
A Graphic Conversion Utility for FM+SGML

Author: Dan Emory

1 Introduction

Conversions from one graphic format to another are often problematic. If you have FM+SGML, you can create a graphic conversion utility that's guaranteed to work because it uses the FM+SGML graphic export filters to produce the conversions.

The graphic conversion utility is implemented as an SGML import/export application that has its own Element Definition Document (EDD), Document Type Definition (DTD), read/write rules, and structured document template containing the EDD's element catalogs. The EDD and the read/write rules (both fully commented to describe the implementation) are provided as attachments to this document.

2 How to Create and Use it

2.1 Creating the Application

- Step 1. Create the EDD and the read/write rules.
- Step 2. Add the Graphic Conversion application to the SGMLAPPS file, specifying the following:
 - a. The filename of the DTD for export (you haven't created the DTD yet, but preassign a filename for it).
 - b. The filename of the read/write rules.
 - c. The filename of the template for export (you haven't created the template yet, but preassign a filename for it).
 - d. The filename of the SGML declaration (the default sgmldecl file will do).
- Step 3. With the SGML Application Definition file still open, choose File > Developer Tools > Reread SGML Application File. When the reread action is completed close and save the SGML Application Definition file.
- Step 4. Open the EDD created in step 1.
- Step 5. Choose File > Set SGML Application, and Select Graphic Conversion as the application.
- Step 6. Choose File > Developer Tools > Save as DTD. The file should be saved under the name specified in step 2a.
- Step 7. Choose File > New to create a new blank paper Portrait file.
- Step 8. Choose File > Import > Element Definitions and import the element definitions from the EDD into the new document created in step 7.
- Step 9. Save the template created in steps 7 and 8 under the name assigned to it in step 1c.

2.2 Converting Graphics with the Utility

- Step 1. Choose File > New and specify the filename of the template for import (specified in the SGML Application Definition file for the Graphic Conversion application) as the template.
- Step 2. Choose File > Set SGML Application, and set the application to Graphic Conversion.
- Step 3. Open the Element Catalog and the Structure View. The Element Catalog should show that only one element is valid--the top-level Graphic element. Insert it. If the graphic to be converted already exists as a file, proceed to step 4, otherwise, proceed to step 5.
- Step 4. In the Element Catalog, insert the External element. The Import file dialog opens. Select the filename of the graphic to be converted, and turn on Import by Reference. Proceed to step 7.
- Step 5. Copy the graphic to be converted to the clipboard.
- Step 6. In the Element Catalog, insert the Internal Element. The Anchored Frame dialog opens. Specify the size of the anchored frame, and paste the graphic to be converted into the anchored frame.
- Step 7. Make any required adjustments to the size of the graphic and the size of the anchored frame.
- Step 8. If you want to export the graphic in the Encapsulated Postscript format, proceed to step 10. Otherwise, proceed to step 9.
- Step 9. In the structure view, click on the Internal or External element. Then, in the element catalog, select the element that produces the desired format, and click the change button.
- Step 10. Choose File > Save As. In the Save Document dialog, specify an appropriate filename (with a .sgm extension), choose SGML from the format menu, and save the document. The converted graphic is saved under a filename whose stemname includes the name of the graphic element (Internal, External, or the graphic element chosen in step 9), and whose extension identifies the graphic type.
- Step 11. You can check the exported graphic by opening the SGML document instance produced in step 10 (i.e., the filename you specified in the Save Document dialog in step 10). FM+SGML will import (by reference) the graphic that was produced in step 10. You can then delete the SGML document instance, since it is no longer needed. Finally, move the exported graphic file to the directory where it is used.

EDD Version is 5.0

SGML Application: Graphic Conversion

Automatically create formats on import.

This EDD is used to create a Graphic Conversion utility, in which, on export to SGML, graphics are unconditionally written out in the format(s) described in the comments below for each graphic element. The EDD, as presently written can export any graphic in the CGM, GIF, MIF, TIFF, IGES, JPEG, PICT, BMP, EPSI or FrameVector format.

You can create the DTD used for export by saving this EDD as the DTD for export. You must also add the Graphic Conversion application to the SGMLAPPS application definition file In the application definition, you must specify:

1. The name of the application (Graphic Conversion)
2. The file name of the DTD for export (created from this EDD).
3. Import this EDD into a template, and specify the filename of that template as the template for import.
4. The filename of the read/write rules file, which defines, for each graphic element, the graphic format in which the graphic is unconditionally written out. The read/write rules also specify that the stemnames to be used for the written-out graphic files are the same as the element names. The filename extension indicates the graphic format type (e.g., BMP for bitmap)

Additional graphic elements can be added to accomplish conversions to other graphic formats. For each such graphic element, the appropriate read/write rule must be added.

