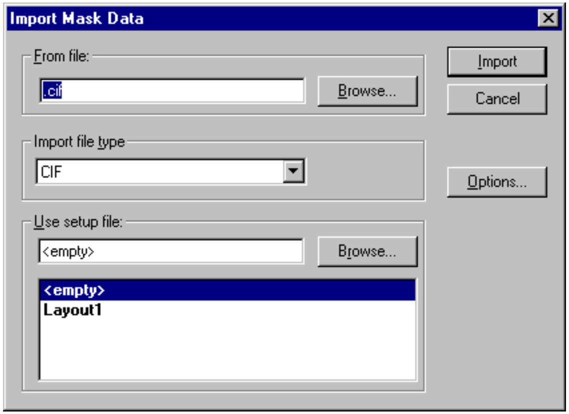
5 Importing and Exporting Files

Importing Files

Use **File > Import Mask Data** to import CIF or GDSII files into L-Edit.



From file: .cif

Import file type: CIF

Use setup file: <empty>

<empty>
Layout1

Options include:

From file Name of the file containing the design data to be imported.

Import file type Format of the file being imported. Options are **CIF** or **GDSII**.

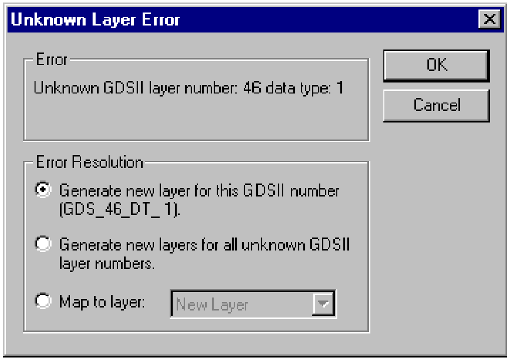
Use setup file Specifies a TDB setup file containing the necessary layer setup information.

Import Imports the specified file

Options Accesses a dialog for selecting import options for CIF or GDSII files, as appropriate. See CIF Import Options on page 1-188 and GDSII Import Options on page 1-189.

After importing a GDSII file, L-Edit produces a log file summarizing settings, showing status, and providing detailed warning and error messages.

If your design has a layer for which there is no corresponding layer in the setup file, L-Edit will display the dialog shown below



Error

Unknown GDSII layer number: 46 data type: 1

Error Resolution

● Generate new layer for this GDSII number (GDS_46_DT_ 1).

○ Generate new layers for all unknown GDSII layer numbers.

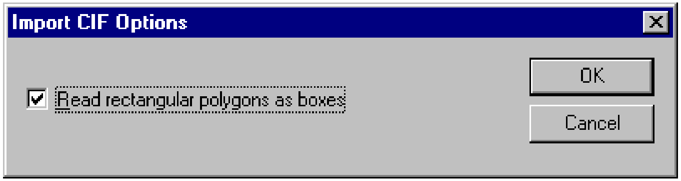
○ Map to layer: New Layer

Options include:

- Creating a new layer for the unknown value
- Generating a new layer for all unknown values
- Mapping the unknown value to an existing layer

CIF Import Options

For CIF files, you can instruct L-Edit to read rectangular polygons as boxes. This option speeds processing, as boxes consume less memory and are drawn faster than polygons. Click the check box to turn this option on.

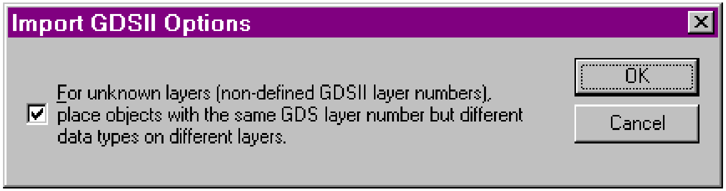


Import CIF Options

☑ Read rectangular polygons as boxes

GDSII Import Options

For GDSII files, you can instruct L-Edit to use both the **GDSII number** and the **GDSII data type** to map an object to a layer. See GDSII Data Type, below for further information:



Import GDSII Options

☑ For unknown layers (non-defined GDSII layer numbers), place objects with the same GDS layer number but different data types on different layers.

