LABELS IN MACROS.....	6-3
6.8 MACRO DECLARATION	6-4
6.9 MACRO PARAMETERS	6-4
6.10 UNSUPPORTED COMMANDS	6-5
6.11 OPERATORS	6-5
6.12 SUPPORTED COMMANDS	6-6

FIGURES

Figure 2-1. XLATE Icon	2-1
Figure 2-2. Converter Window	2-2
Figure 2-3. Conversion Completed	2-2
Figure 2-4. Specifying Output File	2-2
Figure 2-5. File With “.ASM” Extension	2-3
Figure 2-6. Renamed Output File	2-3
Figure 4-1. Command-Line Options	4-1
Figure 4-2. Version Number And Project Directory	4-2
Figure 4-3. Output File Definition.....	4-2

TABLES

Table 6-1. Supported Commands.....	6-6
------------------------------------	-----



CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

This manual describes the P&E to MCUEz Assembler Converter. The Converter is designed to convert P&E Microcomputer Systems assembler source files to MCUEz assembler files.

1.2 USER INTERFACE

The P&E to MCUEz Assembler Converter offers a command line interface and an interactive interface. If you do not enter an argument on the command line, the interactive interface opens and prompts you for the information.

1.3 INPUT FILES

The Converter requires a valid P&E assembler input file, but does not require a specific file extension. The default input file extension is “.S”.

The conversion process does not alter the input file in any way. The P&E assembler syntax in the input file is converted to MCUEz assembler syntax in a separate output file. The specific syntax conversions are described in the Environment chapter of this manual.

1.4 OUTPUT FILES

When a valid input file is successfully converted, the Converter generates an output file in MCUEz assembler syntax and saves the output file to the directory in which the input file was located. The output file has the same name as the input file, but the extension for the output file is “.ASM”.

CHAPTER 2

USING THE ASSEMBLER CONVERTER

2.1 INTRODUCTION

This chapter explains how to operate and access the P&E to MCUEz Assembler Converter.

2.2 CONVERTER ENVIRONMENT

The Converter must be associated with a project directory to execute correctly.

A project directory is a directory that contains all the files needed to configure your development environment.

During the MCUEz installation process, the project directory is automatically set to “C:\MCUEZ\DEMO\WMMDS05A”. This default project directory contains initialization files that must be present in the MCUEz environment.

To define a new project directory or to switch between projects, refer to the *MCUEz User's Manual*.

2.3 CONVERTING A P&E ASSEMBLER SOURCE FILE

When the project directory has been configured, you can begin to convert your application.

The following steps show how to convert an example file to MCUEz assembler syntax. In the example, the file is named “test.s” and is located in the project directory.

1. Activate the MCUEz Shell.
2. Select the “XLATE” icon from the MCUEz Shell Menu:

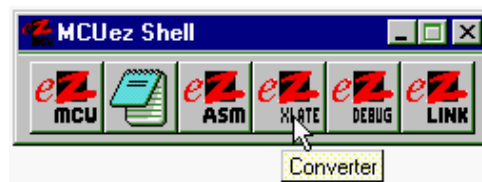


Figure 2-1. XLATE Icon

3. The P&E to MCUEz Assembler Converter window opens:

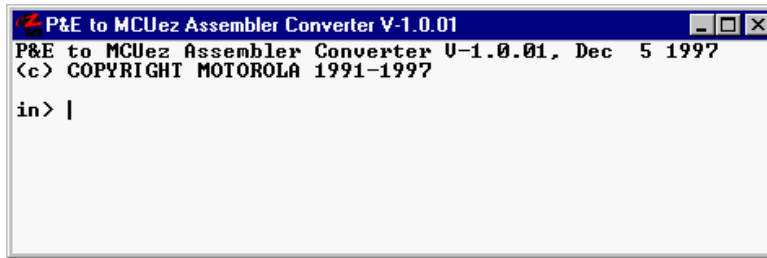


Figure 2-2. Converter Window

4. Type the source file name and press the Enter key to start the conversion process.
5. When the output file location appears, the conversion has been successfully completed:

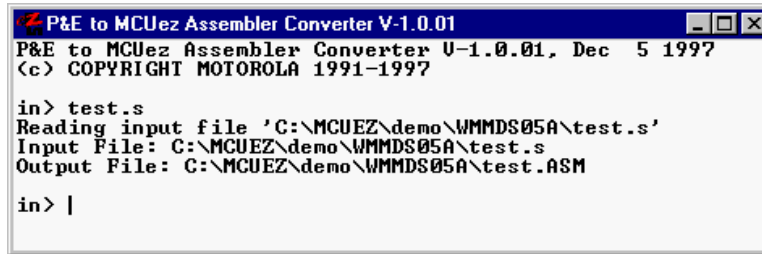


Figure 2-3. Conversion Completed

2.4 SPECIFYING THE OUTPUT FILE NAME

Use the code, “-O=”, to specify the name of the output file. At the “in>” prompt, type the name of the input file, followed by a space. Type “-O=”, followed by the name of the output file. Press Enter to start the conversion process. The Converter creates the specified output file in the project directory:

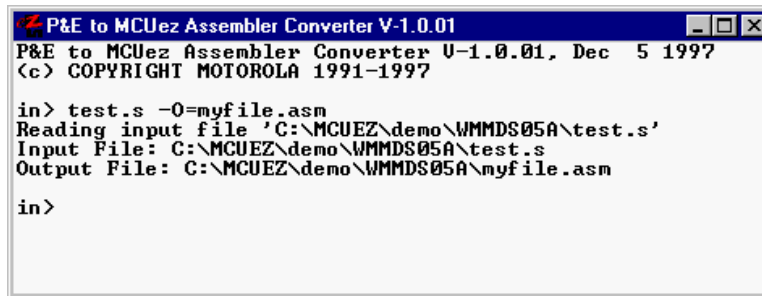
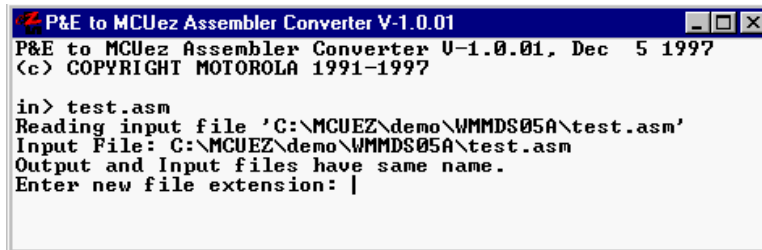


Figure 2-4. Specifying Output File

In the example figure, the input file is “test.s” and the output file is “myfile.asm”.

2.5 INPUT FILE WITH AN ASM EXTENSION

If the input file has an “.ASM” extension, the Converter requires you to enter another extension for the output file.



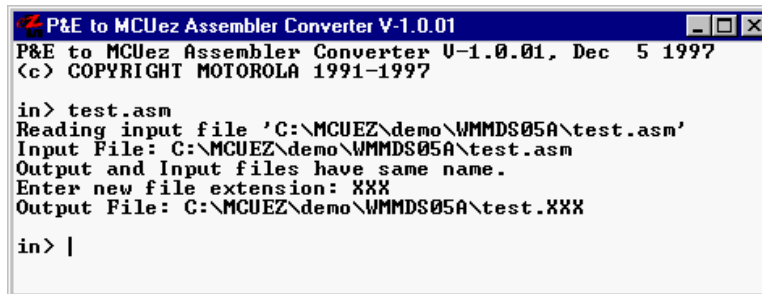
```
P&E to MCUEz Assembler Converter V-1.0.01
P&E to MCUEz Assembler Converter U-1.0.01, Dec 5 1997
(c) COPYRIGHT MOTOROLA 1991-1997

in> test.asm
Reading input file 'C:\MCUEZ\demo\WMMDS05A\test.asm'
Input File: C:\MCUEZ\demo\WMMDS05A\test.asm
Output and Input files have same name.
Enter new file extension: |
```

Figure 2-5. File With “.ASM” Extension

At the “Enter new file extension:” prompt, type the new extension for the output file.

The new output file extension is reflected in the output file name:



```
P&E to MCUEz Assembler Converter V-1.0.01
P&E to MCUEz Assembler Converter U-1.0.01, Dec 5 1997
(c) COPYRIGHT MOTOROLA 1991-1997

in> test.asm
Reading input file 'C:\MCUEZ\demo\WMMDS05A\test.asm'
Input File: C:\MCUEZ\demo\WMMDS05A\test.asm
Output and Input files have same name.
Enter new file extension: XXX
Output File: C:\MCUEZ\demo\WMMDS05A\test.XXX

in> |
```

Figure 2-6. Renamed Output File





CHAPTER 3

ENVIRONMENT

3.1 INTRODUCTION

This chapter describes how to define the environment variables that control the P&E to MCUez Assembler Converter.

