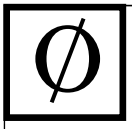
Chapter Pub

Publishing Results



Pub.0 Overview

After building a model, using it in a simulation, and analyzing the results, you may need to report the results to others. In addition, it is often important to document the design of a model, report the status of model development to other members of a project team, or provide a “snap-shot” of a model to technical support staff.

OPNET supports these documentation needs by providing the ability to capture and print graphics, print detailed model descriptions, and print data files. All files use standard formats, such as PostScript and ASCII text, to give you flexibility when processing the files to meet documentation needs.

The first three sections of this chapter describe the documentation support available in OPNET for graphics, reports, and text files. These sections give instructions for the following types of operations:

- Printing model graphics
- Capturing bitmap graphics
- Printing model reports
- Printing text files and the contents of edit pads

The last section discusses how OPNET uses print spooling, where it stores output files, and what environment attributes affect the printing and formatting of output graphics and text files.

Pub.1 Graphics

OPNET provides two operations for capturing graphics from the workstation screen—the **Print Graphics** operation and the **Bitmap** operation. Most OPNET tools and utilities support at least one of these operations; many support both.

Both operations allow you to define a portion of the tool window for capture. The primary difference in these operations is that Print Graphics allows you to place several different graphics on the same page before printing, whereas **Bitmap** captures a single graphic per file and may not print automatically. The following table summarizes the differences between these graphics operations.

Print vs. Bitmap Graphics

	Print Graphics	Bitmap Graphics
Graphics captured from…	specified region of tool window	any portion or all of program window
Number of graphics per file	many	one
File formats	PS (PostScript)	PS, EPSI, or TIFF
Automatically sent to printer?	yes	yes (PS, EPSI) no (TIFF)

Sections *Pub.1.1 Print Graphics* and *Pub.1.2 Bitmaps* describe these graphics operations in detail.

OPNET puts all graphics output files in the temporary file directory of your administration directory (**$HOME/op_admin/tmp**). The files remain in this directory until you remove them, either manually or with the **m3_clrtmp** utility. This allows you to edit the files, print more copies, or import the graphics into other documents.

Pub.1.1 Print Graphics

The **Print Graphics** operation allows you to document the graphic content of a model. It produces a printed page containing one or more graphics captured from tool windows. You control the size of the captured region, the size of the printed graphic, and the orientation of the graphic (either portrait or landscape). You can add a title and any number of additional graphics (any number of tools) before printing the page.

Printer Page

When you choose Print Graphics the first time, **opnet** creates a *printer page* to hold the captured graphics. The printer page is a PostScript file that stores the captured graphics until they are printed. Initially, this file contains only the information necessary to define the start of an empty PostScript page. Each time you capture a graphic, **opnet** appends PostScript commands describing the graphic according to a layout you specify. The file

remains incomplete until you activate the **Print Page** operation. **opnet** then completes (saves) the file, sends it to the print spooler, and clears the printer page.

File Storage and Naming

As mentioned, the printer page is stored as a PostScript file. The PostScript commands are in ASCII format, which allows you to view or modify them with a text editor. The ASCII format also permits easy transfer by electronic mail.

The name of each graphics file is composed of:

- A file prefix indicating the tool that initially created it
- The process ID of the program's process
- A printout sequence number for the process
- The suffix "**.ps**" (indicating the PostScript format)

As shown below:

<*file_prefix*>.<*process_id*>.<*sequence_num*>.ps

The following table lists the possible file prefixes.

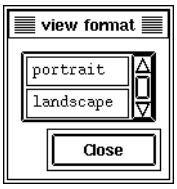
Print Graphics File Prefixes	
Tool Name	**<file_prefix>**
Network Editor	**ntgfx**
Analysis Tool	**angfx**
m3_vuorb Editor	**viewgfx**

To capture a model using Print Graphics…

1) With the desired model graphic displayed in a tool window, activate the **Print Graphics** action button.

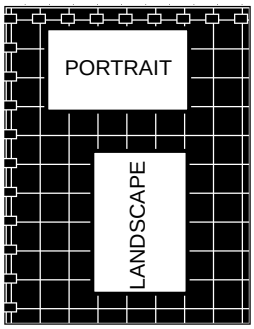
2) Select an area in the tool window to be captured.

   ➡ The **`view format`** dialog box pops up.



3) In the **`view format`** dialog box, select portrait or landscape orientation for the captured graphic. Portrait orientation places the graphic on the page as seen on the screen. Landscape rotates the graphic 90 degrees counter-clockwise before placing it on the page.