During GDSII import, L-Edit will create an XrefCell for each external cell and will attempt to automatically establish a link by locating the cell using its cell name in each Xref file. The files will be searched in the order that they appear in the **Setup > Design—Xref files** dialog (see Cross Reference File Designation on page 1-151). Once L-Edit finds a matching cell name, the link is established and no further searching is done.

If L-Edit does not find the referenced external cell in any of the cross-referenced TDB libraries, it will create a blank cell and open the **Examine XrefCell Links** dialog to allow you to redirect the missing cell definition at the end of the GDSII import operation (see Examining XrefCells on page 1-401).

GDSII Data Type

If an object has a combination of GDSII layer number and data type that corresponds to an existing layer, L-Edit maps the object to that layer.

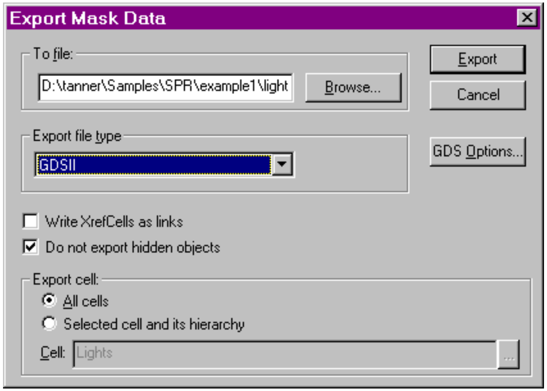
However, if an object has a GDSII number/data type combination that does not correspond to any existing layer, L-Edit will:

- Map the object to the first layer with a matching GDSII number and GDSII data type of 0.
- Map the object to the first layer with a matching GDSII number and unassigned GDSII data type.

If the design file does not contain a layer with a corresponding GDSII number, L-Edit will prompt you to create a new layer or select an existing layer.

Exporting Files

Use **File > Export Mask Data** to export CIF or GDSII files from L-Edit.



To file: D:\tanner\Samples\SPR\example1\light

Export file type: GDSII

□ Write XrefCells as links

☑ Do not export hidden objects

Export cell:

● All cells

○ Selected cell and its hierarchy

Cell: Lights

Options include:

To file	Name of the file to which you want to export mask data.

Export file type

Type of file to be exported. Options include **CIF** or **GDSII**.

Write XRefCells as links

When this option is checked, L-Edit will export XrefCells as cell references only; their contents will not be included. An XrefCell that is not instanced will not be written out. When this option is not checked, L-Edit will write all cell references and their contents to the GDSII file, whether the cell is an XrefCell or not.

Note: only the cell name is exported, not the contents of the cell. All cells that are linked must have a local cell name that is the same as their XrefCell name for GDSII to successfully export when using this option. If a local cell name is not the same as its external cell name when the **Write XrefCells as links** option is on, the GDSII export operation will report an error and abort.

Do not export hidden objects

When this option is checked, L-Edit will not write any objects that are hidden either by object type or by layer. When this option is off, all objects will be written to the GDSII file, whether hidden or not.

Export cell:
All cells

Exports all cells to the GDSII or CIF file.

Export cell:
Specified cell and its hierarchy

Exports just the specified cell and all cells that it depends upon (instanced in that cell and down the cell's hierarchy).

GDS Options
CIF Options

Accesses a dialog for selecting export options for CIF or GDSII files, as appropriate. See CIF Export Options on page 1-194 and GDSII Export Options on page 1-195.

Click the **Export** button.

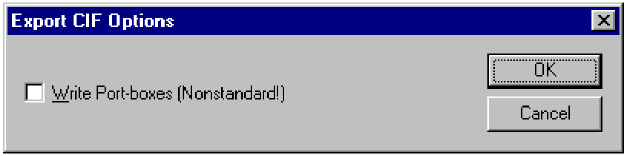
Warning: L-Edit does not create backup files for CIF and GDSII files. If you try to write to an existing CIF or GDSII file, L-Edit will warn that you are about to overwrite a file.

L-Edit produces a log file with detailed warning and error messages after exporting a GDSII file.

L-Edit cannot export layers without legal CIF names, but it does accept and preserve non-standard GDSII numbers when importing and exporting GDSII files. For further information on assigning and propagating GDSII data types, see Assigning Data Types on page 1-215.