3.2 ENVIRONMENT VARIABLE DETAILS

This section describes the environment variable available to the Converter:

- “DEFAULTDIR”
- “ENVIRONMENT”
- “GENPATH”

3.2.1 DEFAULTDIR

Description: Specifies the default directory for the MCUez Assembler, Linker, and Debugger. The directory specified with “DEFAULTDIR” replaces the MCUez directory defined by the operating system or launching tool.

Syntax: “DEFAULTDIR=” <directory>.

Arguments: <directory>: Directory to be used as the default directory.

Example: DEFAULTDIR=C:\INSTALL\PROJECT

NOTE

“DEFAULTDIR” is a system-level environment variable (global environment variable) and cannot be specified in a default environment file such as “DEFAULT.ENV”.

3.2.2 ENVIRONMENT

Description: Specifies the environment file for the MCUEz Assembler, Linker, and Debugger. The Converter normally searches for the environment file named “DEFAULT . ENV” in the current directory. “ENVIRONMENT” allows you to set a different file name for the environment file.

Syntax: “ENVIRONMENT=” <file>.

Arguments: <file>: file name with path specification, without spaces

Example: ENVIRONMENT=\MCUEZ\prog\global . env

NOTE

“ENVIRONMENT” is a system-level environment variable and cannot be specified in a default environment file such as “DEFAULT . ENV”.

3.2.3 GENPATH

Description: Defines paths to search for input files. The Converter searches for input files in the project directory, then in the directories indicated by “GENPATH”. If a directory name in the path list is preceded by an asterisk, “*”, the Converter searches that directory and its subdirectories:

Syntax: “GENPATH=” {<path>}.

Arguments: <path>: Paths separated by semicolons, without spaces.

Example: GENPATH=\obj;...\...\lib;

3.3 DEFINING ENVIRONMENT VARIABLES

Some of the Converter’s environment parameters can be altered through environment variables. This section explains the methods, syntax, paths, and line continuation symbols used to define the environment variable.

The most effective method of defining and manipulating environmental variables for the Converter is the “DEFAULT . ENV” file in the MCUEz project directory. The “DEFAULT . ENV” file allows you to store projects in separate directories and gives you the ability to store a unique environment with each project.

NOTE

If you use an external editor (WinEdit, Codewright), do *not* set the system environment variable “DEFAULTDIR”. If you set “DEFAULTDIR” and its project directory does not match the editor’s project directory, the files you save may be stored in an unexpected location.



3.3.1 Methods of Defining Environment Parameters

Three methods may be used to define environment parameters:

- Using system environment variables supported by your operating system.
- Putting the definitions in the project directory file named “DEFAULT.ENV” (environment variable entries in “DEFAULT.ENV” can be up to 1024 characters).
- Putting the definitions in a file determined by the value of the system environment variable “ENVIRONMENT”.

When the Converter searches for environment variables, it searches the system environment first, then the “DEFAULT.ENV” file, and, finally, the file determined by the “ENVIRONMENT” system variable.

If you choose not to define an environment variable, the Converter uses a default definition.

3.3.2 Syntax For Parameter Definition

The syntax used to define a variable is the name of the environment variable followed by an equals sign, “=”, and the parameter definition:

```
GENPATH=C:\INSTALL\LIB;D:\PROJECTS\TESTS;\usr\local\lib;
```

NOTE

Do not include a space character before or after the equals sign, “=”, in a variable definition.

3.3.3 Specifying Paths In Parameter Definition

Most environment variables include a path list made up of directory names separated by semicolons. The path lists determine where the Converter begins to search for files.

The syntax used to create a path list is the name of the path list followed by the equals sign, “=”, and the path definition:

```
GENPATH=C:\INSTALL\LIB;D:\PROJECT\TEST;\usr\loc\hicross\lib;\home\me
```

The Converter searches the directories in the order in which they appear in the path list.

