System Overview

GenFlex™

A dedicated CAM system for flex and rigid-flex manufacturing

Frontline
PCB SOLUTIONS
An Orbotech Valor Company

GenFlex System Overview

Published October, 2008

Frontline PCB Solutions
4 Faran St., Ashtrum Park-Northern Industrial Area
P.O. Box 680, 81106 Yavne, Israel
Tel: +972 8 9322183, Fax: +972 8 9322186
Email: info@frontline-pcb.com
Web: http://www.frontline-pcb.com

OVERVIEW

WELCOME TO GENFLEX

GenFlex is a feature-rich, powerful set of tools for preparing flexible Printed Circuit Board (PCB) data for effective manufacturing.

GenFlex is a CAM system for the pre-production engineer at a bareboard fabrication plant. It is the complete solution for data reception and preparation, from customer files to the production floor.

Frontline PCB Solutions philosophy, as implemented in its software products, advocates seamless integration between design and manufacturing. This is reflected in the ODB++ database, which can be output directly from most common EDA CAD systems. The database and user interface modules, using a common language, break the barriers that have existed for years between design and manufacturing.

Learning the GenFlex system with all its capabilities is an ongoing process. The system is composed of modules, some of which are basic and straightforward while others require knowledge of various manufacturing processes. The learning process may be performed concurrently with system operation, together producing tangible material cost benefits.

The Frontline policy of open architecture provides tools that are highly customizable by system administrators and advanced users. New modules are being introduced on an on-going basis and provide additional opportunities for increased efficiency and productivity.

GENFLEX SYSTEM FUNCTIONS

The functions of the GenFlex system can be categorized as follows:

Input

The input stage imports data from multiple sources/formats into a common job “container” in the system database. Input complexity depends on the source CAD system. GenFlex can import directly from system databases, through ODB++, or from low level formats (such as Gerber, Excellon, DXF, etc.).

Cleanup and Analysis

The cleanup process prepares imported data for effective analysis. The level of cleanup depends on the source system. The analysis processes that are performed after cleanup verify the correctness and the manufacturability of the design. This stage can be customized to a number of depth levels, depending on the specific requirements of the designer/manufacturer.

Editing and Optimization

GenFlex provides an extensive set of tools for automatic/manual editing and optimization of design data. These tools provide single board editing, panelization, and manufacturing tool preparation.

Output

Once the job is complete and ready for production, the system provides a large set of tools for sophisticated output file generation. The output tools generate data for bareboard fabrication (artwork files, drill files and netlists), as well as data for board assembly and drawings.

AOI

GenFlex can interface to a variety of Automated Optical Inspection (AOI) systems, to aid the user in defining parameters for the physical inspection of PCBs. GenFlex can interface with several AOI systems manufactured by Orbotech, including the Discovery, PC-14, Inspire, Infinex, and Vision AOI systems.

Electrical Testing

GenFlex includes an Electrical Testing Manager module (ETM), an open-ended ET software solution based on CAM techniques. The GenFlex ETM maintains the design intelligence of the board, including netlist and hole-plating information, thus ensuring the integrity of the customer's board design, and provides outputs for grid testers and flying probe testers as well as to generic output formats.

Additional Functions

In addition to the functions described above, the system provides a rich set of tools for organizing and monitoring data processing. Work Forms provides a unique and customizable tool for data maintenance.

All aspects of the system lend themselves to automation. Implementing automation is crucial to achieving a competitive edge in the market. Utilizing standard programming languages **`(csh, sh, perl, Tcl/Tk and C)`** in conjunction with GenFlex commands and routines, users can streamline their operation into an automatic and cost-effective process.

The rest of this book covers the subjects described in this introduction in more detail utilizing pictures and examples. For a new user of the system, this book is an introduction to a whole new world of capabilities and discoveries.

Sit back, read and enjoy!

WHO IS THIS BOOK FOR?

The System Overview book (Doc. 0101) is intended for first time users of GenFlex. It is recommended as the first book to be read by a user of the system. Knowledge of basic PCB terms is assumed, although the book provides explanations for some of the terms. Knowledge of the process of releasing data to manufacturing, and the manufacturing processes themselves, is recommended, although not necessary. This book has no prerequisites in the documentation library. It contains many references to additional books in the library for extended reading.

ORGANIZATION OF THIS MANUAL

CHAPTER 1

is a general overview of the concepts covered by the book as well as its scope and structure.

CHAPTER 2

describes the basic system functions at an introductory level. Pictorials are provided for better visualization of concepts. The goal of this chapter is to describe the capabilities of the system rather than providing detailed information on how to operate the system.

CHAPTER 3

is intended for the system administrator or implementor. It describes "what is available" rather than "how to" (as described in the corresponding manuals).

CHAPTER 4

provides guidelines for reader as to recommended reading and further actions that can be taken.

APPENDIX A

shows a typical flow chart of the system.

APPENDIX B

list frequently asked questions by readers.

APPENDIX C

describes how error messages are presented.

APPENDIX D

provides the system administrator with a description of some of the advanced functions of the system.

Basic System Functions

THE ENGINEERING TOOLKIT WINDOW

The Engineering Toolkit window enables the management of jobs in the Genesis system. A job contains all the data related to a single PCB design.

Structure of a GenFlex Job

The job structure design goals, as implemented in the system, provide the following qualities:

SELF-CONTAINED

a job contains all the information needed for correct representation of the design. In other words, any references it maintains to external libraries are not required in order to view its data correctly.

PORTABLE

a job directory can be shared between GenFlex systems installed on different hardware and operating system platforms, loaded on the destination system and worked upon without changes.

COMPLETE

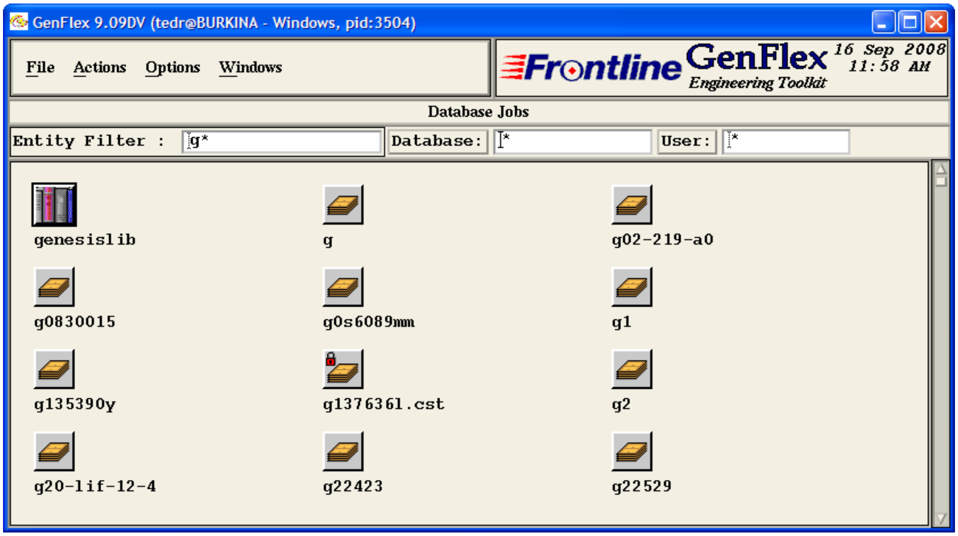
a job contains a comprehensive collection of data representing all aspects of its manufacturability. This includes graphical information, layer structure, panelization information, results of analysis in the form of reports, various attributes attached to features and layers, instructions and images inside work forms, annotations, and much more.

OPEN

The Genesis database is composed of a collection of all the jobs in the system plus a special job, the central library job. The database can be physically distributed across networks, yet remain logically centralized. Users can view all the jobs in the database whether on the local disk or elsewhere on the network.

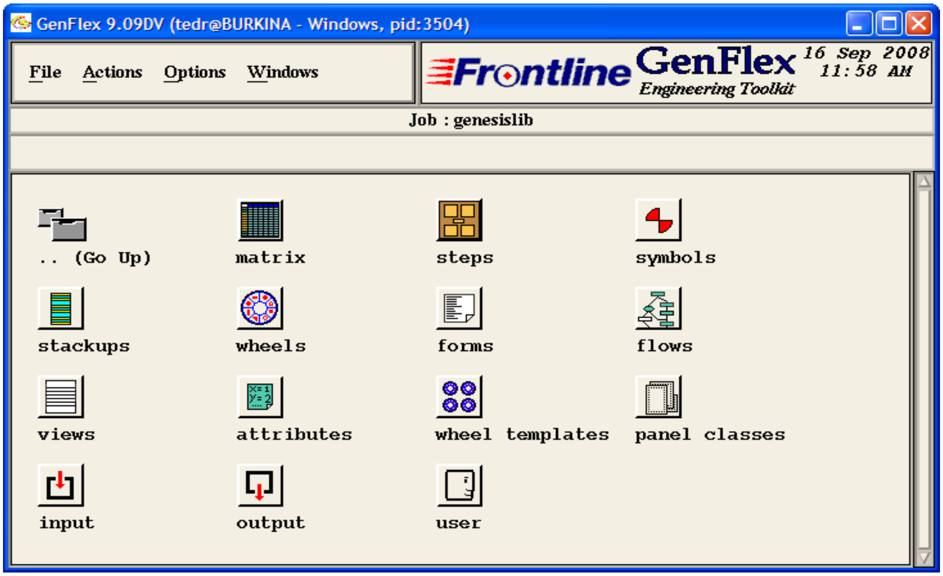
The following capabilities are provided by the Engineering Toolkit window:

1 **Viewing the job database by selecting icons in a graphical interface.**

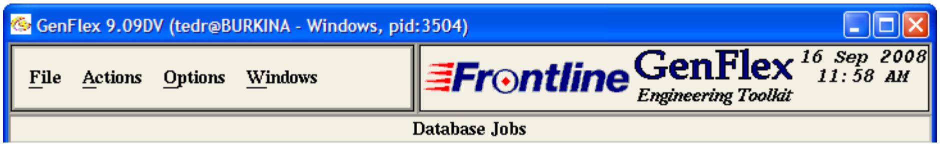


Entity Filter : g*
Database: *
User: *

2 **Navigating within a job, viewing different job entities by clicking their icon.**

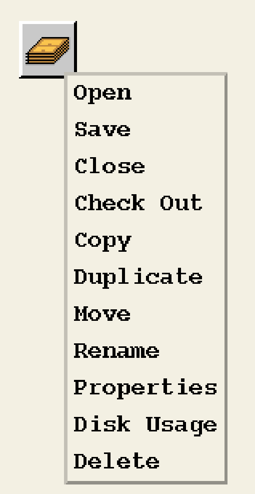


3 **Performing job housekeeping functions such as copying, renaming, deletion.**

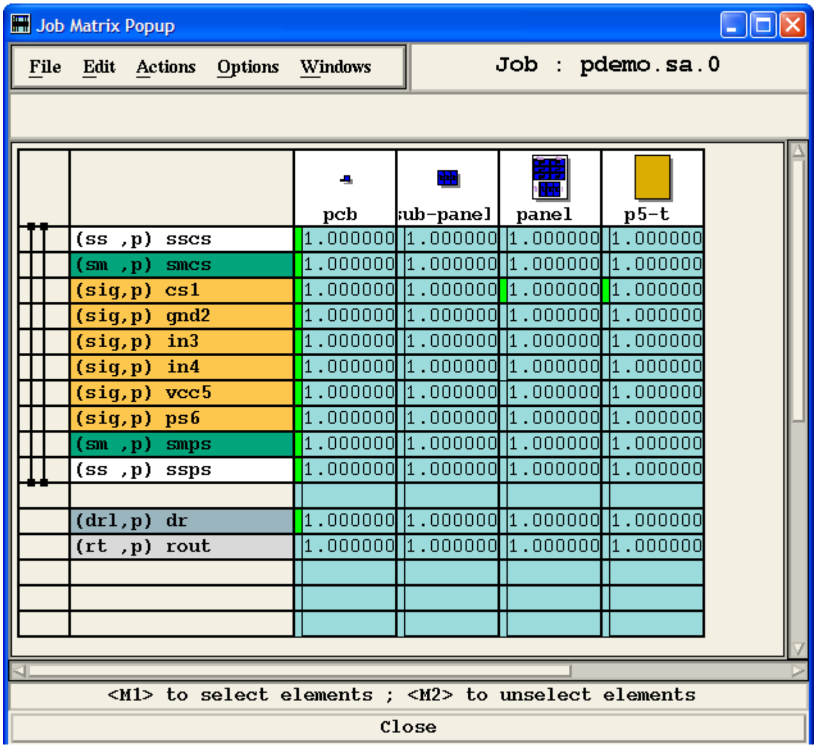


Main Menu Bar of the Engineering Toolkit Window

4 **Other housekeeping functions and job properties are made available by clicking M3 (right-click) on a job icon.**



5 **Viewing and editing layer and panel information in the Job Matrix, the main job structure display. This window controls both the layer structure of the job and step & repeat (panelization) information.**



Job : pdemo.sa.0

	pcb	;ub-panel	panel	p5-t
(ss ,p) sscs	1.000000	1.000000	1.000000	1.000000
(sm ,p) smcs	1.000000	1.000000	1.000000	1.000000
(sig,p) cs1	1.000000	1.000000	1.000000	1.000000
(sig,p) gnd2	1.000000	1.000000	1.000000	1.000000
(sig,p) in3	1.000000	1.000000	1.000000	1.000000
(sig,p) in4	1.000000	1.000000	1.000000	1.000000
(sig,p) vcc5	1.000000	1.000000	1.000000	1.000000
(sig,p) ps6	1.000000	1.000000	1.000000	1.000000
(sm ,p) smps	1.000000	1.000000	1.000000	1.000000
(ss ,p) ssps	1.000000	1.000000	1.000000	1.000000
(drl,p) dr	1.000000	1.000000	1.000000	1.000000
(rt ,p) rout	1.000000	1.000000	1.000000	1.000000

<M1> to select elements ; <M2> to unselect elements

Job Matrix

Experience with the Engineering Toolkit is mandatory for any user of Genesis. See Doc. 0102, The Engineering Toolkit, “Overview” for a more detailed description.