CIF Export Options

For CIF files, the export option is **Write port-boxes (Nonstandard)**.



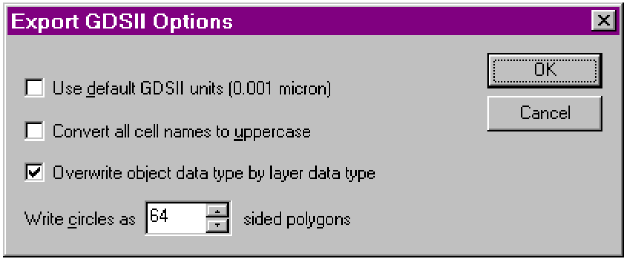
Export CIF Options

□ Write Port-boxes (Nonstandard!)

L-Edit writes out ports using the **.cif** extension, where the port's label is written along with the center of the port box. When this box is checked, L-Edit writes port boxes using the center/length/width syntax of CIF boxes. *This does not conform to standard CIF syntax.* For more information see Extensions on page 1-207.

GDSII Export Options

For GDSII files, the following export options are available:



Export GDSII Options

□ Use default GDSII units (0.001 micron)

□ Convert all cell names to uppercase

☑ Overwrite object data type by layer data type

Write circles as 64 sided polygons

Use default GDS II units When checked, L-Edit converts object dimensions into units of 0.001 micron (the default GDSII *database unit*) when exporting a GDSII file.

For example, a 10×10 box with 1 internal unit = 1 lambda = 1 micron in the L-Edit layout would be recorded in the GDSII file as having dimensions of 10,000×10,000 database units. When this GDSII file is reopened, the technology setup is automatically changed so that the resulting box is 10×10 microns = 10,000×10,000 *internal units* in size.

When the **Use default GDSII units** box is not checked, L-Edit writes object dimensions into the GDSII file as specified, with an appropriate comment on the units being used. No technology setup changes are made when the file is reopened.

Convert all cell names to uppercase

Some GDSII systems do not recognize lowercase letters. When this box is checked, L-Edit changes all letters written to a GDSII file to uppercase.

Overwrite data type on export

When this box is checked, an object will be written to the GDSII file with the data type of the layer on which it is drawn. If the layer does not have a data type, the object retains its data type.

Write circles as … sided polygons	By default, L-Edit circles writes out circles as 64-sided polygons in GDSII format. The number of sides can be changed here.

For further information, see Assigning Data Types on page 1-215.

GDSII Export Restrictions

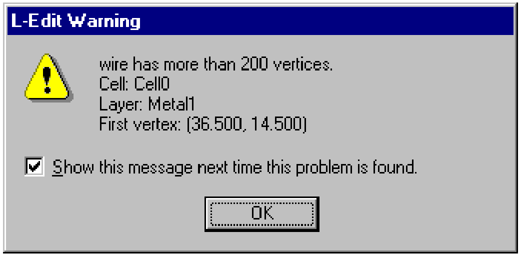
CIF and GDSII files do not contain curves. If your design contains curves they are automatically approximated during export, using the parameters set in the **Setup Design—Curves** dialog. For more information on these parameters, see Curve Approximation Parameters on page 1-148.

Note: CIF format files can support CIF circles. Circles are not automatically approximated during the export process.

If you are exporting a GDSII file, L-Edit assigns a number to each layer in the design in order to conform to GDSII syntax. To modify a GDSII layer number prior to exporting the file, use **Setup > Layers**—**General**. Select a layer in the **Layers** list and enter the appropriate value in the **GDSII number** field.

Note that L-Edit will accept and preserve non-standard GDSII numbers when importing and exporting GDSII files.

Wires and polygons in GDSII files cannot contain more than 200 vertices. If your design contains a wire or polygon with more than 200 vertices, a warning appears.



wire has more than 200 vertices.
Cell: Cell0
Layer: Metal1
First vertex: (36.500, 14.500)

☑ Show this message next time this problem is found.

CIF Files

Caltech Intermediate Form (CIF) is an ASCII file format for the interchange of mask geometry information among IC designers and foundries. CIF is defined in *Introduction to VLSI Systems* by Mead and Conway (Addison-Wesley, 1980). CIF files are typically saved with the **.cif** extension.