If a directory name in the path list is preceded by an asterisk, “*”, the Converter searches that directory and its subdirectories:

```
LIBPATH=*C:\INSTALL\LIB
```

3.3.4 Line Continuation in Parameter Definition

The line continuation character “\” is used to define an environment variable over multiple lines in the “DEFAULT.ENV” file:

```
COMPOPTIONS=\
-W2 \
-Wpd
```

has the same meaning as:

```
COMPOPTIONS=-W2 -Wpd
```

but this feature may be dangerous using it together with paths, e.g.

```
GENPATH=.\
TEXTFILE=.\txt
```

will be interpreted as:

```
GENPATH=.TEXTFILE=.\txt
```

To avoid such problems, use a semicolon, “;”, after the line continuation character, “\”:

```
GENPATH=.\;
TEXTFILE=.\txt
```


CHAPTER 4

COMMAND-LINE OPTIONS

4.1 INTRODUCTION

The P&E to MCUEz Assembler Converter offers a number of command-line options that can be used to control the Converter's operations. These options are entered at the command line and consist of a dash, "-", followed by an argument. The argument is made up of from 1 to 128 letters/digits.

Command-line options are not case sensitive, e.g. "-o=test.xxx" produces the same action as "-O=TEST.XXX".

4.2 COMMAND-LINE OPTION DETAILS

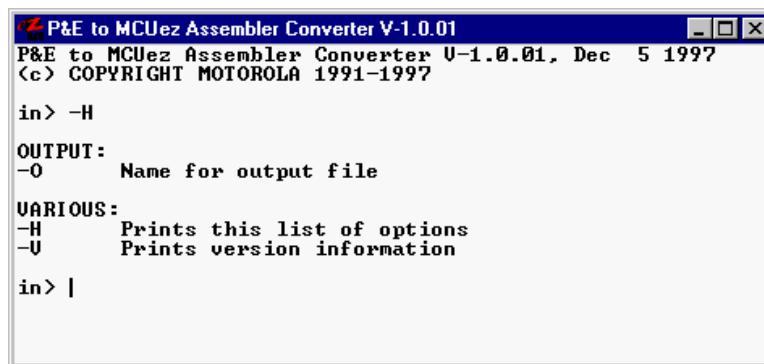
This section describes each command-line option available for the Converter.

4.2.1 -H

Syntax: "-H"

Arguments: none

Description: Displays a list of command-line options with a short description of each



```
P&E to MCUEz Assembler Converter V-1.0.01
P&E to MCUEz Assembler Converter V-1.0.01, Dec 5 1997
(c) COPYRIGHT MOTOROLA 1991-1997

in> -H
OUTPUT:
-O      Name for output file

VARIOUS:
-H      Prints this list of options
-U      Prints version information

in> |
```

Figure 4-1. Command-Line Options

COMMAND-LINE OPTIONS

4.2.2 -V

Syntax: “-V”

Arguments: none

Description: Displays the Converter version number and the location of the project directory.

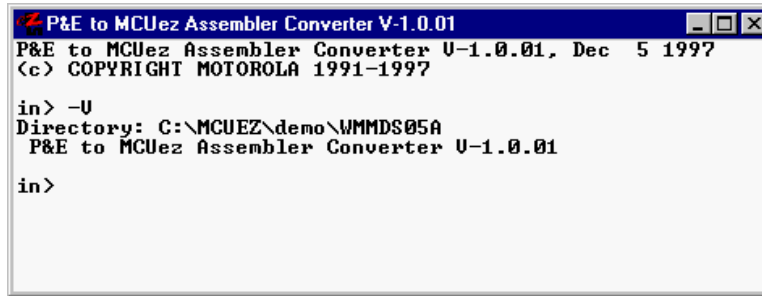


Figure 4-2. Version Number And Project Directory

4.2.3 -O

Syntax: “-O=” <file name>.

Arguments: <file name>: Name of the MCUez assembler source file which must be generated by the conversion.

Description: Defines the name of the output file.

