



# Microchip MIB Compiler

MPLAB Harmony Integrated Software Framework

## Volume VII: Utilities

---

This volume describes the utilities that are available in MPLAB Harmony.

### Description



MPLAB Harmony provides utilities to simplify the development process of middleware technologies, such as TCP/IP and Wi-Fi, as well as the Microchip Proprietary File System (MPFS2) and the Microchip Management Information Base (MIB) Compiler.

## Microchip MIB Compiler

This section provides a description of the Microchip Management Information Base (MIB) Compiler.

### Description

Microchip's SNMP Agent uses a custom script to describe the MIB. This script is designed to simplify the MIB definition and its integration with the main application. The actual MIB used by the SNMP Agent is a binary image created by the Microchip Management Information Base (MIB) to Binary Information Base (BIB) compiler named `mib2bib`.

### Microchip MIB Script Commands

A Microchip MIB file is an ASCII text file consisting of multiple command lines. Each command line consists of a single command, starting with the dollar sign character (\$), and one or more command parameters delimited with commas and enclosed in parentheses. Lines that do not start with a dollar sign are interpreted as comments and are not processed by the compiler. Commands must be written in a single line; they cannot span multiple lines.

The MIB script language includes a total of five commands, each having a specific syntax. Only one command, `DeclareVar`, is mandatory; the others are optional depending on the application and the types of information to be defined. In practice, at least one other command will be used in defining an MIB.



#### Notes:

1. Both the `ASN.1` MIB file and the Microchip MIB script use the same `.mib` file extension; however, the files have distinctly different purposes. The `ASN.1` MIB file is used by the MIB browser (NMS) to properly display context for your application.
2. The Microchip MIB script is compiled using `mib2bib` to create a BIB file. The BIB file is later converted using `MPFS2` to store the MIB data for your application in internal Flash or EEPROM.

### MIB Compiler

The `mib2bib` compiler converts the Microchip MIB script into a binary format compatible with the Microchip SNMP Agent. It accepts a Microchip MIB script in ASCII format and generates two output files: the binary information file, `snmp.bib`, and the C header file, `mib.h`. The binary file can be included in a Microchip File System (MPFS2) image.

The complete command line syntax for `mib2bib` is:

```
# java -jar mib2bib.jar [/?] [/h] [/q] <MIBFile> [/b=<OutputBIBDir>] [/I=<OutputIncDir>]
```

where:

- `/?` - Displays command line help
- `/h` - Displays detail help for all script commands
- `/q` - Overwrites existing `snmp.bib` and `mib.h` files
- `<MIBFile>` - is the input MIB script file
- `<OutputBIBDir>` - is the output BIB directory where `snmp.bib` should be copied. If a directory is not specified, the current directory will be used
- `<OutputIncDir>` - is the output Inc directory where `mib.h` should be copied. If a directory is not specified, the current directory will be used

For example, the command, `Java -jar mib2bib.jar MySNMP.mib`, compiles the script, `MySNMP.mib`, and generates the `snmp.bib` and `mib.h` output files in the same directory.

Conversely, the command, `mib2bib /q MySNMP.mib /b=WebPages`, compiles the `MySNMP.mib` script file and overwrites the existing output files. It also specifies that the `snmp.mib` file is located in the subdirectory, `WebPages`. Because it is not specified, `mib.h` is assumed to be in the current directory.

If compilation is successful, `mib2bib` displays the statistics on the binary file, including the number of OIDs and the Agent ID, and the output file size.

The MIB compiler is a simple rule script compiler. While it can detect and report many types of parsing errors, it has these known limitations:

- All command lines must be written in single line
- All command parameters must immediately end with either a comma or a right parenthesis. For example, `$DeclareVar(myOID, ASCII_STRING ...)`, will fail because the `ASCII_STRING` keyword is not immediately followed by a comma.
- All numerical data must be written in decimal

#### Example: Typical Output Display for a `mib2bib` Compilation

```
C:\microchip\harmony\v0_70b\utilities\mib2bib>java -jar mib2bib.jar snmp.mib
mib2bib v1.0.1 (Oct 14 2003)
Copyright (c) 2003 Microchip Technology Inc.
Input MIB File : C:\microchip\harmony\v0_70b\utilities\mib2bib\snmp.mib
Output BIB File: C:\microchip\harmony\v0_70b\utilities\mib2bib\snmp.bib
Output Inc File: C:\microchip\harmony\v0_70b\utilities\mib2bib\mib.h
BIB File Statistics:
    Total Static OIDs           : 9
    Total Static data bytes     : 57
    Total Dynamic OIDs         : 10
    (mib.h entries)
```

```

Total Read-Only OIDs      : 3
Total Read-Write OIDs    : 7
-----
Total OIDs                : 19
Total Sequence OIDs      : 4
Total AgentIDs           : 1
=====
Total MIB bytes          : 224
(snmp.bib size)

```



**Note:** For additional details about the Microchip MIB script, please refer to the Microchip application note *AN870 - "SNMP V2c Agent for Microchip TCP/IP Stack"* (DS00870).