A CIF file may contain a single design or a library of designs. CIF assumes a right-handed geometry, with the *x*-axis increasing to the right and the *y*-axis increasing upward. The basic unit of measurement is 0.01 micron.

Commands may be used to scale object sizes, use different layers, and change the placement of objects. Comments may be added to a CIF file by enclosing them in parentheses. All CIF commands and comments must be terminated with semicolons.

Symbols

CIF symbols are defined with the **DS** and **DF** commands. **DS** begins a symbol definition:

```
DS nnn a b;
```

where ***nnn*** is the symbol number and ***a*** and ***b*** are the (optional) scaling factors. All commands that follow the **DS** command and precede the **DF** command are included in the symbol. CIF symbols are always given numeric names.

The optional scaling factors ***a*** and ***b*** are applied to the integer coordinates and distances within a symbol by multiplying each value by ***a*** and then dividing the result by ***b***. Scaling helps to shorten the length of CIF files by eliminating trailing zeros. By default, coordinates and distances in CIF are specified in units of 0.01 micron; **a = 100** and **b = 1** would allow values to be specified in microns instead. The coordinates (10,6) with **a = 100** and **b = 1**, for example, are equivalent to (1000,600) with **a = 1** and **b = 1**. If ***a*** and ***b*** are not specified, then they are both assumed to be 1, and all integers are mapped to the 0.01 micron standard.

The **DF** command ends the last open **DS** command:

```
DF;    (end of symbol definition);
```

If no symbol is open when a **DF** command is encountered, then a warning message is generated.

Symbols may be instanced within other symbols and are functionally equivalent to L-Edit cells.

Calls (Instances)

Once a symbol is defined, it may be instanced with the **C** (call) command. In addition to instancing the named symbol, the **C** command also permits a variety of optional transformations to be applied:

```
C integer transformation;
```

where ***integer*** is the number of the symbol being called and ***transformation*** is an optional transformation. A transformation may be composed of several translations, mirrors, or rotations. Combinations of transformation operations are unambiguously applied from left to right as they are encountered within the command. Great care should be exercised when determining the order of transformation operations since the commutative property does not hold.

The *translation* operation specifies a coordinate. The coordinate represents the endpoint of a vector originating at (0,0). For example:

```
C 55 T -100,10;     (call command with translation);
```

calls symbol 55 and translates it 100 units in the negative *x* direction and 10 units in the positive *y* direction.

The *mirroring* operations, **MX** and **MY**, correspond to multiplying the *x* and *y* coordinates by –1, respectively. For example:

```
C 99 MX;      (call symbol 99 and flip horizontally);
```

```
C 22 MY;      (call symbol 22 and flip vertically);
```

The *rotation* operation rotates the called symbol in the specified direction. Direction is indicated by a *direction vector*: a coordinate whose vector from the origin (0,0) sets the angle to which the symbol's *x*-axis is rotated. Only the direction of the vector is significant; the magnitude is ignored. For example:

```
C 44 R 0,1;   (call command with rotation);
```

calls symbol 44 and rotates its *x*-axis by 90°.

Geometric Primitives

CIF provides commands for creating four types of geometric primitives: boxes, polygons, roundflashes (circles), and wires.

The **B** (box) command defines a rectangular box of fixed length and width. The center coordinates locate the box, and a direction vector indicates its orientation. For example:

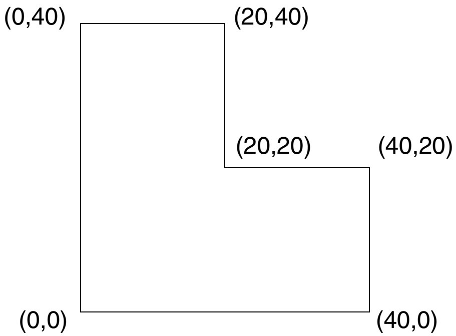
```
B 25 60 80,40 -20,20;      (box command);
```

describes a box of length 25 and width 60, with center at (80,40) and direction vector (–20,20). The length of the box is parallel to the direction vector, and its width is perpendicular to the direction.