ODB++ Data Exchange

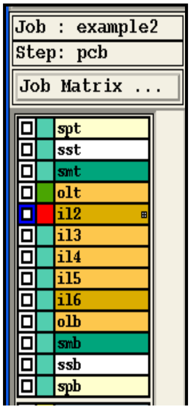
Genesis can interface directly to the intelligent database of an EDA layout system, providing the most efficient method of importing data. GenFlex imports this intelligence through the ODB++ Data Exchange Format.

The main advantage of a direct EDA interface is its ability to capture the intelligence of the design. Other input methods rely on lower level formats which inherently miss this intelligence. Capturing design details such as net names, component names and other attributes, contributes greatly to accurate analysis. In addition, the data read directly from the design is far more optimized than that read from low level formats.

Genesis inputs Gerber and drill files in low level format. Though less desirable than direct EDA interface, Gerber and drill input still facilitate PCB pre-production and many other aspects of manufacturability analysis.

The information read from any EDA system falls into the following categories:

- **Layer structure of the board, including copper layers, mask layers, drill and rout layers, and miscellaneous drawing layers.**

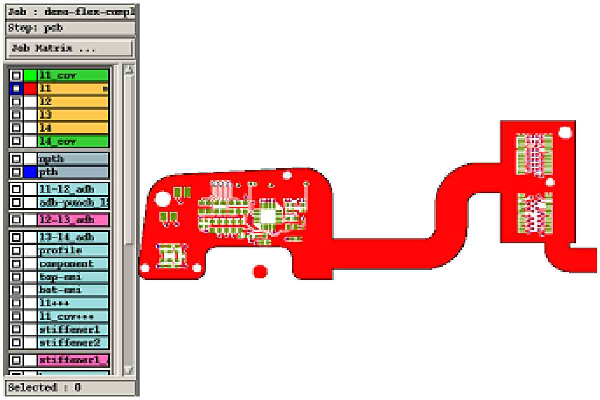


Job : example2
Step: pcb
Job Matrix ...

□ spt
□ sst
□ smt
□ olt
□ il2
□ il3
□ il4
□ il5
□ il6
□ olb
□ smb
□ ssb
□ spb

Layer List in the Graphic Editor

- **Graphical data to be placed in board layers.**



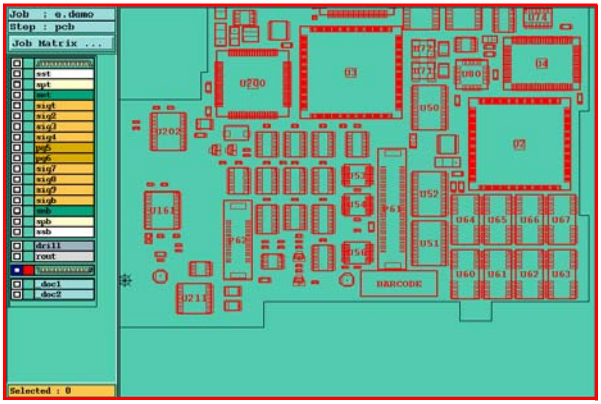
Job : demo-flex-compl
Step: pcb
Job Matrix ...

□ l1_cov
□ l1
□ l2
□ l3
□ l4
□ l4_cov
□ npth
□ pth
□ l1-l2_adh
□ adh-punch_l
□ l2-l3_adh
□ l3-l4_adh
□ profile
□ component
□ top-emi
□ bot-emi
□ l1+++
□ l1_cov+++
□ stiffener1
□ stiffener2
□ stiffener1_

Selected : 0

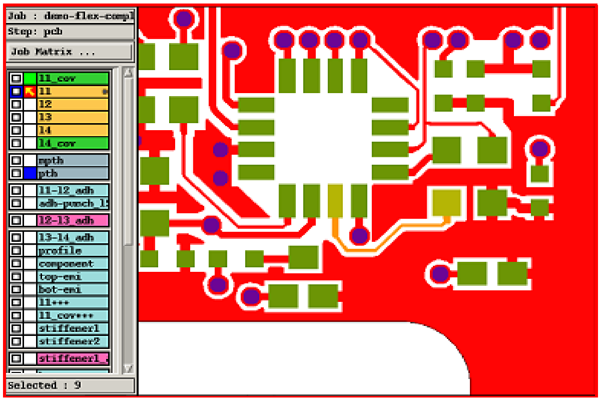
Signal Layer displayed in the Graphic Editor

- **Component information, including part and package names, pin information, outline, location and rotation.**



Component Layer in the Graphic Editor

- **Netlist data, providing electrical connectivity information between physical points in the design.**



Connection Points of a Net

- **Textual attributes attached to components and nets.**

AUTO INPUT

Auto Input is a process by which a job can be constructed from a collection of files in low level format. This type of operation is still the most common way of transferring data from design to manufacturing.

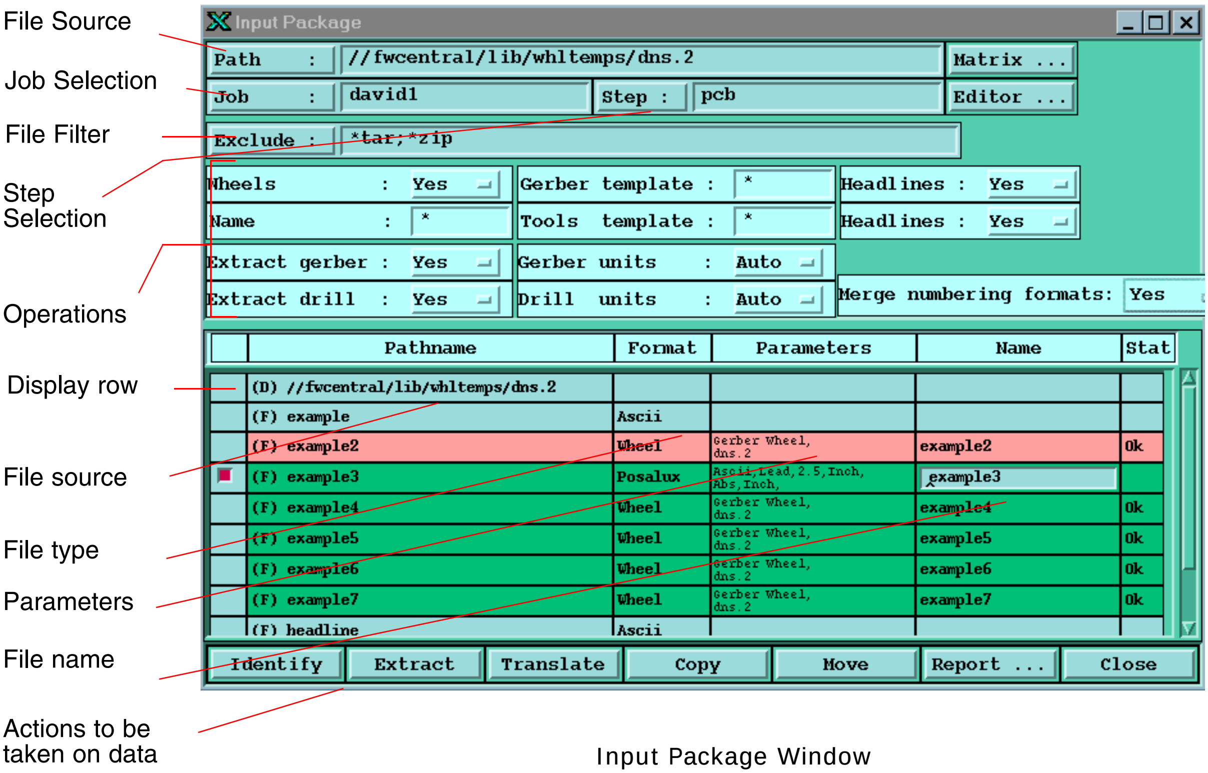
Genflex includes many highly reliable and field-proven translators for most industry-standard formats. All translators generate data in ODB++

format inside a job in the system's memory database. Once the relevant data is loaded into memory, the job can be saved on disk as a permanent entity.

Input formats vary widely in their level of completeness. Some formats, such as DPF and RS274X, contain in one file all the information required for translation, making the translation a simple, one-step operation. Other formats, including the most common Gerber and Excellon formats, are not as complete and rely on external files to define aperture shapes and other parameters.

Auto Input makes the process of inputting files as simple and painless as possible. Yet it is powerful enough to provide the experienced user with the flexibility to handle wide variations in incoming data.

At the heart of the input operation lies the Input Package Window - a window which lists each file contained in a given directory and its sub-directories, automatically identifying the parameters and type of file.



Path : //fwcentral/lib/whltemps/dns.2

Job : david1   Step : pcb

Exclude : *tar;*zip

Wheels : Yes   Gerber template : *   Headlines : Yes

Name : *   Tools template : *   Headlines : Yes

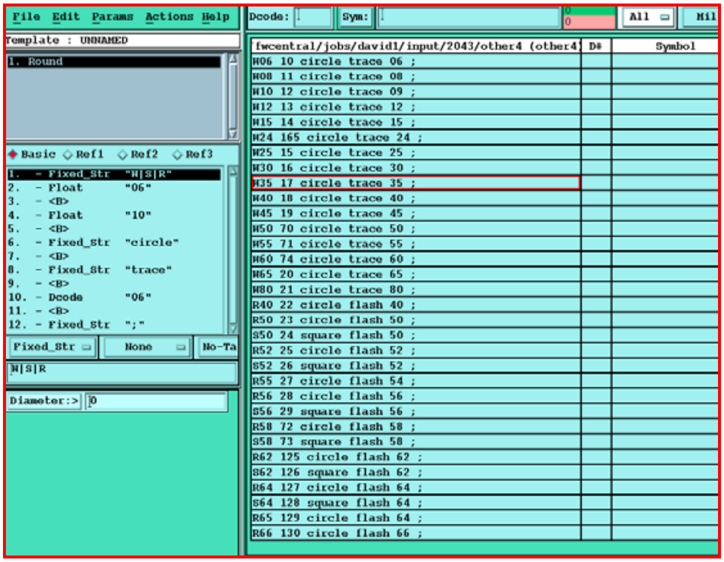
Extract gerber : Yes   Gerber units : Auto

Extract drill : Yes   Drill units : Auto   Merge numbering formats: Yes

	Pathname	Format	Parameters	Name	Stat
	(D) //fwcentral/lib/whltemps/dns.2				
	(F) example	Ascii			
	(F) example2	Wheel	Gerber Wheel, dns.2	example2	Ok
■	(F) example3	Posalux	Ascii,Lead,2.5,Inch, Abs,Inch,	example3	
	(F) example4	Wheel	Gerber Wheel, dns.2	example4	Ok
	(F) example5	Wheel	Gerber Wheel, dns.2	example5	Ok
	(F) example6	Wheel	Gerber Wheel, dns.2	example6	Ok
	(F) example7	Wheel	Gerber Wheel, dns.2	example7	Ok
	(F) headline	Ascii			

Input Package Window

Some files can be easily identified, as their format is predictable and non-ambiguous (e.g. DPF, DXF, RS274X, HPGL). Other files present a tougher challenge. Genesis can handle all of them. An example is the wheel file, which lists the symbols that match the various aperture positions (Dcodes) inside Gerber files. Identifying, and more importantly, interpreting wheel files, requires the preparation of "templates" - files which describe the structure of the wheel files. The Auto Input module includes a highly sophisticated Wheel Template Editor which creates templates using a follow-the-example procedure.