The normal procedure is to create a new structured document from the template for import. Then, insert the top-level Graphic element. Next, do one of the following:

1. If the graphic already exists as an external file, insert an External element. The Import File dialog will open, allowing you to select the graphic to be imported by reference.
2. If the graphic does not exist as an external file, first copy the graphic (but NOT including its anchored frame) from a FrameMaker document to the clipboard, then, insert an Internal element, paste the graphic into the empty anchored frame that is produced, and adjust the anchored frame's height/width to fit the graphic size.

The Internal and External elements will both export the graphic in the encapsulated postscript (EPSI) format. If you want to export the graphic in a format other than EPSI, change the name of the graphic element from Internal or External to EX2XXX, where XXX is the desired graphic format. Finally, set the SGML Application to Graphic Conversion, and save the document to SGML.

You can verify that the graphic conversion was successfully accomplished by opening in FM+SGML the SGML document instance produced on export. Since the SGML document instance has no other purpose, it can be deleted after you have verified that the conversion is satisfactory.

=====

This is the top-level wrapper element for the graphic element(s) containing the anchored frame(s). Note that the general rule states that any number of graphic elements of different types can be contained under the top-level Graphic element.

Element (Container): Graphic

General rule: (Internal | External | EX2MIF | EX2TIF | EX2CGM | EX2GIF | EX2PIC | EX2JPG | EX2IGS | EX2BMP | EX2FMV)*

Valid as the highest-level element.

Use this one to create an anchored frame into which you then paste a graphic that was copied to the clipboard, and/or graphic objects created with the FrameMaker drawing tools. When the graphic is ready for export, adjust the size of the anchored frame to fit the graphic, and/or scale the graphic up or down in size. The graphic will be unconditionally exported in the EPSI format. If you want to export it in a format other than EPSI, change the element name to EX2XXX, where XXX is the desired export format.

Element (Graphic): Internal

Initial graphic element format

1. In all contexts.
Insert anchored frame.

Use this one to import an existing graphic contained in an external file. The graphic will be unconditionally exported in the EPSI format. If you want to export it in a format other than EPSI, change the element name to EX2XXX, where XXX is the desired export format.

Element (Graphic): External

Initial graphic element format

1. In all contexts.
Insert imported graphic file.

If you want to export the graphic in MIF format after first importing it into an External element, or after pasting it or drawing it into an Internal element, change the element name from Internal or External to EX2MIF.

Element (Graphic): EX2MIF

Initial graphic element format

1. In all contexts.
Insert anchored frame.

If you want to export the graphic in TIFF format after first importing it into an External element, or after pasting it into an Internal element, change the element name from Internal or External to EX2TIF

Element (Graphic): EX2TIF

Initial graphic element format

1. **In all contexts.**
Insert anchored frame.

If you want to export the graphic in CGM format after first importing it into an External element, or after pasting it into an Internal element, change the element name from Internal or External to EX2CGM

Element (Graphic): EX2CGM

Initial graphic element format

1. **In all contexts.**
Insert anchored frame.

If you want to export the graphic in GIF format after first importing it into an External element, or after pasting it into an Internal element, change the element name from Internal or External to EX2GIF

Element (Graphic): EX2GIF

Initial graphic element format

1. **In all contexts.**
Insert anchored frame.

If you want to export the graphic in PICT format after first importing it into an External element, or after pasting it into an Internal element, change the element name from Internal or External to EX2PIC

Element (Graphic): EX2PIC

Initial graphic element format

1. **In all contexts.**
Insert anchored frame.

If you want to export the graphic in JPEG format after first importing it into an External element, or after pasting it into an Internal element, change the element name from Internal or External to EX2JPG

Element (Graphic): EX2JPG

Initial graphic element format

1. **In all contexts.**
Insert anchored frame.

If you want to export the graphic in BMP/DIB format after first importing it into an External element, or after pasting it into an Internal element, change the element name from Internal or External to EX2BMP

Element (Graphic): EX2BMP

Initial graphic element format

1. **In all contexts.**
Insert anchored frame.

If you want to export the graphic in FrameVector format after first importing it into an External element, or after pasting it into an Internal element, change the element name from Internal or External to EX2FMV

Element (Graphic): EX2FMV

Initial graphic element format

1. **In all contexts.**
Insert anchored frame.

If you want to export the graphic in IGES format after first importing it into an External element, or after pasting it into an Internal element, change the element name from Internal or External to EX2IGS

Element (Graphic): EX2IGS

Initial graphic element format

1. **In all contexts.**
Insert anchored frame.

fmsgml version is "5.5";
/*

=====

READ/WRITE FULES FILE FOR GRAPHIC CONVERSION UTILITY

Note that each rule specifies that the filename of the exported graphic is specified as `$(entity).typ`, where:

- entity** is the name of the graphic element containing the graphic. This becomes the stemname of the exported file.
- typ** is the extension used to identify the graphic type that is exported.

If the FM+SGML document contains multiple graphic elements having the same name, a unique number is appended to each stemname. Also, if you export the same FM+SGML document multiple times, different numbers are appended to the stemname of each file each time the FM+SGML document is re-exported.