The **P** (polygon) command defines a polygon with a certain number of sides and vertices. **P** accepts a path of coordinates and creates the enclosed polygonal region in the order in which the vertices are specified (the edge connecting the last vertex with the first is implied). For example:

```
P 0,0 0,40 20,40 20,20 40,20 40,0;
```

describes an L-shaped polygon with vertices at (0,0), (0,40), (20,40), (20,20), (40,20), and (40,0).

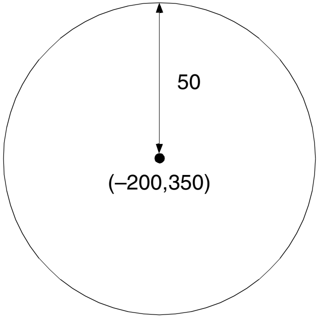


To convert rectangular polygons to boxes when reading CIF into L-Edit, you must check the **Read rectangular polygons as boxes** option in the **Import CIF Options** dialog. To access this dialog, choose **File > Import Mask Data** and click **Options**.

The **R** (roundflash) command defines a roundflash (circle) of fixed diameter and position. For example:

```
R 100 -200,350;     (roundflash command);
```

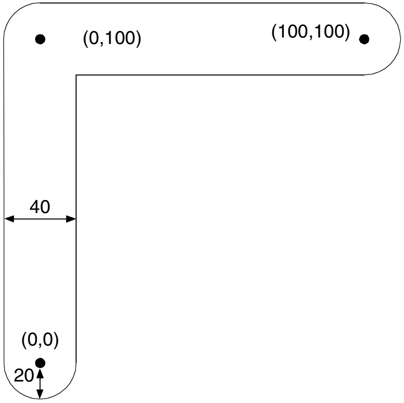
describes a circle of diameter 100 with center at (–200,350).



The **W** (wire) command defines a wire with fixed width along a specified path. A wire can be described as a long run of uniform width; ideally, the locus of points within one-half width of the given centerline or path and one-half width of the endpoints (semicircular caps). For example:

```
W 40 0,0 0,100 100,100;    (wire command);
```

describes a wire of width 40 with centerline vertices at (0,0), (0,100), and (100,100).



Layers

All primitive geometry elements must be associated with a particular fabrication mask or technology layer. Layers are specified with the **L** (layer) command. Primitives created after an **L** command belong to that layer until the layer is reset by the next **L** command. The form of the **L** command is:

```
L shortname;  (layer command);
```

where ***shortname*** is the 1–4 character layer name. Layer names must be unique and correspond to fabrication masks being constructed. You should therefore take care that the layer names you use accord with the conventions established by your fabricator. The **General** tab of the **Setup Layers** dialog correlates CIF layer names and technology layers; the CIF names are used instead of the L-Edit layer names during the conversion of the design file into CIF. Layer names that do not conform to legal CIF syntax must be modified before saving. Layer name specifications are preserved across symbol calls.

Layer names in the setup file must agree with the layer names of CIF files read in; otherwise, the geometry information on the non-matching layers in the CIF file will be transferred to the Icon layer. Your fabricator may apply additional restrictions and extensions to the CIF standard.

Fabrication Cell

One piece of information which must be supplied to your fabricator is the name of the cell which represents the top level of your design. The fabricator will typically choose the top-level cell in your design, if it is the only such cell. However, if you do not specify this information and your fabricator has a choice about which cell to fabricate, the wrong one might be chosen.

L-Edit does not accept geometry other than CIF symbols. A CIF call (instance) to the top level of a design is achieved by choosing **Cell > Fabricate**. **Fabricate** causes a CIF **C** command (or call to the selected cell) to be created at the top level, effectively identifying that cell as the cell to be fabricated. L-Edit only allows a single call outside of a symbol definition. If any rotations or transformations are embedded in this outside call, L-Edit suppresses them when the file is read.

Warning: Once a fabricate cell has been chosen, it will remain the fabricate cell until a new one is chosen, even if it ceases to be the top-level cell in your design. Be sure to check the fabricate cell before writing a CIF file!

Restrictions

L-Edit accepts forward references (symbol calls before the symbol definitions they reference), but removes them during conversion of the design into CIF.

L-Edit does not support the CIF **DD** (delete symbol definition) command.