Template : UNNAMED

```
1. Round
```

```
1.   - Fixed_Str   "W|S|R"
2.   - Float       "06"
3.   - <B>
4.   - Float       "10"
5.   - <B>
6.   - Fixed_Str   "circle"
7.   - <B>
8.   - Fixed_Str   "trace"
9.   - <B>
10.  - Dcode       "06"
11.  - <B>
12.  - Fixed_Str   ";"
```

W|S|R

Diameter:> 0

fwcentral/jobs/david1/input/2043/other4 (other4)	D#	Symbol
W06 10 circle trace 06 ;		
W08 11 circle trace 08 ;		
W10 12 circle trace 09 ;		
W12 13 circle trace 12 ;		
W15 14 circle trace 15 ;		
W24 165 circle trace 24 ;		
W25 15 circle trace 25 ;		
W30 16 circle trace 30 ;		
W35 17 circle trace 35 ;		
W40 18 circle trace 40 ;		
W45 19 circle trace 45 ;		
W50 70 circle trace 50 ;		
W55 71 circle trace 55 ;		
W60 74 circle trace 60 ;		
W65 20 circle trace 65 ;		
W80 21 circle trace 80 ;		
R40 22 circle flash 40 ;		
R50 23 circle flash 50 ;		
S50 24 square flash 50 ;		
R52 25 circle flash 52 ;		
S52 26 square flash 52 ;		
R55 27 circle flash 54 ;		
R56 28 circle flash 56 ;		
S56 29 square flash 56 ;		
R58 72 circle flash 58 ;		
S58 73 square flash 58 ;		
R62 125 circle flash 62 ;		
S62 126 square flash 62 ;		
R64 127 circle flash 64 ;		
S64 128 square flash 64 ;		
R65 129 circle flash 64 ;		
R66 130 circle flash 66 ;		

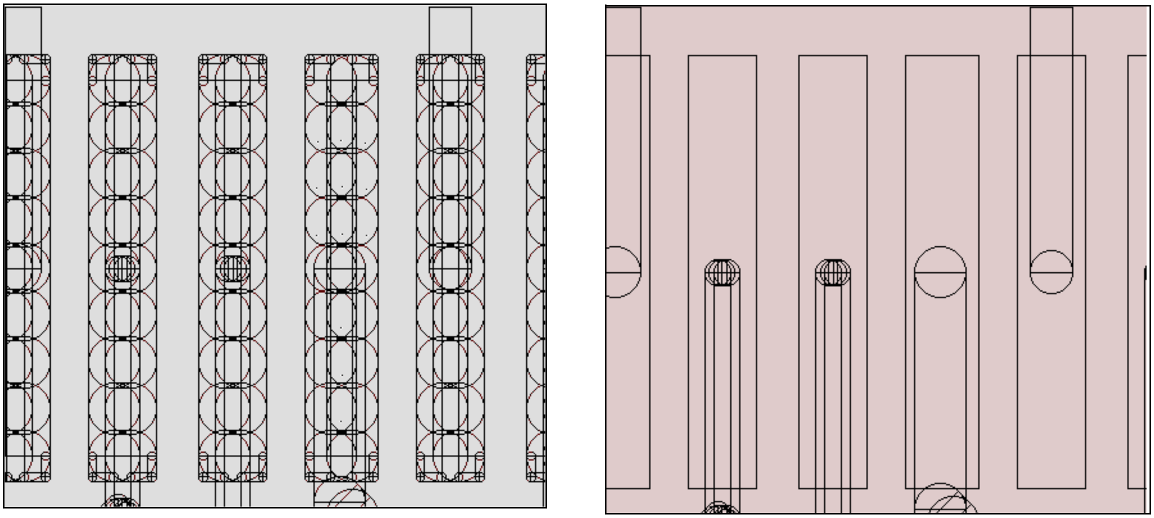
Wheel Template Editor

For more details about Auto Input please refer to book set 04 (Auto Input). The first book in the set, Doc. 0401, The Input Process, provides a detailed description of the input process and references to other books in this set.

CLEANUP AND ANALYSIS

After a job is created and populated with data from incoming files, the system can perform a thorough analysis for manufacturability.

Some analysis operations rely on representing the data in a certain manner. A typical example is drawn SMD pads. When outputting layer data in Gerber format, many EDA systems are limited by the different apertures (Dcodes) they can produce. As a result, rectangular pads are drawn as a large collection of lines, instead of being flashed as one pad. This phenomena, known as "drawn pads", causes problems to any analysis package. Pads are important objects in a PCB, and having them drawn causes the analysis to miss many of the measurements. One of the GenFlex cleanup operations automatically identifies drawn pads and converts them to flashed pads.



SMD - Before and After

Cleanup operations prepare the data for analysis and DFM actions. DFM actions are intended to improve manufacturability by modifying the graphical image. In some instances, all the cleanup action does is attach textual attributes to graphical features. These attributes are then used by the analysis actions to eliminate false alarms and speed up detection. Examples of such cleanup actions are Legend Detection and Set SMD Attribute.

See Doc. 0502, Cleanup Actions.

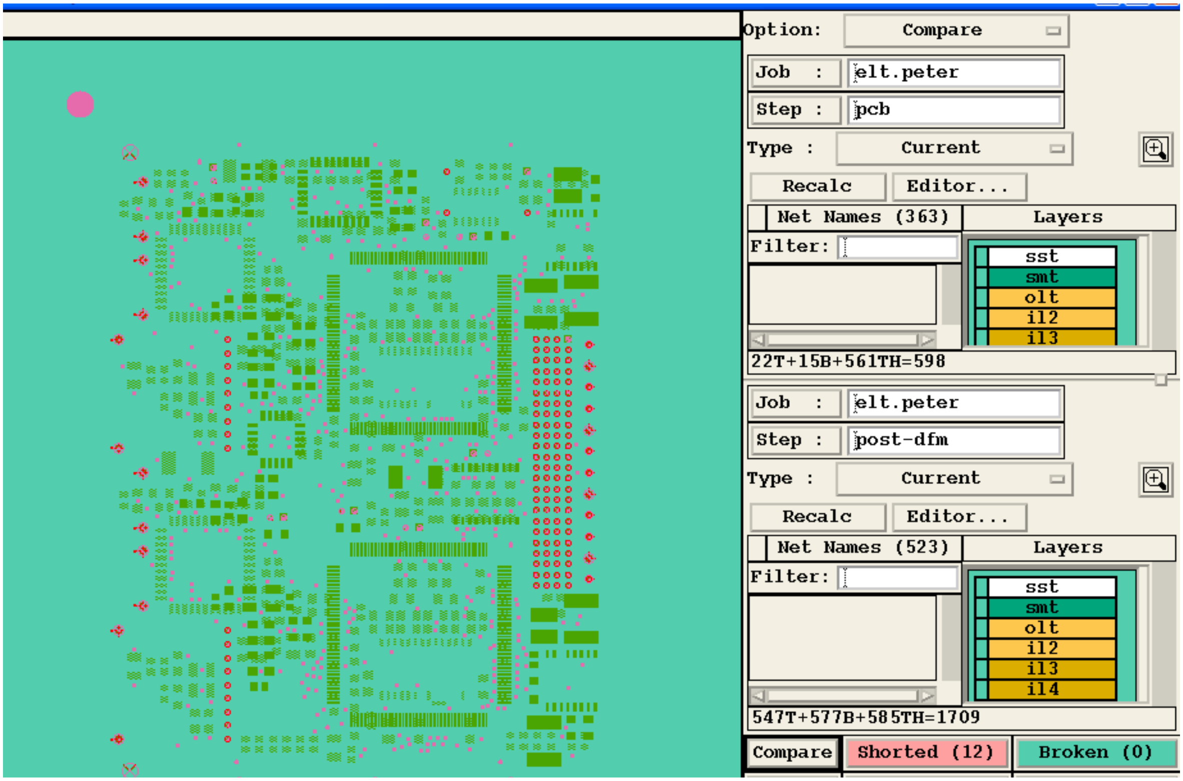
The Analysis Module is divided into two groups:

- Netlist Analysis
- Fabrication Analysis

Several analysis actions can be combined in a "checklist" to be run sequentially on all or part of the layers of the job.

Netlist Analysis

The Netlist Analyzer confirms that the original design connectivity netlist has actually been achieved by the features in the board's copper layers. If all processes were bug-free and operators never made mistakes, this process would be redundant. However, manual routing is sometimes erroneous, aperture lists may be wrong, and drill layer definition may be incomplete, to name just a few of the many problems. The Netlist Analyzer uses an intelligent imaging technology to compare netlists. Utilizing this technology, the graphical features in the various copper layers are converted to a large pixel-oriented image. This image forms the basis for connectivity calculation, avoiding complexities due to composite merges, odd shape symbols, etc. The results of the netlist comparison can be examined graphically to locate and fix problem areas. See Doc. 0506, The Netlist Analyzer.



Option: Compare

Job : elt.peter
Step : pcb
Type : Current
Net Names (363)
Layers: sst, smt, olt, il2, il3
22T+15B+561TH=598

Job : elt.peter
Step : post-dfm
Type : Current
Net Names (523)
Layers: sst, smt, olt, il2, il3, il4
547T+577B+585TH=1709

Example of the Netlist Analyzer Screen

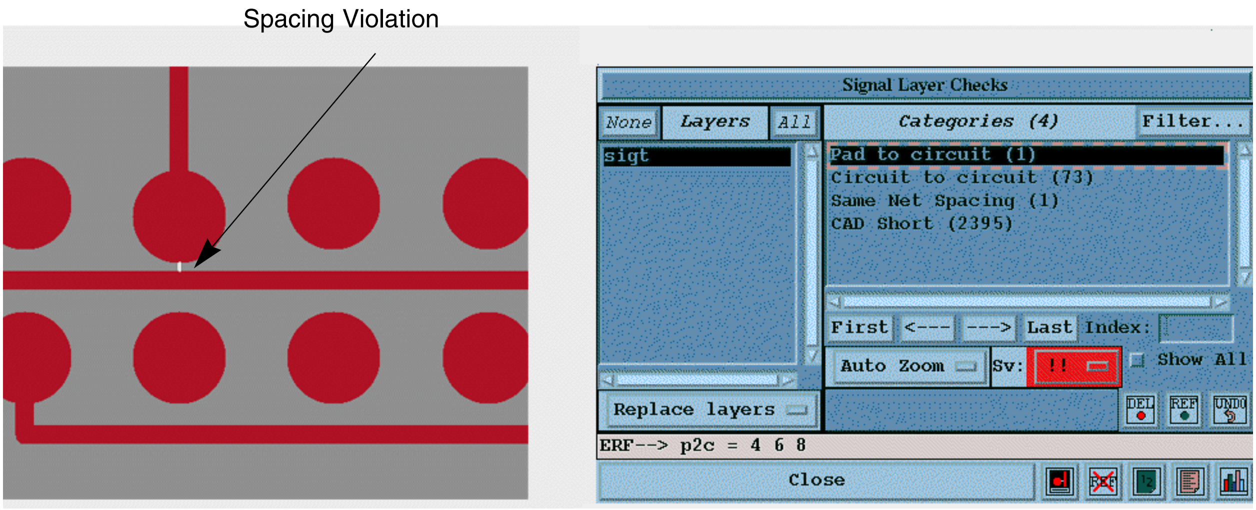
Fabrication Analysis

Fabrication Analysis examines data from the point of view of the bareboard fabricator. Below is a table with a partial listing of fabrication processes checked by the GenFlex system.

Spacing between copper features	Solder mask correctness	Non-plated holes tolerances
Annular ring of drill holes	Silk screen correctness	Thermals with insufficient contact ties
Double drill hits	Slivers in copper planes	

Eight independent actions are applied to over 70 analysis categories, with each category possessing user-defined tolerances that are specified in External Resource Files (ERFs). The system implementor defines the ERFs according to manufacturing requirements. Once the ERFs and the checklists have been set, running the analysis module becomes a simple one-step operation. A graphical Results Viewer window enables viewing specific defects as well as providing graphical histograms and textual reports.

See Doc. 0503, Fabrication Analysis.



Pad Spacing Violation Identified by the Results Viewer

In addition, there are specific Flex analysis actions which check, among other things, specific design problems on Flex boards. A partial list of such design problems is listed below.

Coverlay clearances to close to bend area	Stiffeners close to or overlapping the bend area	Sharp trace joints
Coverlay/drill/stiffener close to or overlapping air gaps in the adhesive	EMI (silver) mask versus board edge	Drills in or close to bend area
Coverlay too close to board edge	EMI mask versus Exposed copper, coverlay and drills	Trace I-Beam on adjacent layers

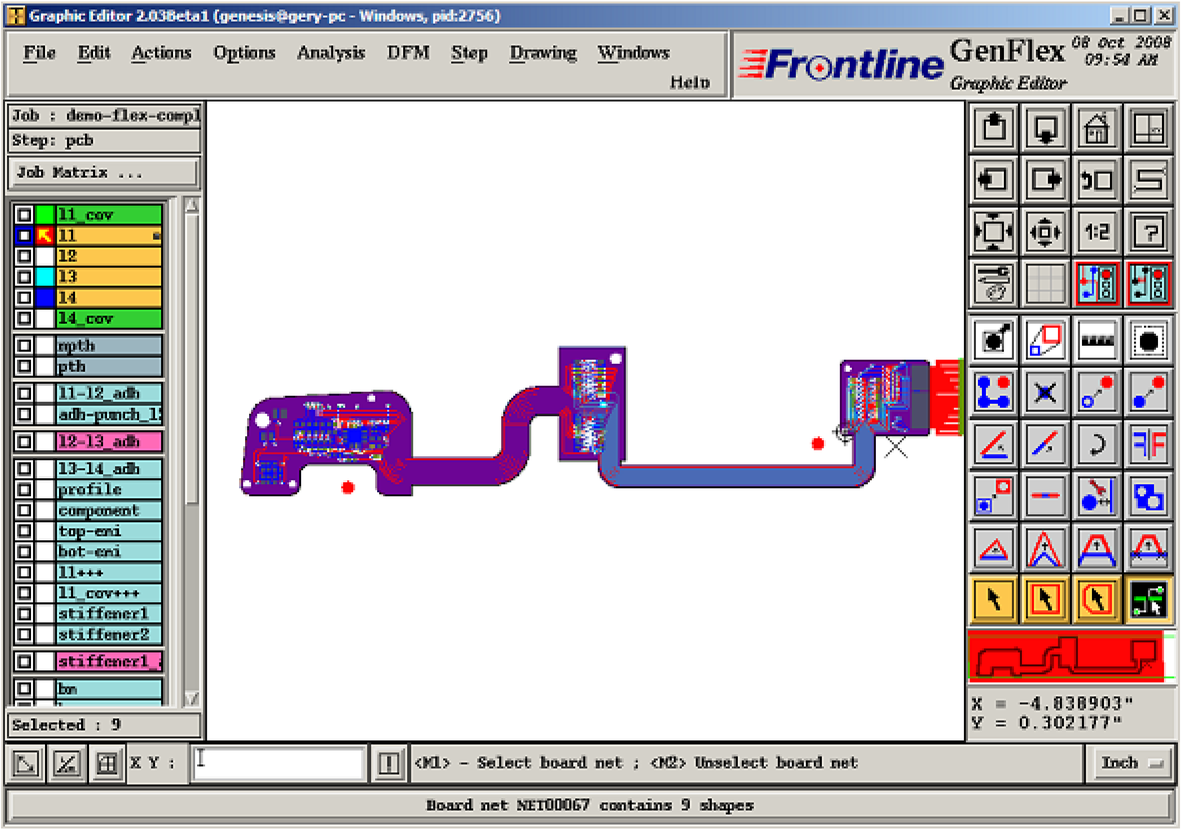
EDITING AND OPTIMIZATION

The GenFlex system has a wide variety of editing and optimization tools, ranging from local operations to sophisticated DFM (Design for Manufacturability) actions for optimization and yield improvement.