   ➡ A representation of the printer page pops up. The numbers give distances from the horizontal and vertical centerlines of the page, in inches.

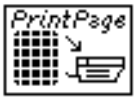


4) Select an area on the printer page to receive the captured graphic. The program automatically constrains the lower-right corner (as oriented) to keep the same aspect ratio as the captured graphic.

5) If desired, repeat steps 1–4 to add other graphics to the printer page. Before capturing each graphic, you can open a new tool window or load other models.

6) If you misplace a graphic or want to start over at any time, activate the **`Clear Page`** system button. This operation clears the internal record of the page and removes the PostScript file.

7) When you have placed all desired graphics on the printer page, activate the **Print Page** system button to send the page to the print spooler.



➡ A dialog box pops up with a text entry area for entering a title to be printed at the top of the page.

8) Type a page title, if desired, and press **<Return>**.

➡ OPNET spools the file and clears the internal record of the printer page.

Pub.1.2 Bitmaps

Although the **Print Graphics** operation is good for producing stand-alone graphics hard copy, its files have several limitations. Therefore, OPNET provides an additional operation called **Bitmap.** All **M3UI**-based programs support the **Bitmap** operation, including **opnet**, **m3_edmap**, **m3_vuanim**, and **m3_vuorb**.

Bitmaps have the following advantages over Print Graphics images:

- They can capture a wider variety of graphics, including areas such as header strips, message areas, and button panels.
- They generally use less memory, if captured in **TIFF** format.
- They are easier to edit and import into desktop publishing programs (in **TIFF** format).

You can capture three types of bitmaps—full screens, partial screens, and pop-up areas. This section first describes the different file formats used by bitmap files. It then discusses each type of bitmap capture and gives instructions for using it.

Bitmap File Formats

The Bitmap operation uses one of three file formats, depending on how the bitmaps are created and which format you specify. Full-screen bitmaps always use the same PostScript format ("**.p**s") used by Print Graphics and described in the previous section. Partial-screen and area bitmaps use either Encapsulated PostScript Interchange (**EPSI**) or Tagged Image File Format (**TIFF**), as specified by the **screen_bitmap_type** environment attribute.

EPSI files are **ASCII** text files that contain a description of the captured bitmap in the PostScript language. They also contain a bitmap preview image that allows programs importing the **EPSI** file to display an image on-screen. This makes **EPSI** files larger than corresponding "**.ps**" files. **EPSI** files are also limited to a single image on the page. Like

PostScript files, **EPSI** files can be spooled directly to PostScript printers and are difficult to edit once generated.

TIFF files are binary files that contain a compressed representation of the captured bitmap. They are much smaller than **EPSI** files—typically one-fourth the size of the same bitmap in **EPSI**. The size advantage makes **TIFF** a better format for graphics that are being imported into desktop publishing documents. Also, **TIFF** files can easily be edited with many bitmap graphics programs.

Full-Screen Bitmaps

A full-screen bitmap captures the entire program window and stores it as a single-page PostScript document. The bitmap graphic is aligned to the upper left corner of the page. In other respects, a full-screen bitmap file has the same characteristics as a **Print Graphics** file.

The name of each full-screen bitmap file includes the bitmap type ("**fsbitmap**"), the process ID of the program's process, a printout sequence number for the process, and a suffix "**.ps**" (indicating the PostScript format), as follows:

fsbitmap.<*process_id*>.<*sequence_num*>.ps

To capture a full-screen bitmap…

1) With the desired screen displayed, middle-click the **Bitmap** system button.

➡ If necessary, OPNET redraws the window in monochrome. It then scans the window as a sequence of small square bitmaps. Each bitmap, in turn, is translated into PostScript and added to the bitmap file. When the scan is complete, **opnet** passes the file to the print spooler.

Partial-Screen Bitmaps

A partial-screen bitmap captures a user-specified region of the program window. You specify the region in the same way as for **Print Graphics**.

The name of each partial-screen bitmap file includes the bitmap type ("**sbitmap**"), the process ID of the program's process, a printout sequence number for the process, and a suffix indicating the file format (either "**.tiff**" or "**.epsi**"), as follows:

sbitmap.<*process_id*>.<*sequence_num*>.tiff

sbitmap.<*process_id*>.<*sequence_num*>.epsi

To capture a partial-screen bitmap…

1) With the desired screen displayed, left-click the **Bitmap** system button.

2) Select an area of the window region to be captured.