Extensions

L-Edit supports two user extensions to the basic CIF syntax. The first extension is a cell name extension of the form:

```
9 cellname;
```

where ***cellname*** is the name of the currently open CIF symbol. This command can only appear within the context of an open symbol (between a **DS/DF** command pair). The cell name may contain spaces and must be terminated with a semicolon. Duplicate, zero-length, and null cell names are not permitted.

If a CIF file does not define cell names for CIF symbols, then L-Edit automatically assigns as the cell name the expression:

```
(DS nnn)
```

where ***nnn*** is the CIF symbol number. This definition is suppressed when the CIF file is written out. You should therefore avoid naming cells with this syntax, or else the name will be suppressed during CIF file conversion. L-Edit reads out-of-order cell numbers, but always orders cells by number while writing out the design in CIF.

The second user extension is a port extension of the form:

```
94 portname width height center_x center_y layer;
```

where ***portname*** is the name of the port (label), ***x*** and ***y*** are the coordinates of the port, and **layer** is the name of the port's layer. This is a relatively standard port or label user extension to CIF. However, it is not as flexible as L-Edit's definition of a port. An L-Edit port can be a point, a line, or a box, and the text can be rotated in a variety of ways; this CIF user extension can only represent a single point, with no information on the position or rotation of the associated text. When L-Edit writes a port into a CIF file, it computes the centerpoint of the port and records this in the CIF file as the position of the port. You can preserve the box associated with the port in CIF as written by L-Edit by checking the **Write**

Port-boxes option in the dialog opened by pressing the **Options** button in the **Export Mask Data** (choose **File > Export Mask Data** to call forth this dialog). This results in the use of nonstandard notation for ports, and other software tools may not be able to read this form of CIF.

Wires

CIF was developed at a time when masks were usually created by Gerber photoplotters. Such plotters could make wires by opening a circular aperture and moving it along a pathway. The resulting wire would therefore have rounded corners and ends. This fabrication method gave rise to the CIF specification for rounded wires. However, present-day mask making is almost entirely raster-based, and thus has a strong affinity toward orthogonal structures. So many fabricators assume CIF wires to have extended wire end styles with mitered corners. Thus to adhere to the fabricators' implementation of wires, all your CIF wires should be of extend end style and layout join style. Upon fabrication, many fabricators such as MOSIS and Orbit run both CIF and GDSII files through CATS (a high-end program used by many fabricators and mask houses to produce formats for specific mask-making equipment from GDSII and CIF layout files). CATS uses its own clipping algorithm for acute angle CIF wires and GDSII paths with a pathtype of 0 or 2. This algorithm corresponds exactly with the L-Edit wire layout join style (the default) which employs a miter length of one-half the width of the wire for wires with an acute join angle. You should check with your fabricator concerning the exact method of fabrication used for wires before using wires in your layout.

Scaling

Apart from the user-selectable scaling of L-Edit internal units, L-Edit incorporates an implicit scaling factor while writing CIF files. Due to the manner in which geometric objects are represented in CIF, it is necessary for L-Edit to apply an implicit multiplication factor of two to all geometry as it is written out to CIF. The reason for this scaling is that CIF represents boxes with integer length, width, and center coordinates. L-Edit, however, can create boxes with fractional center coordinates: a box of width and length 3 with lower left corner at (0,0) has its center at (1.5,1.5), for example. L-Edit circumvents this problem by multiplying all coordinates by two when writing a CIF file. The same box, after being written out to a CIF file, would have a length and width of 6 and be centered at (3,3). L-Edit incorporates this multiplication by 2 into the scaling factors recorded in the CIF file, so that when the file is read in by a CIF reader it is scaled correctly.

GDSII Files

GDSII stream format is a binary file format for interchanging mask geometry information between different IC CAD systems. The L-Edit implementation of GDSII file reading and writing conforms to the Calma Stream Format, GDSII release 6.0, with some limitations. GDSII files are typically saved with the **.gds** extension.

A GDSII file may contain a single design or a library of designs. GDSII assumes right-handed geometry, with the *x*-axis increasing to the right and the *y*-axis increasing upward. The basic unit is set to the GDSII default (user unit = 1 micron and 1000 database units per user unit).

Most L-Edit elements have a one-to-one correspondence with elements of GDSII stream files. GDSII last access time information is not supported by L-Edit. L-Edit circles are approximated by GDSII polygons. L-Edit cell names may be modified on export to GDSII.