Changes at board level are important due to the fact that the master database, from which the design is produced, actually resides in the EDA system's database. There is a justifiable concern that changes made in Genflex are not retained when a new EDA revision of the board is read.

Graphic Editor

Board level editing functions are performed in the GenFlex Graphic Editor. In addition to optimizing, the Graphic Editor is a powerful tool that provides unique ways of viewing board data. Viewing is WYSIWYG (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et). The layers on the screen appear as if etched on copper or imaged on a mask. Tool Tips are available for most buttons and functions, as demonstrated below.



Graphic Editor 2.038eta1 (genesis@gery-pc - Windows, pid:2756)

08 Oct 2008 09:54 AM

Job : demo-flex-compl
Step: pcb

Selected : 9

X = -4.838903"
Y = 0.302177"

<M1> - Select board net ; <M2> Unselect board net

Board net NET00067 contains 9 shapes

Genesis Graphic Editor

VIEWING

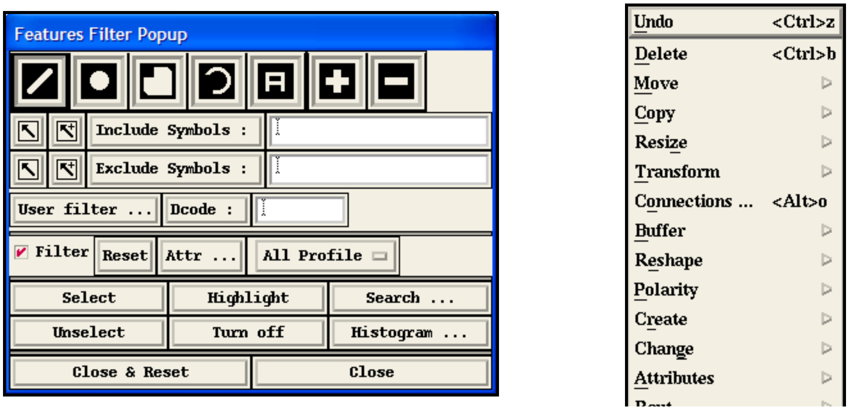
Viewing functions in the Graphic Editor include:

- Zoom, pan and snake path
- Skeleton, outline and full width modes
- Popview (magnifier glass) functions
- Various measurement tools (with snap capabilities)
- Information display for features and components
- Histogram list by feature symbols, component types and package names
- An extensive search facility
- Copper area measurement

EDITING

Editing capabilities in the Graphic Editor include:

- Single feature changes (add, move, copy, delete, etc...)
- Global changes using a filter to select features (resize, move, copy, delete, contourize, fill,...)
- Bus editing, polyline operations, Join clearances, Interactive spacing repair, round corners, tapering, and more
- Layer operations (copy, rename, merge, delete,...)
- And more.



Filter popup and the Edit menu

See Doc. 0601, The Graphic Editor.

Automating Board Level Changes

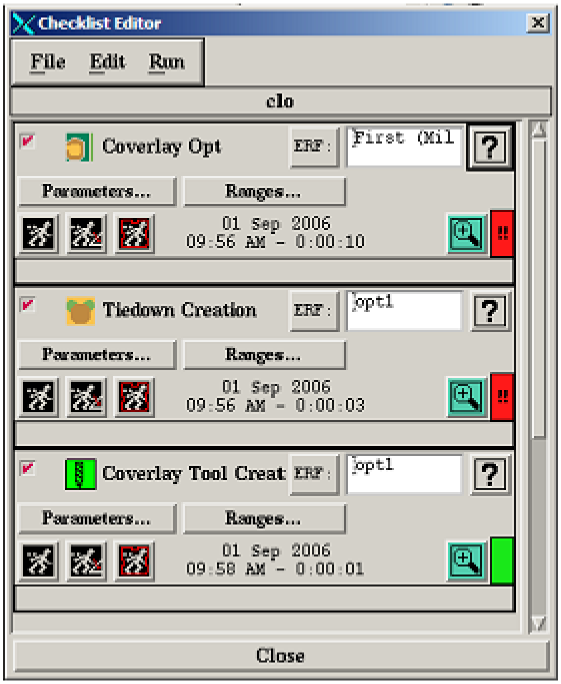
Changes at board level are important due to the fact that the master database, from which the design is produced, actually resides in the EDA system's database. There is a justifiable concern that changes made in GenFlex are not retained when a new EDA revision of the board is read.

To ensure the consistency and continuity of board-level editing, GenFlex offers checklists to automate board level changes.

CHECKLISTS

The solution is to use a number of global optimization actions bundled into a repeatable checklist. It is thus possible to repeat a series of actions in a standard procedure which consistently optimizes data before the data are output to manufacturing.

Doc. 0501, Checklist Operations describes the checklist mechanism.



clo

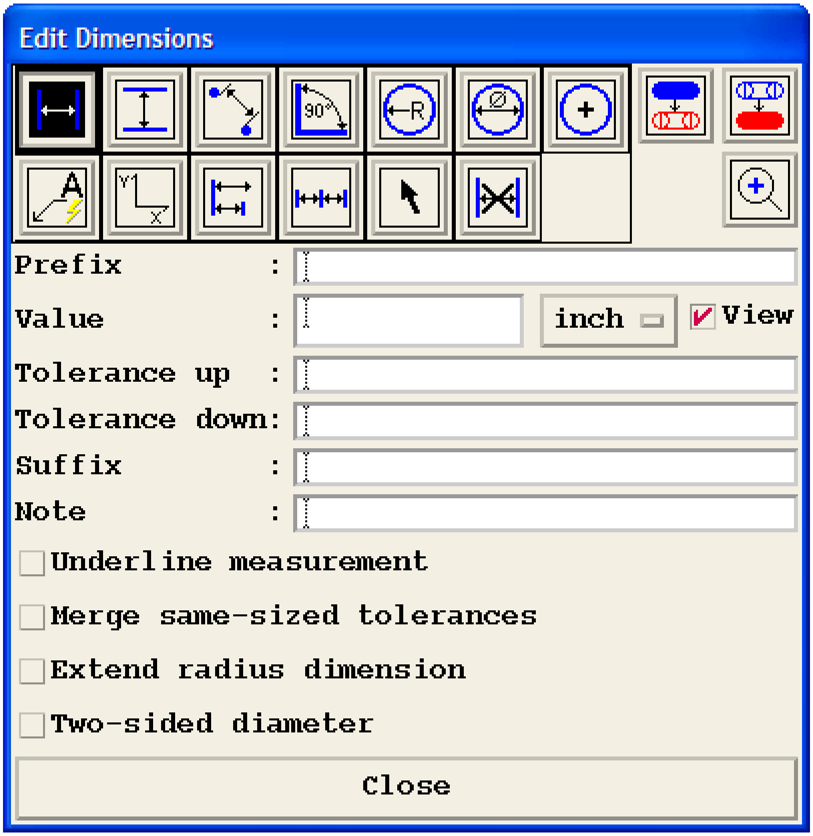
Coverlay Opt ERF: First (Mil
01 Sep 2006
09:56 AM - 0:00:10

Tiedown Creation ERF: opt1
01 Sep 2006
09:56 AM - 0:00:03

Coverlay Tool Creat ERF: opt1
01 Sep 2006
09:58 AM - 0:00:01

Drawing

GenFlex's unique **Dimensioning** module enables you to create printable drawings of layers or features that include dimensions. Dimensions show the geometric measurements of objects, the distances or angles between objects, or the distance of a feature from an origin you specify.



Prefix :

Value : inch ☑ View

Tolerance up :

Tolerance down:

Suffix :

Note :

□ Underline measurement

□ Merge same-sized tolerances

□ Extend radius dimension

□ Two-sided diameter

Close

Edit Dimensions Popup

See **Creating Dimensiong Drawings** in Graphic Editor, Doc. 0601gf.

Graphic Panelization

Panels are required in order to save setup time and provide efficient usage of materials. When the board is small, it is much more efficient to create an array (step & repeat) of boards on one large panel.

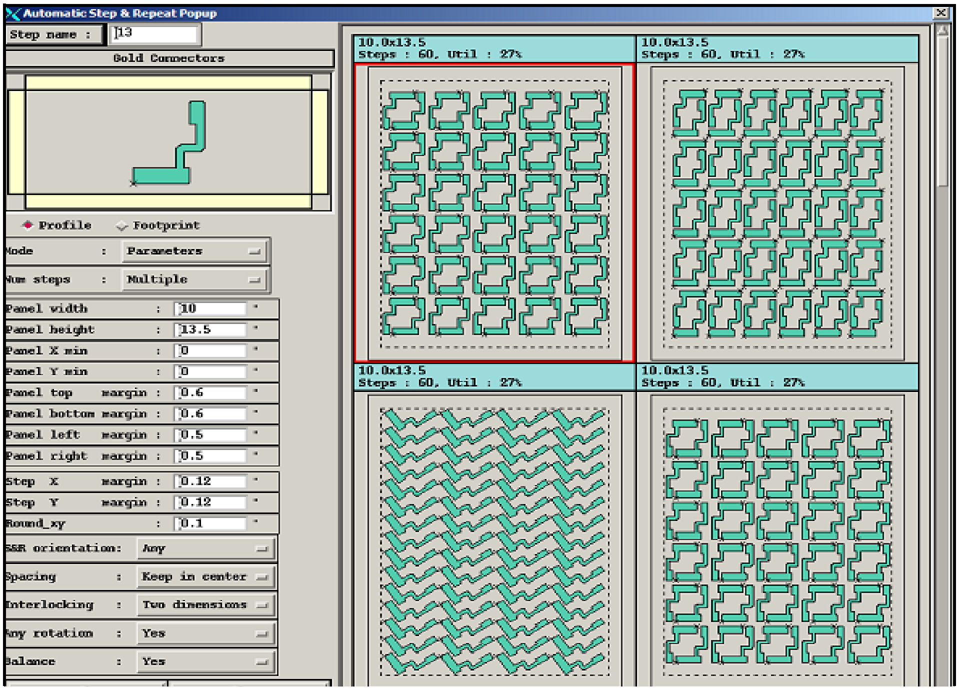
In the manufacturing environment, numerous additions to a panel are required to ready it for production. Thieving bars and venting patterns have to be added around boards. Test coupons have to be inserted in specific locations. Mounting holes and optical targets (fiducials) are required for the use of various production machines. Text and logos are also mandatory for easy identification.

GenFlex can create panels either through the Graphic Editor, or through a special Panelization Wizard.

PANELIZATION IN THE GRAPHIC EDITOR

The same Graphic Editor described earlier is used in creating the panel. In the job structure of Genesis, a panel which is built from a PCB is just another “Step” in the job matrix. The graphic tool works simultaneously on multiple steps, reflecting the changes that are made in a step into its parent step.

Special functions are required for panel creation. The automatic panelization function provides optimal placement based on a predefined set of panel classes (sizes). Automatic panelization includes the possibility of interlocking placement and any-angle rotation of boards inside a panel. Manual placement functions are also available to modify the automatic selection or to create a panel from scratch. Other tools include pattern fills based on any user-defined symbol, coupon placement tools and complete overlay additions.