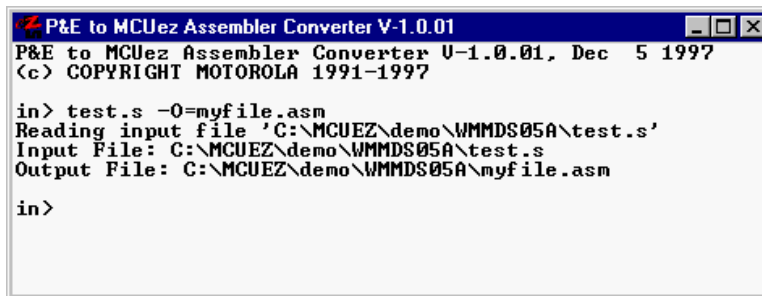


Figure 4-3. Output File Definition



CHAPTER 5

AMBIGUITIES IN P&E ASSEMBLER SYNTAX

5.1 INTRODUCTION

P&E Assembler Syntax contains three ambiguities that may affect the conversion process:

- Dashes in labels
- Percent characters in macros
- Labels in macros

5.2 DASHES IN LABELS

Problem:

P&E Assembler syntax allows the dash, “-”, character in label names. The presence of a the dash can lead to ambiguity in converting label names. For example, “lab1-lab2” may refer to a label called “lab1-lab2” or two labels, “lab1” and “lab2”.

Solution:

If an expression with a dash in it begins the first column of the input file, the Converter assumes the expression is a label. In the output file, the Converter substitutes the underscore character, “_”, for dashes used in labels and adds the comment, “; CONV_01: Dash detected in Label.”.

5.3 PERCENT CHARACTERS IN MACROS

Problem:

P&E Assembler syntax perceives that, in macro definitions, the percent character, “%”, refers to a macro parameter number. This creates ambiguity in the output file because, in MCUEz Assembler syntax, an expression such as “label%2” may refer to appending the parameter number to the string “label” or the remainder of the value of the variable “label” divided by two.

Solution:

The Converter replaces the percent character with the backslash, “\”, character and adds the comment “; CONV_02: Percent detected in Label.”.



5.4 LABELS IN MACROS

Problem:

In P&E Assembler syntax, a suffix is added to all the labels defined within a macro. During conversion, the Converter must determine whether to add a suffix to a label referenced in a macro definition.

Solution:

The Converter determines where the label is defined. If the label is defined inside the macro, the prefix “\@” is added to the label. This generates unique labels when the macro is expanded several times in a source file. The Converter also adds the comment “; CONV_03: Label detected in Macro.”.



CHAPTER 6

MODIFICATIONS DURING CONVERSION

6.1 INTRODUCTION

This chapter explains the syntax changes the P&E to MCUEz Assembler Converter makes during the conversion process.

6.2 CONSTANTS

P&E Assembler uses suffixes to indicate the type of number a constant represents. MCUEz Assembler uses different suffixes to represent the same information.

6.2.1 Q Suffix

The suffix “Q” indicates a binary integer in P&E syntax. The converter substitutes the MCUEz suffix, “B”, for the P&E suffix “Q”.

6.2.2 T Suffix

The suffix “T” indicates a decimal integer in P&E syntax. The converter substitutes the MCUEz suffix, “D”, for the P&E suffix “T”.

6.3 DASHES IN LABELS

P&E Assembler syntax allows the dash, “-”, character in label names. The presence of a the dash can lead to ambiguity in converting label names. For example, “lab1-lab2” may refer to a label called “lab1-lab2” or two labels, “lab1” and “lab2”.

If an expression with a dash in it begins the first column of the input file, the Converter assumes the expression is a label. In the output file, the Converter substitutes the underscore character, “_”, for dashes used in labels and adds the comment, “; CONV_01: Dash detected in Label.”.

In the following example, the labels “lab-lab2” and “label2-3” need to be converted:

P&E input file:

```
$macro macro1
lab-lab2
  lda lab-lab2
  lda label2-3
  lda label2
$macroend

                ORG ROM
label1:
  lda label2-label3
  lda label2-3
label2:
  nop
label3:
  nop
label2-3:
  nop
```

MCUez output file:

```
macro1: MACRO
\@lab_lab2 ; CONV_01: Dash detected in Label. ; CONV_03: Label detected in
Macro.
  lda \@lab_lab2 ; CONV_01: Dash detected in Label. ; CONV_03:Label
detected in Macro.
  lda label2_3 ; CONV_01: Dash detected in Label.
  lda label2
ENDM

                ORG ROM
label1:
  lda label2-label3
  lda label2_3 ; CONV_01: Dash detected in Label.
label2:
  nop
label3:
  nop
label2_3: ; CONV_01: Dash detected in Label.
  nop
```

Note that the dash in the line “lda label2-label3” does not change because the expression does not begin in the first column. The dash in the following line changes because the expression “label2-3” was defined in the previous lines.