➡ If necessary, OPNET redraws the selected area in monochrome (see *Color Output* in section *Pub.4.2 Printing Environment Attributes* for an explanation). It then scans the area as a sequence of small square bitmaps. Each bitmap, in turn, is added to the bitmap file.

➡ When the scan is complete, OPNET translates the file into the specified format (either **TIFF** or **EPSI**) and stores the file. Files in **EPSI** format are also passed to the print spooler.

Area Bitmaps

Area bitmaps make it easy to capture a bitmap of buttons and pop-up objects such as dialog boxes and text edit pads. Unlike partial-screen bitmaps, area bitmaps automatically adjust to the size of the object being captured. Area bitmaps also allow you to specify a name for the file or use a default name.

The default name of each area bitmap file includes a prefix based on the type of area captured, the process ID of the program's process, a printout sequence number for the process, and a suffix indicating the file format (either "**.tiff**" or "**.epsi**"), as follows:

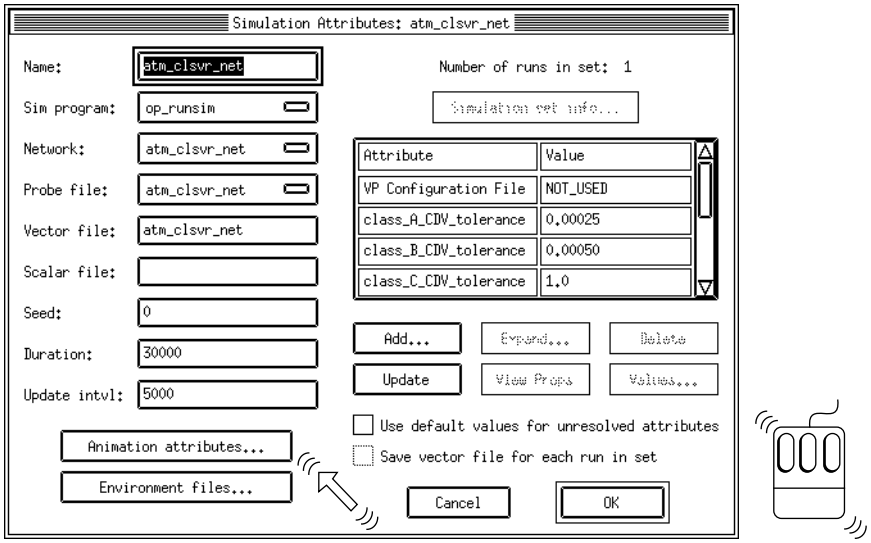
<prefix>.<process_id>.<sequence_num>. [*tiff* | *epsi*]

The following table lists the possible file prefixes.

Area bitmap File Prefixes	
Pop-Up Area Type	**<prefix>**
Dialog box	**dbox**
Edit pad	**edpad**
Button	**button**

To capture an area bitmap…

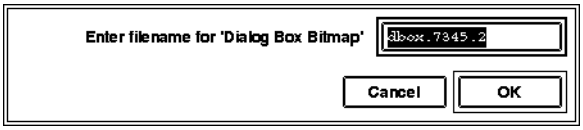
1) Move the cursor over the desired object.



2) Press **<Control>+<B>**.

Note: This is the default key combination. You can change it with the **m3_edkey** *program.*

➡ A dialog box for naming the bitmap file appears.



3) Type a name for the bitmap file and press **<Return>**, or click **OK** to accept the default name.

➡ The naming dialog box disappears and OPNET scans the object into a bitmap file. If necessary, it first redraws the selected area in monochrome. When the scan is complete, OPNET translates the file into the specified format (either TIFF or EPSI) and stores it. Files in EPSI format are also passed to the print spooler. The name of the bitmap file generated is <*name*>.tiff or <*name*>.epsi, where <*name*> is the bitmap file name entered in the naming dialog box.

