## Name

**makeobj** — convert a binary file to an OMF file.

## Synopsis

**makeobj** [options] *file*

## Description

**makeobj** converts a binary file (such as image or sound data) to an OMF (object module format) file. The OMF file can be linked and accessed as if it was externally declared.

## Example

```
makeobj -o binary.o -n picture binary.data
```

The data can now be accessed as if it were an external array:

```
extern char picture[];
```

## Options

**makeobj** recognizes the following options:

**-a** *number* Set the OMF file segment. The alignment must be a power of 2. The default alignment is 0.

**-n** *name* Set the segment name. The default loadname is the same as the input filename (minus any file extension). This is the name of the data in your program. C programs are case sensitive. If you're using assembly language, the segment name should be in capital letters if case sensitivity is off.

**-l** *name* Set the segment load name. The default load name is blank. The segment load name is used for splitting large programs into multiple segments.

**-k** *kind* Set the segment kind. Valid values are "CODE", "DATA", "INIT", or "STACK". The default kind is DATA. Please note that DATA segments may cross bank boundaries when loaded. If your data is < 65,535 (`$ffff`) bytes in length and you do not want it to cross bank boundaries, specify the CODE kind.

**-o** *file* Set the output file name. The default output file name is the same as the input filename, but with a ".o" extension.

**-h** Display help and version information.