Note: There is a UPI macro that allows you to assign GDSII properties to objects; see the upilib.wri file in the upilib folder of your L-Edit installation directory for documentation.

The table below shows the correspondence between L-Edit elements and their GDSII names. GDSII data types for L-Edit boxes, wires, and polygons can be viewed and edited in the **Edit Object(s)** dialog with **Edit > Edit Object(s)**.

L-Edit	*GDSII*
File	Stream file
Cell Definition	Structure
Box	Boundary *
Box **	Box
Polygon	Boundary
Wire	Path
Circle	Boundary ***
Instance	SRef
Array	ARef
Port	Text
Data type	Data type

* L-Edit boxes are written to GDSII files as 4-sided boundaries (polygons). When reading boundaries from a GDSII file, L-Edit checks each one to see if it is a 4-sided orthogonal polygon, and if so, represents it as an L-Edit box.

** GDSII boxes are not intended to be mask geometry and are generally discarded by mask-making software. If L-Edit encounters GDSII boxes while reading a GDSII file, a dialog is presented with two options: discard all GDSII boxes or convert them to L-Edit boxes (mask geometry).

*** L-Edit circles are written by default as 64-sided polygons.

GDSII allows only the following restricted set of characters in cell names. “a” … “z”, “A” … “Z”, “0” … “9”, underscore “_”, question mark “?”, and dollar sign “$”. L-Edit cell names may include a fuller set of characters, some of which would be illegal in GDSII. Therefore, L-Edit checks each cell name before writing it out to a GDSII file. If any spaces “ ” are found, L-Edit replaces them with underscores “_” in the GDSII file. If any other illegal characters are found, L-Edit requests that you change the cell name.

Some GDSII systems do not recognize lower case letters in cell names. For interfacing with these systems, L-Edit provides the capability to write all cell names to a GDSII file in upper case. This option is enabled by a check box in the dialog opened by pressing the **Options** button in the **Export Mask Data** (choose **File > Export Mask Data** to access this dialog).

GDSII Date Formats

The GDSII format allows for the year to be stored in one of three formats:

- current year (e.g. 103 representing the year 2003)
- full representation (e.g., 1999)

- last two digits of the year (e.g., 32 representing the year 2032 or 1932)

When a year is read from a GDSII file, it may need to be modified to represent the correct year, depending on which date format is used. The current year format is the default. If the last two digits of the year is detected during GDSII import, L-Edit will use the algorithm shown below to modify the date that was read.

- for years less than 60, add 2000 to the year
- for years greater than or equal to 60 or less than or equal to 1960, add 1900 to the year
- for years greater than 60, do not modify the year

This approach will handle all three date formats until the year 2060.

GDSII Shape Definition

GDSII does not contain a specification for circles. Therefore, L-Edit approximates circles using 64-sided polygons. Thus, circles are not preserved when a GDSII file is written and read back in.

L-Edit supports all-angle rotations of instances (in a precision up to .01 degree; fractional angles are rounded without warning) and 90° rotations of text.

L-Edit treats four-sided polygons as boxes. If you export, then reimport, a design that contains four-sided polygons that are orthogonally oriented rectangles, L-Edit will convert them into boxes. For the purposes of fabrication, there may be no difference between a box and its equivalent polygon.

Many different versions of GDSII readers and writers exist. Some newer versions produce elements which are not compatible with older versions of GDSII. The elements in L-Edit are confined to elements which are common to all.

GDSII Data Type

GDSII layers are identified by the **GDSII number** assigned to that layer or, alternately, by the combination of the assigned **GDSII number** and **GDSII data type**. You can use this data type in conjunction with the **GDSII layer number** to overcome the 64-layer limitation in the GDSII database.

The GDSII specification indicates that the GDSII layer number and GDSII data type should have a range of 0 to 255. However, L-Edit supports non-standard GDSII layer number and GDSII data type values in the range of -32,768 to 32,767 for compatibility with other tools that are able to output numbers outside the 0-255 range. L-Edit will write a warning to the log file during import and export indicating that the GDSII file that was read or written does not adhere to the GDSII Stream Specification.