Automatic Step & Repeat Popup

```
Step name : 13
Gold Connectors
◆ Profile   ◇ Footprint
Mode               : Parameters
Num steps          : Multiple
Panel width        : 10
Panel height       : 13.5
Panel X min        : 0
Panel Y min        : 0
Panel top    margin : 0.6
Panel bottom margin : 0.6
Panel left   margin : 0.5
Panel right  margin : 0.5
Step  X      margin : 0.12
Step  Y      margin : 0.12
Round_xy           : 0.1
SSR orientation: Any
Spacing        : Keep in center
Interlocking   : Two dimensions
Any rotation   : Yes
Balance        : Yes
```

```
10.0x13.5
Steps : 60, Util : 27%
```

```
10.0x13.5
Steps : 60, Util : 27%
```

```
10.0x13.5
Steps : 60, Util : 27%
```

```
10.0x13.5
Steps : 60, Util : 27%
```

Panel Optimization

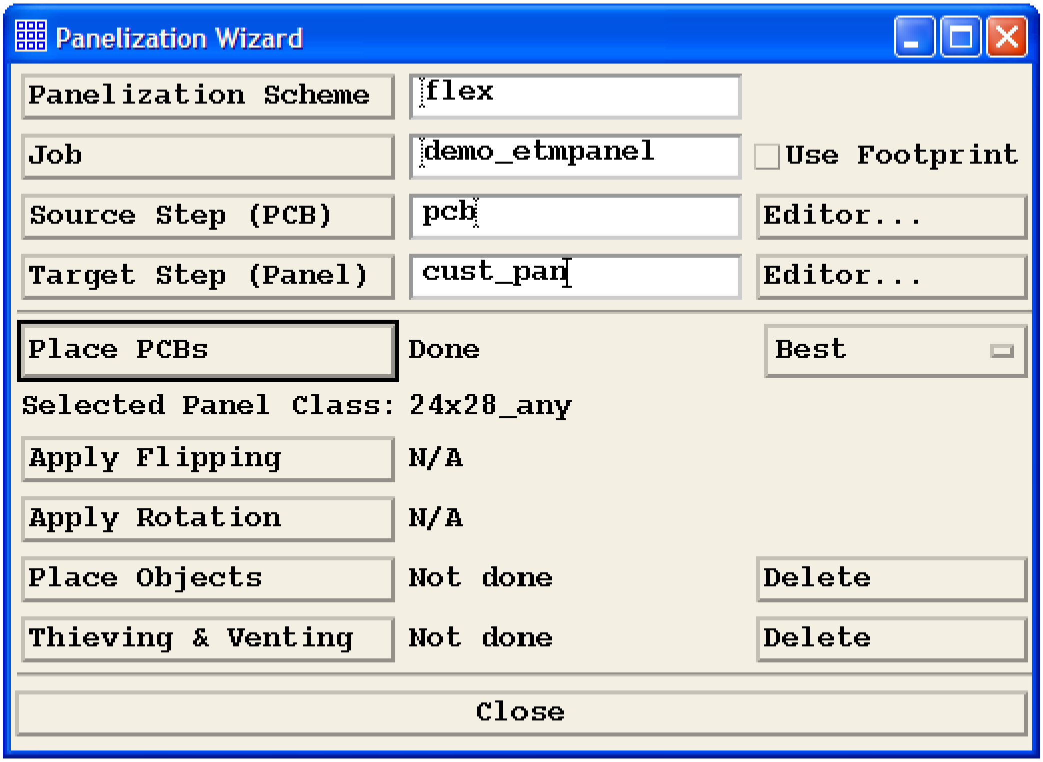
See Doc. 0601, The Graphic Editor.

PANELIZATION THROUGH THE PANELIZATION WIZARD

The Panelization Wizard is a tool used by the operator to build a panel from a PCB using a panelization scheme. The data is placed on the panel in stages. Between stages the operator is able to make any necessary changes manually. This is useful when the panelization scheme is not defined to handle a specific situation.

To call up the Panelization Wizard popup, select **Actions > Panelization Wizard** from the Main Menu bar of the Engineering Toolkit. The Panelization Wizard can also be opened by clicking on the **Preview** button of the Panelization Setup and Panelization

For details, see Doc. 0608 – Automatic Panelization Package.



Panelization Scheme	flex	
Job	demo_etmpanel	□ Use Footprint
Source Step (PCB)	pcb	Editor...
Target Step (Panel)	cust_pan	Editor...
Place PCBs	Done	Best
Selected Panel Class:	24x28_any	
Apply Flipping	N/A	
Apply Rotation	N/A	
Place Objects	Not done	Delete
Thieving & Venting	Not done	Delete
Close		

Panelization Wizard - Initial Setup

DFM Actions

A large number of DFM (Design for Manufacturing) actions perform optimization routines on various layers, such as solder mask, silk screen and signal layers. These routines utilize sophisticated algorithms as well as features of the database (negative data, contour features) to provide unique advantages. The optimization procedure increases the tolerances in the layers, effectively improving production yield and product reliability.

All editing functions at board level can be configured to operate in a netlist/DRC check mode that provides instant feedback on changes, and the ability to monitor these changes for compliance with a user-defined set of rules. If any of the rules are infringed, the system blocks the save privileges of the user until the violation is remedied, or an authorization issued with the necessary access privileges.

See Doc. 0602, DFM Actions.

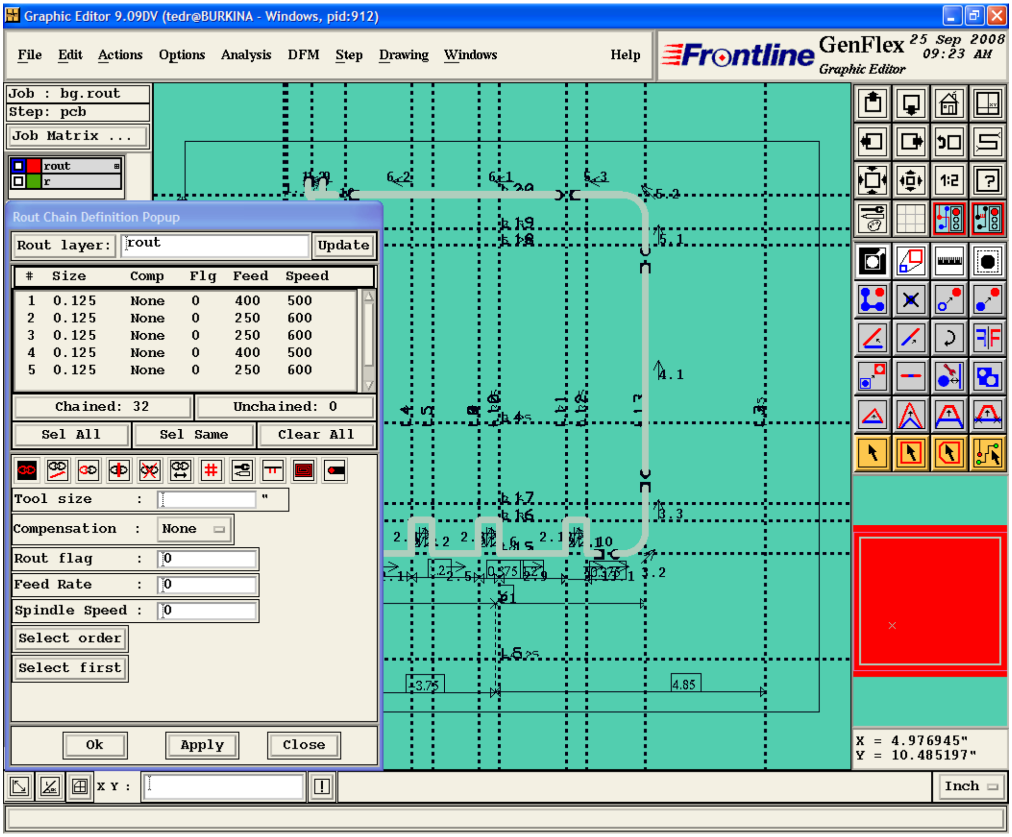
Routing

Normally, in fabrication, routing (milling) preparation can be a time consuming and error prone process. The Rout Editor, in conjunction with the Graphic Editor, is a powerful and flexible package that significantly improves this process.

Rout data typically arrives at the fabrication site in the form of a drawing. The Rout Editor translates a drawing into digital form using construction lines and points bonded by dynamic dimensions. Corners are built into intersections in a variety of shapes (rounded, chamfered, etc.). Finally, a sequence of rout lines and arcs are grouped into chains, each provided with the plunge and exit point instructions for the router.

The Rout Editor is geared mostly toward fabricators. Some of the information provided, such as compensation, router feed and speed, do not concern the designer. Still, it is advisable for the designer to examine the tool as a replacement for current drawing tools. Using ODB++, the basic rout drawing can be transferred digitally to a fabricator using GenFlex for completion. This method is more efficient than recreating the rout from the drawing.

See Doc. 0606, The Rout Editor.



Job : bg.rout
Step: pcb

Rout layer: rout

#	Size	Comp	Flg	Feed	Speed
1	0.125	None	0	400	500
2	0.125	None	0	250	600
3	0.125	None	0	250	600
4	0.125	None	0	400	500
5	0.125	None	0	250	600

Chained: 32
Unchained: 0

Tool size : "
Compensation : None
Rout flag : 0
Feed Rate : 0
Spindle Speed : 0

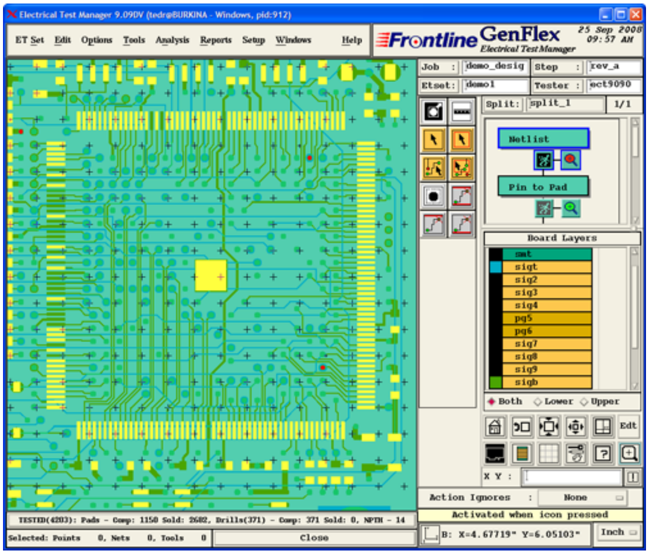
X = 4.976945"
Y = 10.485197"

Rout Editor

Netlist Optimization for Electrical Testing

The Netlist Optimizer module examines a CAD netlist or a netlist generated from Gerber data and optimizes it for efficient bareboard electrical testing. Optimization includes elimination of redundant test points and staggering test points on fine pitch devices to facilitate access for test probes.

The Netlist Optimizer can provide direct output of the required netlist files for flying probe testers and generic tester formats. If output to BON testers is required, the ETM module is also required.

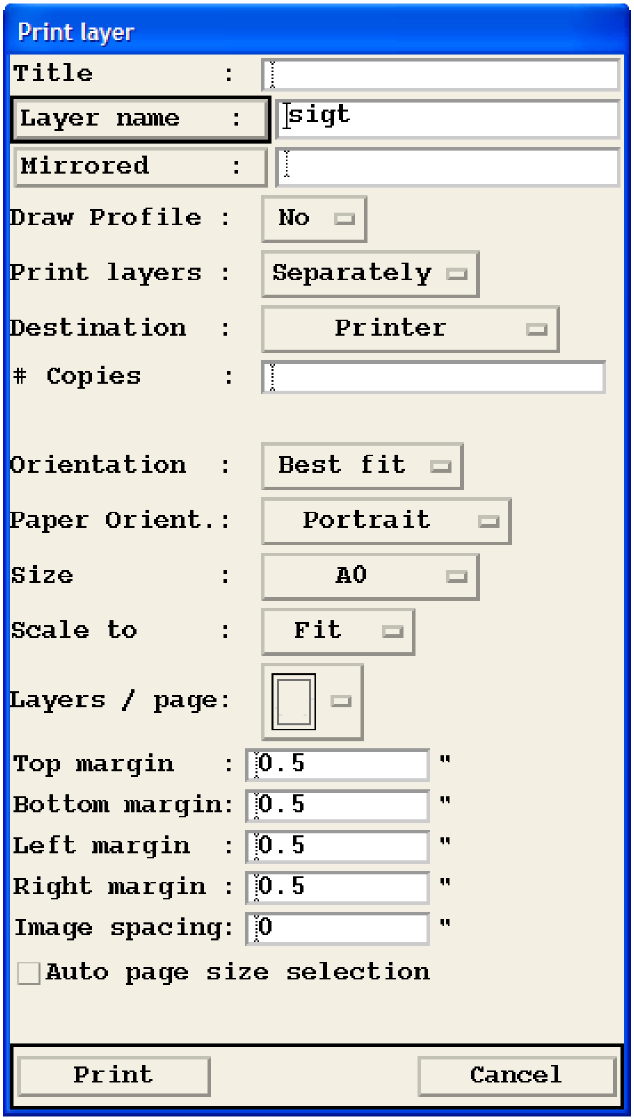


Job : demo_desig
Step : rev_a
Etset: demo1
Tester : ect9090
Split: split_1 1/1
X Y :
Action Ignores : None
Activated when icon pressed
TESTED(4203): Pads - Comp: 1150 Sold: 2682, Drills(371) - Comp: 371 Sold: 0, NPTH - 14
Selected: Points 0, Nets 0, Tools 0
B: X=4.67719" Y=6.05103"

Netlist Optimizer

PRINTING

The Print Layer operation enables you to print graphic layers from the currently displayed screen in PDF or Postscript format. To open the window, select **File > Print**. Select your print criteria, and send to print to the printer or to a file.