6.4 DEFAULT BASE

The P&E Assembler use base 16 as its default base number. The Converter uses base 10 and inserts the directive “BASE \$10” at the beginning of each output file.

6.5 DIRECTIVES IN THE FIRST COLUMN

In P&E Assembler syntax, directives start with a symbol; “/”, “#”, “\$”, or “.”; in the first column. In MCUEz Assembler syntax, only labels can begin in the first column.

To distinguish directives from labels, the Converter replaces the symbol in front of each directive with a space character.

6.6 EXPRESSIONS

Where the P&E Assembler uses brackets, “{” and “}”, the MCUEz Assembler uses parentheses, “(” and “)”. The Converter simply replaces brackets with parentheses.

6.7 LABELS IN MACROS

In P&E Assembler syntax, a suffix is added to all the labels defined within a macro. During conversion, the Converter must determine if a suffix should be added to a label referenced in a macro definition.

The Converter determines where the label is defined. If the label is defined inside the macro, the prefix “\@” is added to the label. This generates unique labels when the macro is expanded several times in a source file. The Converter also adds the comment “; CONV_03: Label detected in Macro.”.

In the following P&E Assembler code, the label “loop” is defined inside the macro:

```
#macro mymacro
    nop
loop:
    nop
    bra loop
#macroend
```

The Converter converts this macro declaration to:

```
mymacro: MACRO
    nop
\@loop: ; CONV_03: Label detected in Macro.
    nop
    bra \@loop ; CONV_03: Label detected in Macro.
ENDM
```

6.8 MACRO DECLARATION

To define a macro with P&E Assembler syntax, you must follow the format:

```
#macro mymacro
    nop
    nop
    nop
#macroend
```

The Converter replaces the P&E macro definition directives with the appropriate MCUEz Assembler directives:

```
mymacro: MACRO
    nop
    nop
    nop
ENDM
```

P&E Assembler syntax also allows you to specify parameters on the macro definition line:

```
#macro mymacro par1 par2 ; my comment
    nop
    nop
    nop
#macroend
```

The Converter carries the parameters over to the converted code and places them in the appropriate position:

```
mymacro: MACRO ; par1 par2 ; my comment
    nop
    nop
    nop
ENDM
```

6.9 MACRO PARAMETERS

P&E Assembler syntax perceives that, in macro definitions, the percent character, “%”, refers to a macro parameter number. This creates ambiguity in the output file because, in MCUEz Assembler syntax, an expression such as “label%2” may refer to appending the string “label” with the parameter number or to the remainder of the value of the variable “label” divided by two.

The Converter replaces the percent character with the backslash, “\”, character and adds the comment “; CONV_02: Percent detected in Label.”.



Therefore, the P&E Assembler code:

```
#macro mymacro par1 par2 ; my comment
  lda #%1
  nop
  ldx %2
#macroend
```

converts to the MCUez code:

```
mymacro: MACRO ;par1 par2 ; my comment
  lda #\1 ; CONV_02: Percent detected in Macro.
  nop
  ldx \2 ; CONV_02: Percent detected in Macro.
ENDM
```

6.10 UNSUPPORTED COMMANDS

The P&E Assembler commands “cycle_adder_on”, “cycle_adder_off” and “subheader” are not supported by the MCUez Assembler.

The Converter replaces the symbols in front of these commands; “/”, “#”, “\$”, or “.”; with a semi-colon, “;”. The semi-colon in the first column converts the commands into comments.

6.11 OPERATORS

The P&E Assembler uses “<” as the shift left operator while the MCUez Assembler uses “<<”. Similarly, the shift right operator in P&E syntax is “>” while it is “>>” in MCUez syntax.

The Converter replaces “<” with “<<”, and “>” with “>>”.

6.12 SUPPORTED COMMANDS

MCUez Assembler supports many P&E Assembler commands under different syntax. The Converter replaces these P&E commands with the matching MCUez commands:

P&E Assembler Syntax	MCUez Assembler Syntax
DB	DC . B
DW	DC . W
EJECT	PAGE
ELSEIF	ELSE
HEADER	TITLE
IF	IFEQ
IFNOT	IFNE
PAGELength	PLEN
PAGEWIDTH	PLEN
SET <label>	<label>: SET 1
SETNOT <label>	<label>: SET 0

Table 6-1. Supported Commands