## mib2bib.jar Run-time Error Codes

This topic lists the run-time error codes for `mib2bib.jar`.

### Description

#### Run-time Error Codes

Error Code	Description	Reason
1000	Unexpected End-Of-File (EOF) found.	End-Of-File was reached before the end of command.
1001	Unexpected End-Of-Line (EOL) found.	End-Of-Line was reached before the end of command.
1002	Invalid escape sequence detected; only ' ', '\', '(', or ')' may follow a '\ '.	All occurrences of ' ', '(', and ')' must be preceded by a '\ '.
1003	Unexpected empty command string received.	Command does not contain any parameter.
1004	Unexpected right parenthesis found.	Right parenthesis was found in place of a parameter.
1005	Invalid or empty command received.	Command does not contain sufficient parameters.
1006	Unexpected escape character received.	A '\ ' character was detected before or after parameters were expected.
1007	Unknown command received.	N/A
1008	Invalid parameters: expected <code>\$DeclareVar(oidName, dataType, oidType, accessType, oidType)</code> .	N/A
1009	Duplicate OID name found.	Specified OID name is already in use.
1010	Unknown data type received.	Data type keyword does not match one of the allowed keywords.
1011	Unknown OID type received.	OID type keyword does not match one of the allowed keywords.
1012	Empty OID string received.	N/A
1013	Invalid parameters: expected <code>\$DynamicVar(oidName, id)</code> .	N/A
1014	OID name is not defined.	N/A
1015	Invalid OID ID received – must be between 0-1023 inclusive.	N/A
1016	Invalid parameters: expected <code>\$SequenceVar(oidName, index)</code> .	N/A
1017	Invalid parameters: expected <code>\$SequenceVar(oidName, index)</code> .	N/A
1018	Current OID already contains a static value.	This OID has already been declared static.
1019	Invalid number of index parameters received.	All SequenceVar must include only one index.
1020	OID of sequence type cannot contain static data.	All sequence OID variables must be dynamic.
1021	This is a duplicate OID or the root of this OID is not the same as previous OID(s), or this OID is a child of a previously defined OID.	All OID strings must contain the same root OID.
1022	Invalid index received; must be BYTE data value.	All sequence index OID must be of data type, BYTE.
1023	Invalid OID access type received; must be READONLY or READWRITE.	N/A
1024	Current OID is already assigned an ID value.	Current OID is already declared as dynamic.
1025	Duplicate dynamic ID found.	Current OID is already declared as dynamic with duplicate ID.
1026	No static value found for this OID.	Current OID was declared static, but does not contain any data.

1027	No index value found for this OID.	Current OID was declared as sequence, but does not contain any index.
1028	OID data scope (dynamic/static) is not defined.	Current OID was declared, but was not defined to be static or dynamic.
1029	Invalid data value found.	Data value for current OID does not match with its data type.
1030	Invalid parameters: expected \$AgentID(oidName, id).	N/A
1031	Only OID data type is allowed for this command.	AgentID command must use OID name of OID data type.
1032	This OID must contain static OID data.	AgentID command must use OID name of static data.
1033	This OID is already declared as an Agent ID.	Only one AgentID command is allowed.
1034	An Agent ID is already assigned.	Only one AgentID command is allowed.
1035	OID with READWRITE access cannot be static.	An OID was declared READWRITE and made static.
1036	OID of OID data type cannot be dynamic.	Current version does not support OID variable of data type, OID.
1037	This OID is already declared as dynamic.	N/A
1038	This OID is already declared as static.	N/A
1039	This OID does not contain the Internet root. The Internet root of '43' must be used if this is an Internet MIB.	All internet OIDs must start with '43'. This is a warning only and will not stop script generation.
1040	The given value was truncated to fit in a specified data type.	An OID was declared as BYTE or WORD but the value given in StaticVar exceeded the data range.
1041	The given string exceeds a maximum length of 127.	All OCTET_STRING and ASCII_STRING must be less than 128.
1042	Invalid OID name detected; OID name must follow standard 'C' variable naming convention.	All OID names must follow 'C' naming convention as these names are used to create 'define' statements in the mib.h file.
1043	Total number of dynamic OIDs exceeds 1023.	This version supports total dynamic OIDs of 1024 only. All dynamic OID IDs must range from 0-1023.

## Index

### M

mib2bib.jar Run-time Error Codes 4

Microchip MIB Compiler 3

### V

Volume VII: Utilities 2