Title :

Layer name : sigt

Mirrored :

Draw Profile : No

Print layers : Separately

Destination : Printer

Copies :

Orientation : Best fit

Paper Orient.: Portrait

Size : A0

Scale to : Fit

Layers / page:

Top margin : 0.5 "

Bottom margin: 0.5 "

Left margin : 0.5 "

Right margin : 0.5 "

Image spacing: 0 "

□ Auto page size selection

Print Layer Window

For details, see Doc. 0601, Graphic Editor

OUTPUT

Once data has been analyzed, edited and optimized, it is ready for output to various destination machines and systems. The following output destinations are supported by Genesis:

Artwork Plotters	NC Drill Machines	NC Rout Machines
Assembly Machines	Electrical Testers and Repair Stations	CAM Stations
Pen Plotters	Direct Imagers	
Orbotech Plotters	Automated Optical Inspection (AOI) machines	

All output operations can be automated as part of the process. Manual output is also available for particular layers or specific parameters. Output operations can be further customized using user-defined “hooks”.

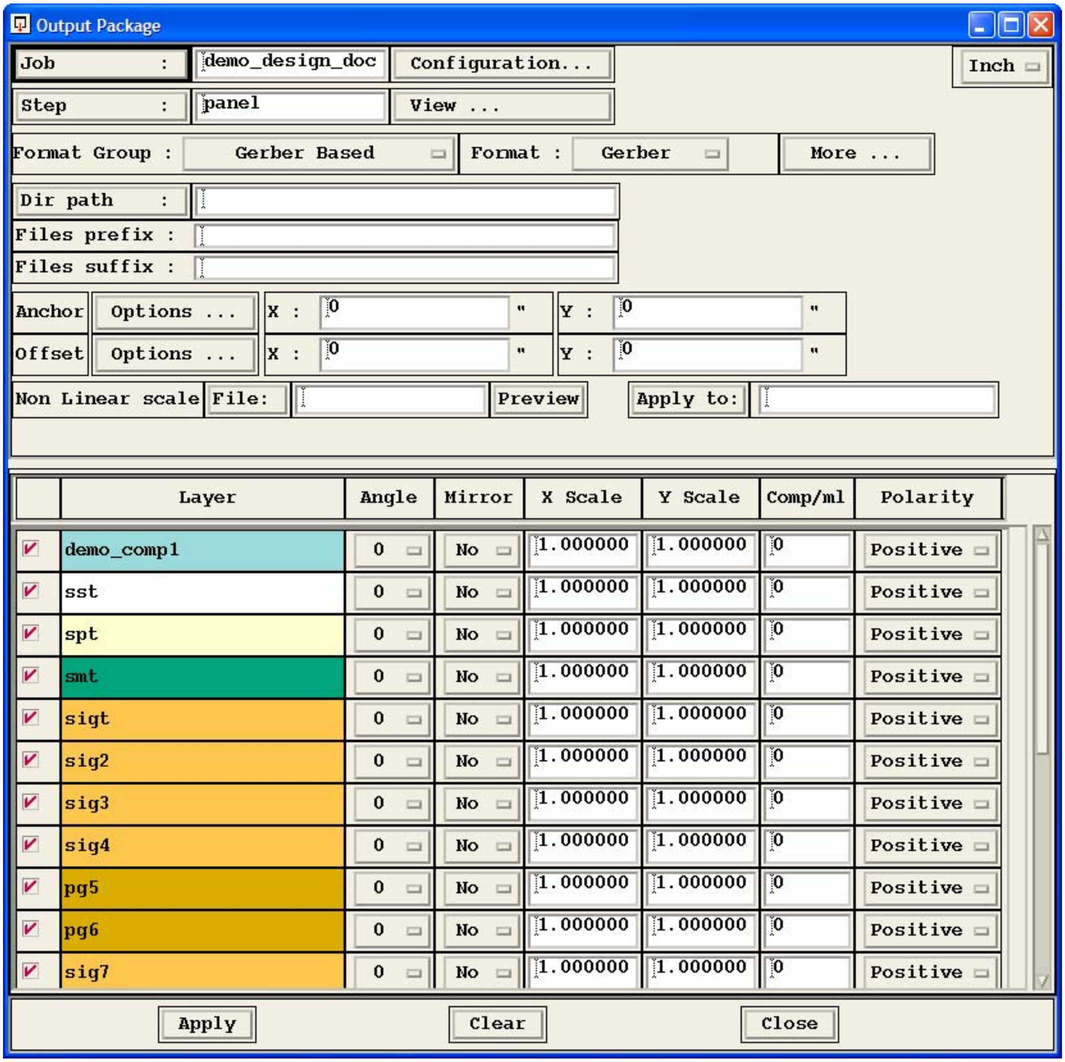
See Doc. 0701, The Output Process and Doc. 0702, Output Formats.

NC Drill and Rout Machines

In addition to the standard output functions, it should be noted that two modules are used to prepare highly optimized data for NC drill and rout machines.

- Auto Drill Manager performs sophisticated path optimization as well as tool management. Issues such as pilot drills, nibbling methods, tool sorting, large panel splits and more are handled easily by this tool.
- Auto Rout Manager controls routing and rout ordering on large panels.

Doc. 0703, The Auto Drill Manager and Doc. 0704, Auto Rout Manager, describe these modules.



Job : demo_design_doc

Step : panel

Inch

Format Group : Gerber Based

Format : Gerber

Dir path :

Files prefix :

Files suffix :

Anchor X : 0 " Y : 0 "

Offset X : 0 " Y : 0 "

Non Linear scale File:

Apply to:

	Layer	Angle	Mirror	X Scale	Y Scale	Comp/ml	Polarity
☑	demo_comp1	0	No	1.000000	1.000000	0	Positive
☑	sst	0	No	1.000000	1.000000	0	Positive
☑	spt	0	No	1.000000	1.000000	0	Positive
☑	smt	0	No	1.000000	1.000000	0	Positive
☑	sigt	0	No	1.000000	1.000000	0	Positive
☑	sig2	0	No	1.000000	1.000000	0	Positive
☑	sig3	0	No	1.000000	1.000000	0	Positive
☑	sig4	0	No	1.000000	1.000000	0	Positive
☑	pg5	0	No	1.000000	1.000000	0	Positive
☑	pg6	0	No	1.000000	1.000000	0	Positive
☑	sig7	0	No	1.000000	1.000000	0	Positive

Output Package Window

AOI Machines

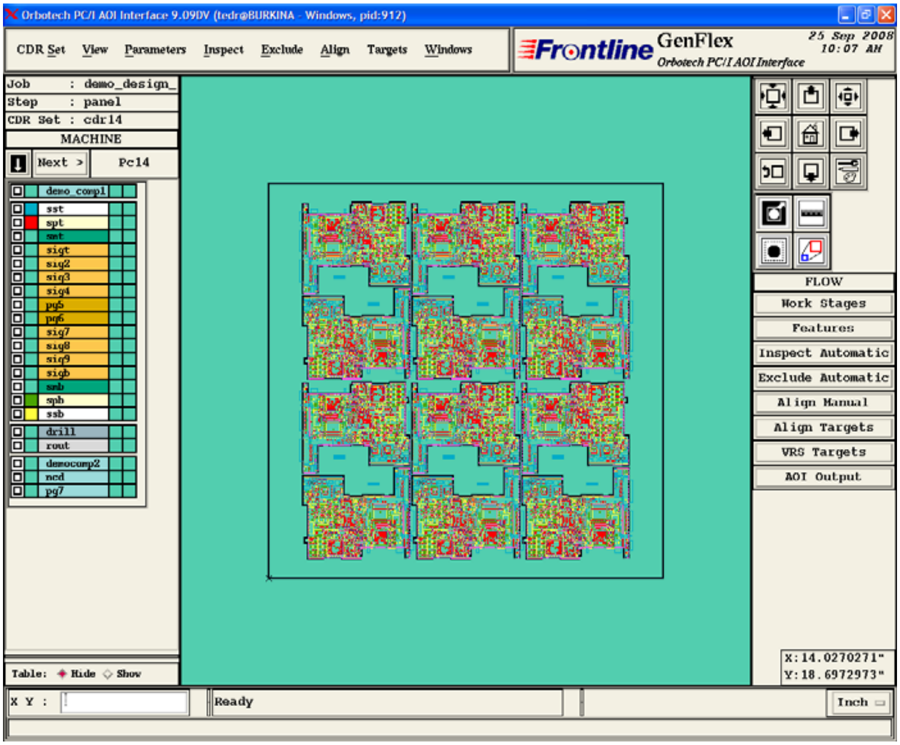
An interface to a variety of AOI machines is available. Current AOI systems for which a Genesis interface is available include:

Orbotech AOI Machines			Other AOI Machines
PC-14	Vision	Discovery	CAMTEK
Inspire	Infinex		Mania

ORBOTECH AOI MACHINES

PC/I AOI

The PC/I AOI Interface can, when properly configured, automatically detects text features on the board, and automatically assigns those text features to Don't Inspect areas.



Orbotech PC/I AOI Interface 9.09DV (tedr@BURKINA - Windows, pid:912)

25 Sep 2008 10:07 AM

Job : demo_design_
Step : panel
CDR Set : cdr14

Ready

X:14.0270271"
Y:18.6972973"

Orbotech PC/I AOI Interface window

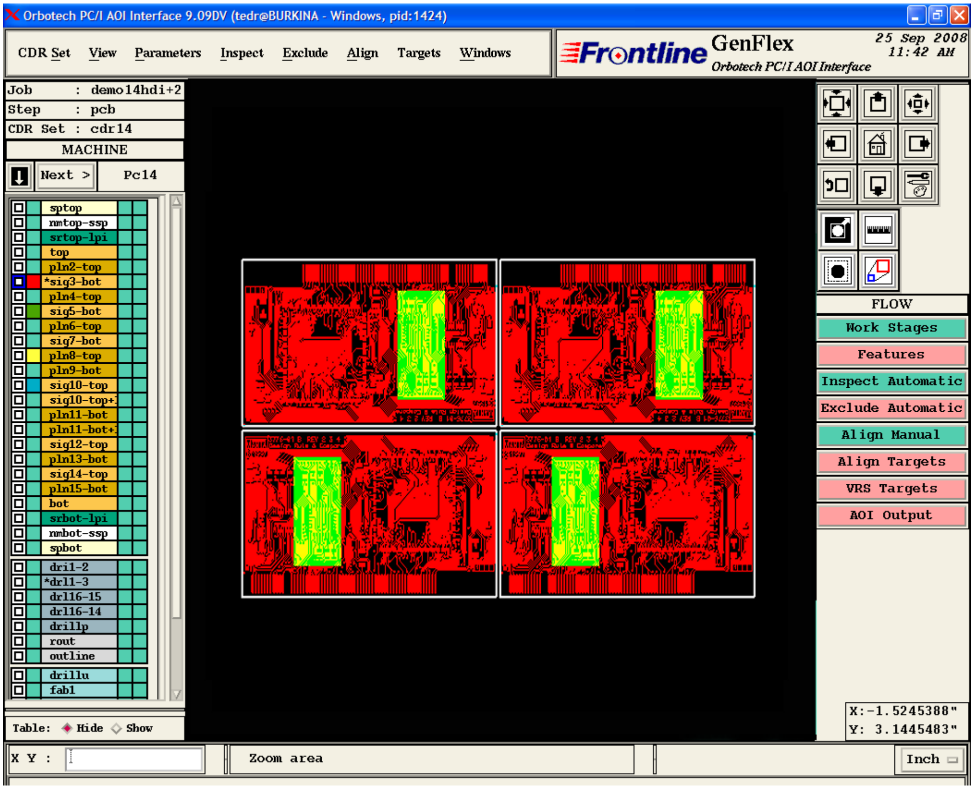
See Doc. 0711, Orbotech AOI Interface

VISION AOI

The Vision AOI interface introduces a new concept in CDR setup. Layer parameters are divided into two groups: CAM-oriented parameters (e.g. what are the features in the layer) and AOI-oriented parameters (e.g. what should be the inspection resolution).

Each group is processed by a different system. CAM-oriented parameters are set in the GenFlex Vision AOI interface. AOI-oriented parameters are set in the AOI Manager. This new concept utilizes the expertise of each system (GenFlex and AOI Manager) in its specific field (CAM and AOI) to achieve the best setup.

The GenFlex Vision AOI interface enables full automatic setup for all CAM-oriented parameters, and can prepare full automatic setup of AOI-oriented parameters in the AOI Manager.