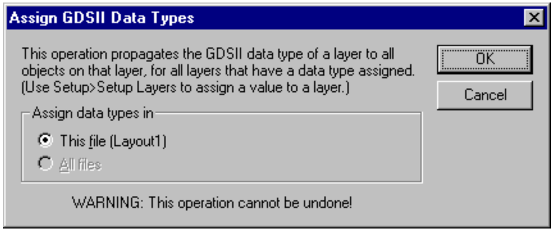
Assigning Data Types

Use **Setup > Layers—General** to assign a data type to a layer or **Edit > Edit Object** to assign a data type to an object.

When you assign a GDSII data type to a layer, all objects subsequently drawn on that layer will acquire that data type. If you subsequently change the data type for a layer, however, the new data type will only be applied to objects drawn after the change.

If you merge intersecting objects with different GDSII data types, L-Edit replaces their respective data type values with the data type for the layer (or 0 if no data type is set for the layer) without a warning.

To propagate a layer's data type value to all objects currently drawn on that layer, use **Draw > Assign GDSII Data Types,** shown below.


Assign GDSII Data Types

This operation propagates the GDSII data type of a layer to all objects on that layer, for all layers that have a data type assigned. (Use Setup>Setup Layers to assign a value to a layer.)

Assign data types in

● This file (Layout1)

○ All files

WARNING: This operation cannot be undone!

This command propagates the data type for a layer to all objects on that layer, overwriting any current values, for all layers in a design. You can perform this operation for the active file or all open files.

Wires

The GDSII layout format allows for three different types of wires (paths), each with a unique pathtype value:

- pathtype 0: butt ends and square corners (corresponds to L-Edit Round end style with Round join style)
- pathtype 1: round ends and round corners (corresponds to L-Edit Butt end style with Layout join style)
- pathtype 2: extended ends and square corners (corresponds to L-Edit Extend end style with Layout join style)

These GDSII pathtypes correspond directly to three of the twelve possible L-Edit wires. When reading GDSII files, L-Edit sets wire end styles and join styles to match the three GDSII pathtypes. When creating GDSII output, L-Edit assigns a GDSII pathtype according to the following table. When the end styles and join styles do not correspond exactly to a GDSII pathype (indcated in the table with an asterisk), L-Edit will provide a warning message.

End style	*Join style*	*GDS II pathtype*
Butt	**Layout**	0
Butt	Miter	0 *
Butt	Round	0 *

End style	*Join style*	*GDS II pathtype*
Butt	Bevel	0 *
Round	Layout	1 *
Round	Miter	1 *
Round	**Round**	1
Round	Bevel	1 *
Extend	**Layout**	2
Extend	Miter	2 *
Extend	Round	2 *
Extend	Bevel	2 *

Many fabricators such as MOSIS and Orbit run GDSII files through CATS (a high-end program used to produce formats for specific mask-making equipment from GDSII layout files). CATS uses its own clipping algorithm for acute angle GDSII paths with a pathtype of 0 or 2. This algorithm corresponds exactly to the L-Edit layout wire join style, the default wire join style. Layout join style employs a fixed miter length of one-half the width of the wire for wires with an acute join angle.

When you are about to use wires for the first time or you are setting up the technology files for others who may use wires, take a moment to set up the wire defaults for each layer according to whether your likely output format will be GDSII. For GDSII, use one of the three legitimate combinations of end style and join style. It is also strongly recommended that you contact your fabricator before you define the wire styles for your design and understand how they will interpret GDSII wires.

Ports

L-Edit uses the default port text size as a reference during import and export. On export, L-Edit calculates the ratio between the default text size and a port's actual text size and writes that value to the GDSII file. On import, it determines the absolute text size for a given port by multiplying the default port text size by the magnification factor in the GDSII file.

To change the text size of ports in exported GDSII files, decrease or increase the default port text size before exporting the design to a GDSII file. You can change the default port text size by choosing **Setup > Design—Drawing** (see Drawing Parameters on page 1-145).

To change the text size of ports in a design imported from GDSII, modify the default port text size in the setup file before importing the design from GDSII.

Note: There is a UPI macro that will scale all port's text size; see the upilib.wri file in the upilib folder of your L-Edit installation directory for documentation.