Note also that the rule for each each graphic element handles all of the following cases:

1. The original graphic was imported by reference into the FM+SGML document.
2. The original graphic was imported by copy into the FM+SGML document.
3. The original graphic was pasted into a pre-existing anchored frame.
4. The original graph was created in a pre-existing anchored frame using the FrameMaker drawing tools.
5. Multi-faceted graphics (i.e., a single anchored frame that contains two or more graphics, and/or graphics that are overlaid with leader lines, callouts, or other other objects created with the FrameMaker drawing tools).

For case 1 above, the **facet default** subrule applies.

For cases 2 thru 5 above, the **anchored frame export to file** subrule applies

```
=====
*/
element "Graphic" {
    is fm element "Graphic";
}
element "External" {
    is fm graphic element "External";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).eps" as "EPSI";
        }
        anchored frame export to file "$(entity).eps" as "EPSI";
    }
}
element "Internal" {
    is fm graphic element "Internal";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).eps" as "EPSI";
        }
        anchored frame export to file "$(entity).eps" as "EPSI";
    }
}
element "exp2mif" {
```

```

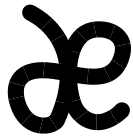
    is fm graphic element "EXP2MIF";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).mif" as "MIF";
        }
        anchored frame export to file "$(entity).mif" as "MIF";
    }}
element "exp2tif" {
    is fm graphic element "EXP2TIF";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).tif" as "TIFF";
        }
        anchored frame export to file "$(entity).tif" as "TIFF";
    }}
element "exp2cgm" {
    is fm graphic element "EXP2CGM";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).cgm" as "CGM";
        }
        anchored frame export to file "$(entity).cgm" as "CGM";
    }}
element "exp2gif" {
    is fm graphic element "EXP2GIF";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).gif" as "GIF";
        }
        anchored frame export to file "$(entity).gif" as "GIF";
    }}
element "exp2bmp" {
    is fm graphic element "EXP2BMP";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).bmp" as "DIB";
        }
        anchored frame export to file "$(entity).bmp" as "DIB";
    }}
element "exp2pic" {
    is fm graphic element "EXP2PIC";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).pic" as "QuickDraw PICT";
        }
        anchored frame export to file "$(entity).pic" as "QuickDraw PICT";
    }}
element "exp2jpg" {

```

```

    is fm graphic element "EXP2JPG";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).jpg" as "JPEG";
        }
        anchored frame export to file "$(entity).jpg" as "JPEG";
    }}
element "exp2fmv" {
    is fm graphic element "EXP2FMV";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).fmv" as "FrameVector";
        }
        anchored frame export to file "$(entity).fmv" as "FrameVector";
    }}
element "exp2igs" {
    is fm graphic element "EXP2IGS";
    writer {
        facet default {
            convert referenced graphics;
            export to file "$(entity).igs" as "IGES";
        }
        anchored frame export to file "$(entity).igs" as "IGES";
    }}

```

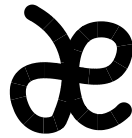


Dan Emory & Associates

Information Design Specialists

Dan Emory

10044 Adams Ave. #208
Huntington Beach, CA 92646
Voice/Fax: 949-722-8971
Email: danemory@primenet.com



Dan Emory & Associates

Information Design Specialists

Dan Emory

10044 Adams Ave. #208
Huntington Beach, CA 92646
Voice/Fax: 949-722-8971
Email: danemory@primenet.com

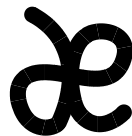


Dan Emory & Associates

Information Design Specialists

Dan Emory

10044 Adams Ave. #208
Huntington Beach, CA 92646
Voice/Fax: 949-722-8971
Email: danemory@primenet.com



Dan Emory & Associates

Information Design Specialists

Dan Emory

10044 Adams Ave. #208
Huntington Beach, CA 92646
Voice/Fax: 949-722-8971
Email: danemory@primenet.com

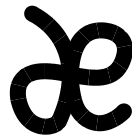


Dan Emory & Associates

Information Design Specialists

Dan Emory

10044 Adams Ave. #208
Huntington Beach, CA 92646
Voice/Fax: 949-722-8971
Email: danemory@primenet.com



Dan Emory & Associates

Information Design Specialists

Dan Emory

10044 Adams Ave. #208
Huntington Beach, CA 92646
Voice/Fax: 949-722-8971
Email: danemory@primenet.com



Dan Emory & Associates

Information Design Specialists

Dan Emory

10044 Adams Ave. #208
Huntington Beach, CA 92646
Voice/Fax: 949-722-8971
Email: danemory@primenet.com



Dan Emory & Associates

Information Design Specialists

Dan Emory

10044 Adams Ave. #208
Huntington Beach, CA 92646
Voice/Fax: 949-722-8971
Email: danemory@primenet.com