Orbotech PC/I AOI Interface 9.09DV (tedr@BURKINA - Windows, pid:1424)

Job : demo14hdi+2
Step : pcb
CDR Set : cdr14

25 Sep 2008
11:42 AM

X:-1.5245388"
Y: 3.1445483"

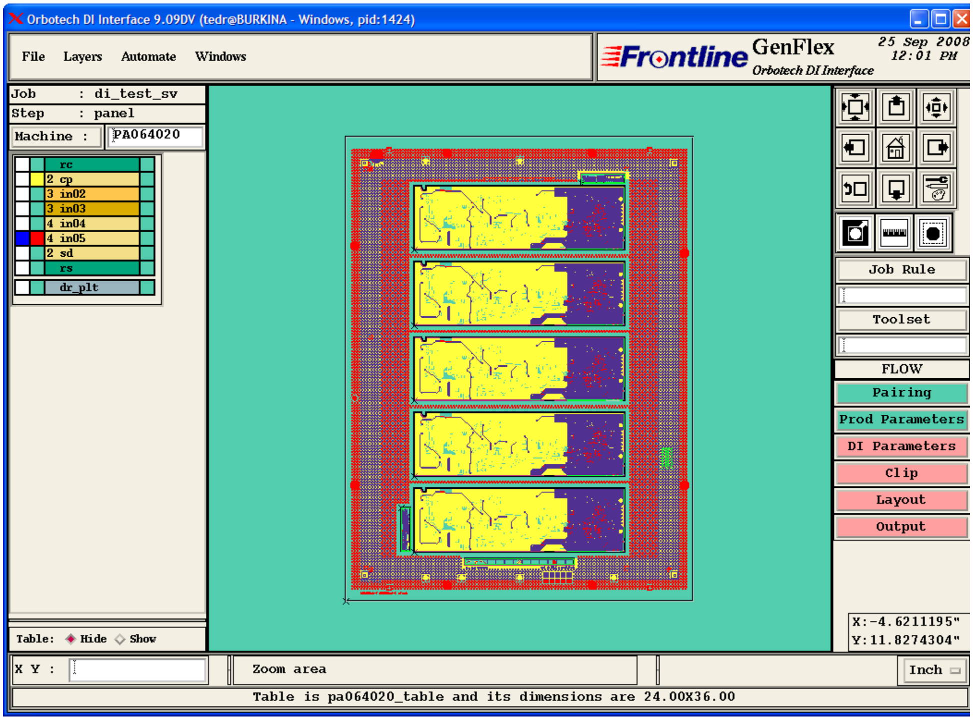
Orbotech Vision AOI Interface window

See Doc. 0710,Orbotech Vision AOI Interface

Orbotech DI (Direct Imaging) Interface

The Orbotech DI (Direct Imaging) interface matches the special needs of the DI machine and simplifies the setup process. The Orbotech DI Interface outputs job data to an Orbotech DI Machine.

A DI setup is a set of parameters for a specific layer, and is attached to a layer. This DI setup is saved with the layer and can be changed when necessary.



Orbotech DI Interface 9.09DV (tedr@BURKINA - Windows, pid:1424)

25 Sep 2008
12:01 PM

Job : di_test_sv
Step : panel
Machine : PA064020

X:-4.6211195"
Y:11.8274304"

Table is pa064020_table and its dimensions are 24.00X36.00

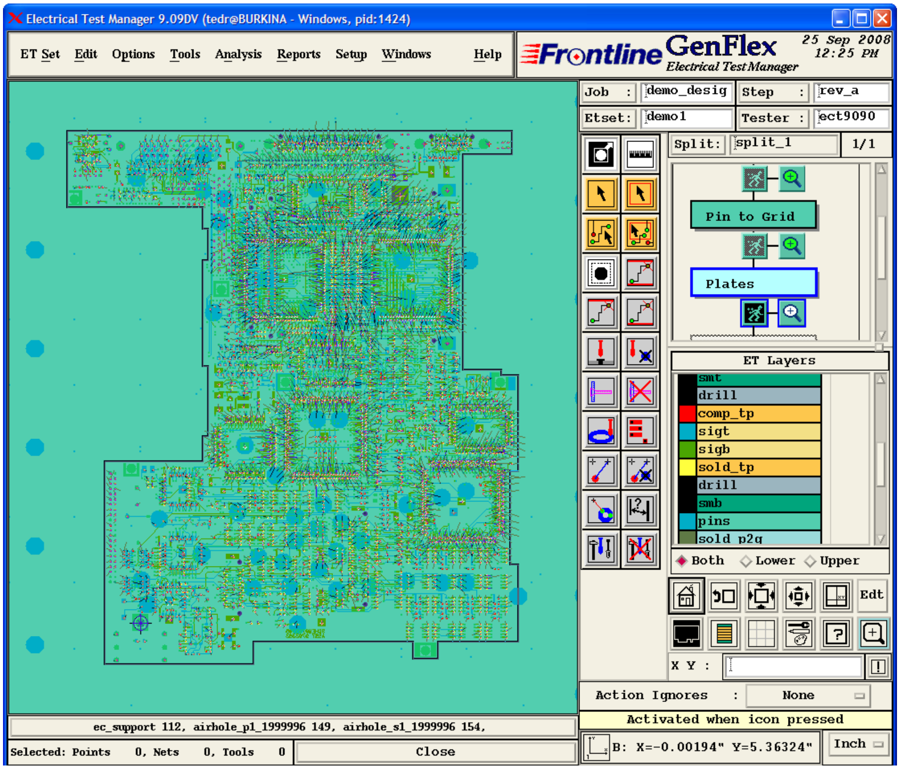
Orbotech DI Interface window

See Doc. 0712, Orbotech DI Interface

ELECTRICAL TESTING

The Electrical Testing Manager (ETM) is a powerful tool for creating test fixtures for all types of BON testers. Fully integrated with ODB++, system algorithms, and netlist analysis and optimization tools, ETM offers unparalleled integration of ET functions within the CAM environment.

See Doc. 0708, Electrical Testing Module



Job : demo_desig
Step : rev_a
Etset: demo1
Tester : ect9090
Split: split_1 1/1

ET Layers:
- smt
- drill
- comp_tp
- sigt
- sigb
- sold_tp
- drill
- smb
- pins
- sold p2g

◆ Both ◇ Lower ◇ Upper

ec_support 112, airhole_p1_1999996 149, airhole_s1_1999996 154,

Selected: Points 0, Nets 0, Tools 0

Action Ignores : None

Activated when icon pressed

B: X=-0.00194" Y=5.36324" Inch

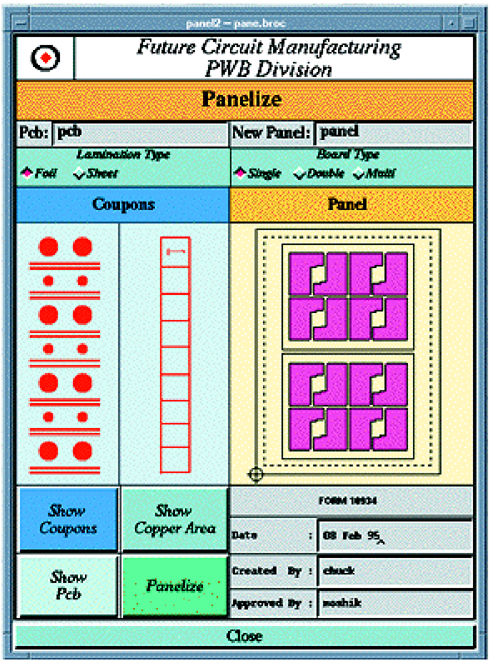
ETM Main Window

INFORMATION MANAGEMENT

One of the main advantages of the Genesis system is that it does not limit itself to performing CAM tasks. It can be viewed as a total environment, providing communication tools between different job functions. In the design environment, information can be exchanged between designers, production engineers and electronic engineers. In the manufacturing facilities, communication can be created between planners, CAM engineers and floor supervisors.

The system provides several tools to assist in creating these communication channels.

Work Forms


Future Circuit Manufacturing
PWB Division

Panelize

Pcb: pcb
New Panel: panel

Lamination Type: ◆ Foil ◇ Sheet
Board Type: ◆ Single ◇ Double ◇ Multi

FORM 10934
Date : 08 Feb 95
Created By : chuck
Approved By : moshik

Example of a Work Form

Work Forms are the electronic equivalent to familiar paper forms, used to automate job-associated paperwork. They are easily built using the graphical Form Builder. The Form Builder can create:

- **labels**
- **text fields**
- **option fields**
- **sliders**
- **graphical images**
- **push buttons**

Actions can be attached to a form in order to make it interactive. A form template in the database library job can be used to attach a form into a new job. Once in the job, data can be added to the fields manually or by a script written in the system. It is thus possible to automate various actions and record the results in forms. This information can be viewed by concerned personnel and can, in turn, be used as input to other scripts in the system.

See Doc. 0801, Work Forms

GENFLEX SUB-PROGRAMS

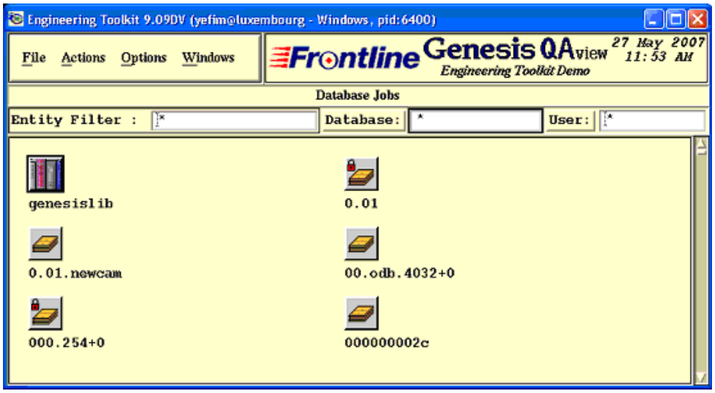
The GenFlex set of tools also contains a variety of tools and utilities that enable you to process your board data in the most efficient manner possible. These tools include:

- **Genesis LT (Genesis Lite)**
- **Genesis QA**
- **Genesis Viewer**
- **AT Gen**
- **GenFlex**
- **GenFlex Core**
- **GenFlex QA View**

GENESIS QA VIEW

The Genesis QA View product is a limited Genesis license which provides all capabilities of Genesis (input, editing, analysis) but cannot save changes or output any files.

It is a low-cost solution, especially useful for QA departments where verification and analysis are required but no outputs are created, and saving changes is not required.



Entity Filter : *
Database: *
User: *

System Administration

INSTALLING THE SYSTEM

The Genesis software can be installed from:

Frontline Internet Servers

CD-ROM

Prior to installation, the user should take the following actions:

Make sure that the system requirements are met as described in Doc. 0201, Software Installation.

Select a directory in which to install the system.

All software modules are installed in one location on the system. There are only minimal changes required in system files for automatic startup and for setup of an administrator user account.

Each user must add a “source” command in his/her startup file that points to a Frontline supplied definition file containing environment variables and aliases necessary for proper operation of the system.

The installation script provides instructions for adjustments to operating system parameters required for proper operation. The site system administrator should be available during the installation to coordinate these changes.

Doc. 0201, Software Installation and the CD-ROM package insert describe this procedure.

SYSTEM MANAGEMENT

After the system is installed and running, it does not require frequent system management. The administration work includes the following issues:

- **Defining GenFlex users and groups**
- **Setting privileges for different users/groups**
- **Modifying system configuration parameters and environment variables**
- **Adding user attributes as extensions to the system database objects**
- **Creating hooks as extensions to the system processes**
- **Defining scripts and bindings (described further on in this chapter)**
- **Retrieving weekly KIT (Keep In Touch) updates from the Frontline Internet server (KIT is described further on in this chapter)**
- **Installing new versions and patches as they are released**
- **Regular archiving of jobs and system files (including, optionally, archiving to the STAR 1000 system)**

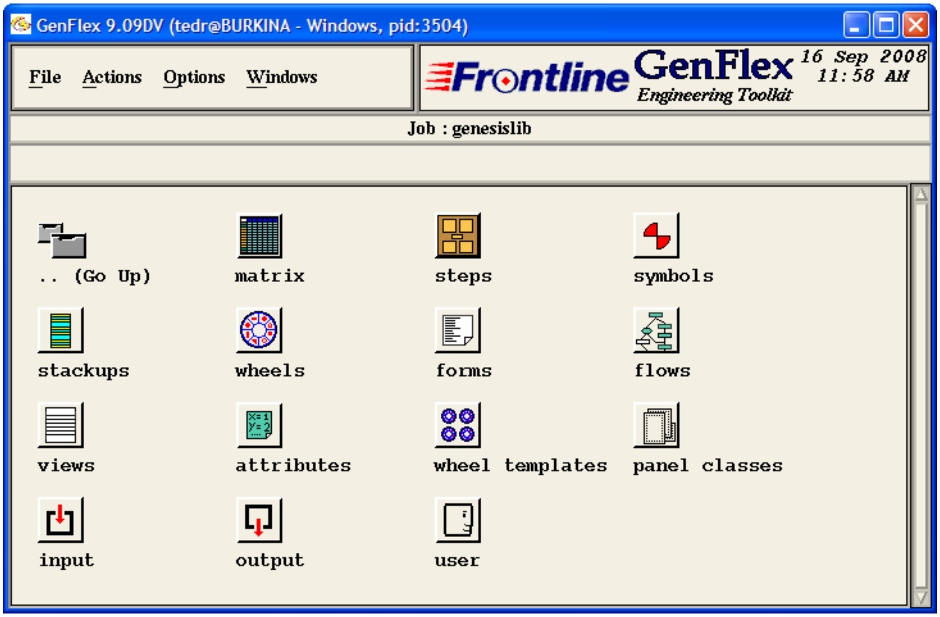
Properly organized system management is like oil to an engine. It keeps the engine running smoothly and provides users of the system with a smoothly run environment to improve productivity.

See Doc. 0203, System Management.

ODB++ DATA ORGANIZATION

ODB++ is a collection of job entities. A job is a self-contained, portable, comprehensive, and open set of data representing all aspects of a PCB design. A job is located in one directory, where subdirectories contain the various sub-entities. At its lowest level are files that contain actual data. These files are all ASCII text files, which makes them readable and open. Some files, because of their size, are compressed using the standard UNIX compress command.