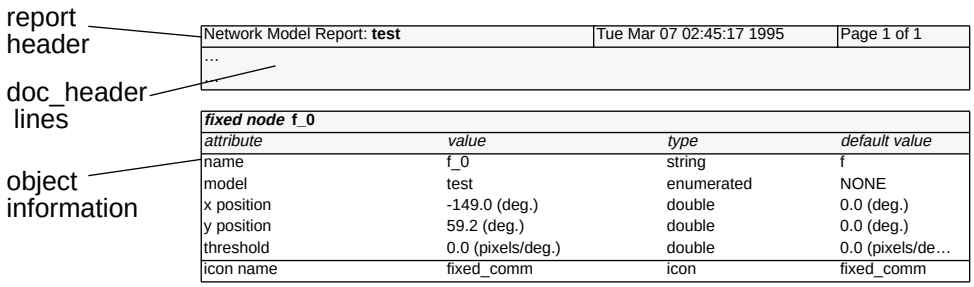
Pub.2 Reports

A detailed model specification requires descriptions of the objects in the model, their attribute values, and the connections among them. Graphics alone cannot provide all of this information. To supplement graphics, therefore, OPNET provides a **Print Report** operation that automatically generates a report describing either the complete model contained within the currently-active tool or selected objects within the model. The following OPNET tools can produce reports:

- Network Editor (**opnet**)
- Probe Tool (**opnet**)
- Simulation Tool (**opnet**)

Each report contains tables of information about objects in the model. The report format depends on which tool produces the report and the needs of that tool. For example, a network model report lists each object (such as a node or link), the attributes of the object, and the value, type, and default value of each attribute.

Sample Report Output



Report Header

Reports automatically include a header at the top of each page. The header gives the tool and model names, the date and time the report was produced, and a page number. You can add up to two lines of additional text to further describe the report or give other needed information. To include the user-defined text, set the environment attributes **doc_header_1** and **doc_header_2**, as described in the *System Environment* chapter of the *OPNET External Interfaces Manual*.

File Creation and Printing

Like **Print Graphics** files, report files are created in **ASCII** PostScript format. Reports can be several pages long, however, and are spooled for printing immediately when you activate the **Print Report** operation.

After spooling, each report is available in the temporary file directory of your administration directory (**`$HOME/op_admin/tmp`**). The name of each report file depends on the tool that initially created it, the process ID of the program's process, a printout sequence number for the process, and the suffix "**`.ps`**" (indicating the PostScript format), as follows:

<file_prefix>.<process_id>.<sequence_num>.ps

The following table lists the possible file prefixes.

Report File Prefixes	
Tool Name	**`<file_prefix>`**
Network Editor	**`ntrep`**
Probe Tool	**`pbrep`**
Simulation Tool	**`seqrep`**
FLM Tool	**`flmrep`**

To print a model report…

1) Load the desired model in the tool window.

2) Decide whether to report on the entire model or only certain objects in the model:

   a) To report on the entire model, make sure none of the objects is selected (no selection markers visible).

   b) To report only on certain objects, select each desired object. Selection markers appear on each object as you select it.

3) Activate the **`Print Report`** action button.

➡ OPNET creates the report file and spools it for printing.

Pub.3 Text Files

Another type of textual documentation you might need to create is listings of **ASCII** text files. OPNET uses such files for several purposes, such as General Data Files (**GDF**s) and C language source code files. You may be able to use various utilities (such as **pr** and **lp**) to format and print these files, but these are not always available. OPNET provides two methods of printing text files—a formatting utility called **m3_prtext** and a special key-stroke combination to print the contents of edit pads.

Pub.3.1 m3_prtext

The OPNET text formatting and printing utility, **m3_prtext**, converts the contents of an ASCII text file into PostScript for printing. In doing so, it performs the following formatting:

- Sets text in a user-specified PostScript font family and point size, as given by the environment attributes **psfont** and **ptsize**.
- Adds page breaks as required by the page capacity at the font point size being used. Also adds page breaks where indicated in the file (by standard **<Control>+<L>** characters) if the **ctrl_page_delim** environment attribute is set.
- Numbers lines in five-line increments.
- Adds page headers with the name of the file being printed, the date and time, the page number and total pages, and two user-specified lines. The environment attributes **doc_header_1** and **doc_header_2** provide text for the user-specified lines.
- Formats C language source code. This includes italic font for comments, courier font for string constants, and optional boldface font for OPNET Kernel Procedure invocations (if the environment attribute **bold_opfun** is set).

After **m3_prtext** formats a text file, it passes the file to the print spooler. After printing, the file is available in the temporary file directory of your administration directory (**$HOME/op_admin/tmp**).

The name of each formatted text file consists of the file identifier "**prtdoc**", the process ID of the program process, a printout sequence number for the process, and the suffix "**.ps**" (indicating the PostScript format), in the following form:

```
prtdoc.<process_id>.<sequence_num>.ps
```

To print a text file…

1) Move to the directory containing the file to be printed.

2) At the command line prompt, type:

```
m3_prtext -f <file_name>
```

➡ where **<file_name>** is the name of the file to be printed. You can also specify environment attributes as options on the command line, if desired.

For a detailed description of the **m3_prtext** utility and the available options, refer to the *Program Descriptions* chapter of the *OPNET External Interfaces Manual*.

Pub.3.2 Text Edit Pads

In addition to printing text files with **m3_prtext**, you can print the contents of a text edit pad directly from M3UI-based programs.

To print a text edit pad…

1) Move the cursor within the boundaries of the text edit pad you want to print.

2) Press **<Control>+<P>**.

Note: This is the default key combination. You can change it with the **m3_edkey** *program.*

➡ OPNET puts the edit pad contents in a text file in the temporary file directory of your administration directory (**$HOME/op_admin/tmp**). The file is then passed to the **m3_prtext** utility, which is spun off as a background process to the current program. From there on, the edit pad text file is treated like other text files, as described in section *Pub.3.1 m3_prtext*.

Pub.4 Printing and File Handling

OPNET programs that generate PostScript output files for graphics and text automatically print those files when they are created. However, these programs do not send their output directly to a printer. Instead, they send the output files to a print spooling program, as described in the next section.

OPNET puts all graphics and textual output files, whether spooled for printing or not, in the temporary file directory of your administration directory (**`$HOME/op_admin/tmp`**). The files remain in this directory until you remove them, either manually or with the **m3_clrtmp** utility. This allows you to edit the files, print more copies, or import the file contents into other documents.

Pub.4.1 Print Spooling

To print output files, OPNET programs pass them to the program specified by the environment attribute **`print_prog.`** This is normally a spooler such as **lpr.** However, you can specify a different program or shell script for special purposes, such as suppressing all printing or moving output files to another location.

Print spooling can occur in the foreground or background. By default, it is spun off as a background process. However, if your spooling program requires that an active spool operation finish before another begins, you can use the **`print_wait`** environment attribute to run the spool process in the foreground.

Refer to the *System Environment* chapter of the *OPNET External Interfaces Manual* for details on using environment attributes.

Pub.4.2 Printing Environment Attributes

Environment attributes control the generation of print output files (both graphics and reports), screen bitmap files, and the spooling of print output files to a PostScript printer. These attributes are valid in all OPNET programs that generate output files. The following table summarizes the printing environment attributes. Refer to the *System Environment* chapter of the *OPNET External Interfaces Manual* for instructions on using environment attributes.

Printing Environment Attributes

Name	Values	Description
`page_format_a4`	**TRUE** **FALSE** (default)	Format PostScript output for European A4 paper or U.S. standard letter size paper (8.5" in. x 11" in.).
`print_color`†	**TRUE** **FALSE** (default)	Generate graphics in color or monochrome.

Printing Environment Attributes		
Name	Values	Description
print_prog	***<name>*** (default is **lpr**)	Use the specified the print spooling program.
print_wait	**TRUE** **FALSE** (default)	Invoke the print spooling program as a foreground or background process.
screen_bitmap_type†	**EPSI** (default) **TIFF**	Generate screen bitmap files in EPSI or TIFF format.
unique_tmp_names	**TRUE** (default) **FALSE**	Include or do not include the process ID in the names of generated graphics and report files.

†. Available in M3UI-based programs only.

To set environment attributes for printing…

1) To set the environment attributes for capturing color **TIFF** graphics (for example), follow one of these steps:

   - At the command line prompt, type:

     ```
     <program_name> -screen_bitmap_type tiff -print_color true
     ```

     and press **<Return>**. (**<program_name>** is the name of the OPNET program you want to run.)

   - Add the environment attributes to the **env_db** file before running a program, as follows:

     ```
     screen_bitmap_type:   tiff
     print_color:          true
     ```

Page Size

The default page size for all reports printed by OPNET is U.S. standard letter size paper (8.5 in. x 11 in.). You can specify European-standard A4 size paper by setting the **page_format_a4** environment attribute to **true**.

Color Output

OPNET can generate graphics files in color or monochrome (black and white). The default is monochrome. To generate color files, set the **print_color** environment attribute to **true**.

Note: Color graphics files are larger than monochrome ones. Also, they are available only from color workstations.

When capturing monochrome graphics from a color display, OPNET programs must first redraw the area to be captured in monochrome. This process, which prepares the screen buffer for graphics capture, may take a few seconds.

Short Filenames

You can shorten the names of graphics and report files by setting the **`unique_tmp_names`** environment attribute to **`false`**. This produces names without the process ID. However, names without a process ID are much less likely to be unique, possibly resulting in overwritten files.