A job is composed of entities, as shown in the Engineering Toolkit window below:



Sub-entities of a Job

- Steps are multi-layer entities (such as a single image, a sub panel array, a production panel or a multi layer coupon). Each step contains a collection of layers. Layers are two-dimensional sheets, containing graphics, attributes and annotations. Layers express physical board layers, mask layers, NC drill and rout layers and miscellaneous drawings. All steps in a job have a common list of layers, albeit the contents may be totally different.

- The Job Matrix consists of rows (job layers) and columns (job steps). Each row contains information such as type, polarity and context of a layer. The matrix is also crucial in defining the physical order of the layers and the relation of drill layers to other layers (through, blind, buried, etc.).

- Symbols are single layer graphic entities which can be referenced from within any graphical layer in a step.

- Work Forms are a user-defined collection of fields (textual and graphical), and buttons.

- Work Flows are user defined procedures, composed of stages, conditions and switches presented hierarchically.

Various additional files are organized each in their own sub- directories (wheels, input, output,...).

Each Step entity contains, in addition to general information and a list of layers, several other important entities:

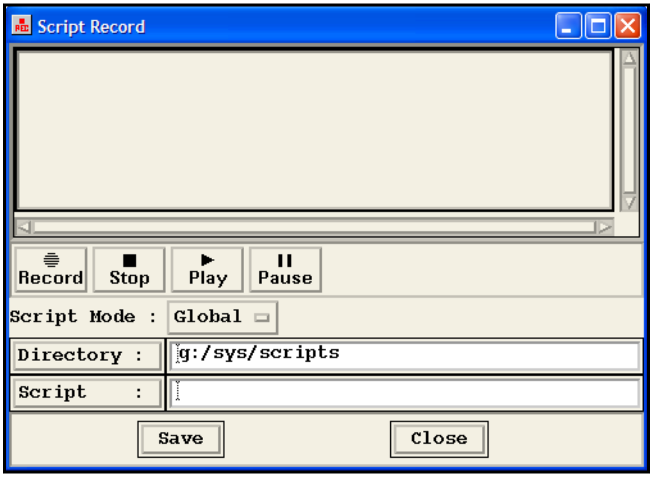
- **Step & repeat information (in the `stephdr` file), containing information on existing prior steps with their relative location and orientation.**
- **Up to four netlists derived from the step (CAD netlist, Reference netlist, Current netlist, and Currently-based CAD netlist).**
- **An EDA object that contains data describing the component packages and pins, and information on the relation of features in the board layers to specific design nets and properties imported from the EDA system.**
- **An unlimited number of checklists, each composed of analysis and/or DFM actions. An action contains parameters and results (measurements) of the last successful run.**
- **A profile, which is a schematic border around a step.**

See Doc. 0202, ODB++

SCRIPT AUTOMATION

Scripting is one of a number of unique and powerful automation tools that GenFlex makes available to the system impelementor. Though most CAD/CAM systems today have some level of scripting, GenFlex takes this concept to a much higher level through the use of standard languages.

Almost every operation which is performed in the system is translated to a line-mode command, complete with parameters. The commands are written to the system log file and to the shell in which the application runs. Additionally, it is possible to start a script recorder which will record only the line mode commands into a file.



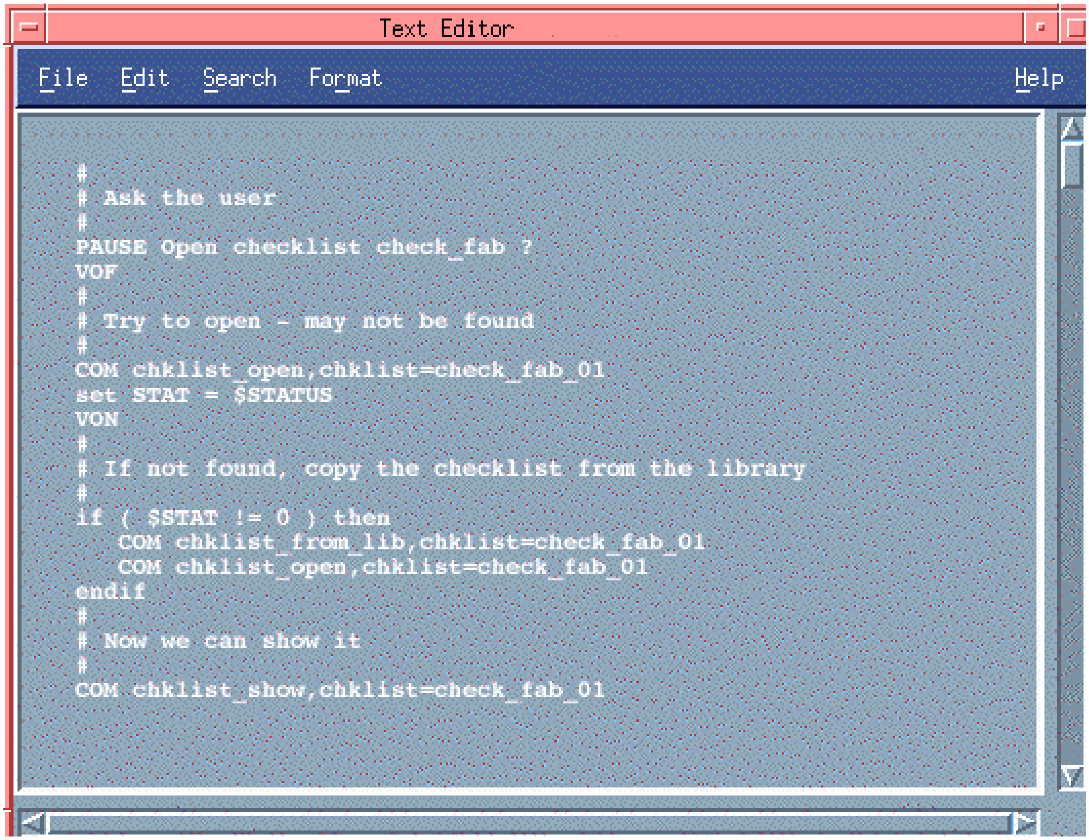
Script Mode : Global

Directory : g:/sys/scripts

Script :

Example of the Script Recorder

Some systems stop at this point and just let you replay a sequence of recorded commands. What makes Frontline applications unique is the ability to extend the recorded script by adding constructs in the most commonly accepted shell languages. Starting from **csh** which is the one of the most common shell languages used in UNIX systems, going to more powerful shell languages such as **sh** and **Tcl/Tk**, and growing into the Perl language, a full-featured programming language. In all these languages, the GenFlex line mode commands are integrated as if they were native commands of the shell language.



```
#
# Ask the user
#
PAUSE Open checklist check_fab ?
VOF
#
# Try to open - may not be found
#
COM chklist_open,chklist=check_fab_01
set STAT = $STATUS
VON
#
# If not found, copy the checklist from the library
#
if ( $STAT != 0 ) then
   COM chklist_from_lib,chklist=check_fab_01
   COM chklist_open,chklist=check_fab_01
endif
#
# Now we can show it
#
COM chklist_show,chklist=check_fab_01
```

Example of a Script

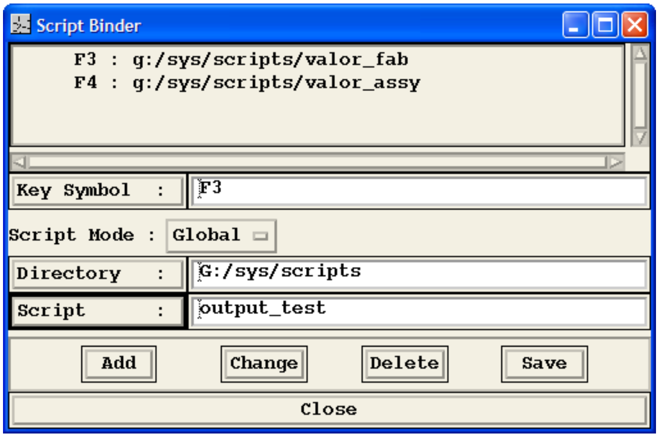
The main benefit to the system implementor is the ability to record a basic sequence of operations and then standardize them to cover a range of potential cases. Using conditions, loops and variables, the writer of a script can accommodate every possible case. Standard operating system commands can be used for file access, text processing, etc. In addition, a simple yet powerful GUI (Graphic User Interface) tool is available to prompt the user with queries and obtain user preferences.

Scripts can be prepared and stored anywhere in the file system. The application supports two default locations for scripts:

A global directory shared by all users

A local directory per user, where experimental scripts can be tested

A very important feature is the ability to bind a script to a function key on the system. All F-keys on the keyboard (with the exception of F1), alone or in conjunction with <ctrl> and <shift> combinations, can be programmed to execute a given script. This provides an efficient way of performing routine operations with a minimum of intervention.



```
F3 : g:/sys/scripts/valor_fab
F4 : g:/sys/scripts/valor_assy
```

Key Symbol : F3

Script Mode : Global

Directory : G:/sys/scripts

Script : output_test

Example of a Script Binder

Scripts can also be bound to a Work Form field or button. There are no differences in results between a script activated directly, or one activated from a form. One thing to note is that a script activated from a form receives additional parameters reflecting the form and field that triggered the operation. A directly executed script may require input from the user to provide values for these parameters.

To summarize, using scripts in the GenFlex is an important enhancement to system operations. The system administrator/implementor should be aware of the advantages of this tool and utilize it to its utmost capacity in order to achieve the highest throughput and efficiency.

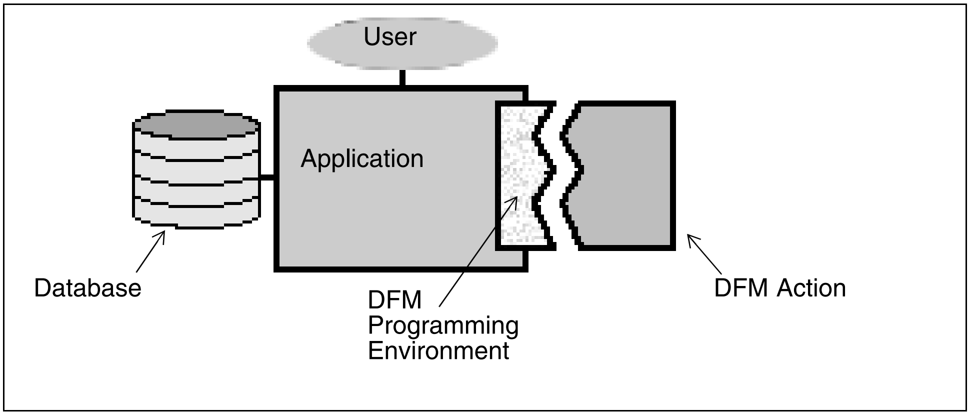
See Doc. 0204, Scripts. Doc. 0206, Line Mode Commands lists all the line-mode commands.

THE DFM PROGRAMMING ENVIRONMENT

While scripts provide an excellent solution for automating repetitive tasks, they are far from ideal for performing complex analysis or data optimization on graphical layers. These types of operations require access to tens of thousands of features in a very short time. The process calls for sophisticated geometrical algorithms and advanced search capabilities.

All this requires the power of a high level language and access to library functions.

Genesis was designed with this type of extensibility in mind. A special plug-in architecture enables users to create their own programs (DFM actions) and include them in the system. When the program initializes, it checks a user directory for the existence of these actions. If found, the action will be loaded into the same process space as the program, where it can be executed from the system menus as part of the application.



Using the DFM Programming Environment

The R&D team at Frontline uses the exact same mechanism for developing the Analysis and DFM functions now available in Genesis. That is why Frontline can confidently recommend that advanced users use this tool for their own special needs. Naturally, the availability of this tool does not affect Frontline's determination to keep adding new functions. It does give users the extra assurance that they are not subject to R&D development priorities. Whenever necessary, users can develop additional logic in-house in a fraction of the time it would have required to develop a function from scratch.

In order to use the DFM Programming Environment, the system implementor requires only a basic knowledge of the C programming language. This environment removes many of the complexities involved in writing algorithmic code. Issues such as memory allocation, geometrical calculations and complex data structures are all handled transparently by C macros which are translated into library calls to the application. Actual projects performed by Frontline customers show that a programming task which may have required a full month to complete in the past can be reduced with the DFM Programming Environment to a period of about two days.

Doc.0205 The DFM Programming Environment manual will provide full details and examples.

VERSION RELEASE POLICY

A Genesis version number is indicated by 2 two-digit numbers and a letter (e.g. 06.0D). The first two-digit number is the major release number. The second is the minor release number. The letter specifies a patch identification.

Frontline version release policy for Genesis is targeted for one major release a year. A major release contains important functionality enhancements. Some enhancements are included free of charge. Others are defined as optional with a requirement for an additional license.

Minor releases are made available between major releases 3-4 times a year. They include bug fixes and completion of functions which did not make it into the major releases.

Patches typically address critical bugs which need immediate response and are released according to need. They include a partial set of the modules that augment the version.

Frontline software is released through the Internet. Customers with Internet access can download and install the versions independently. Customers without internet access can receive the software on CD-ROM.

Doc. 0201, Software Installation and the CD-ROM insert booklet contain instructions for loading and installation of Genesis software.

ON-LINE DOCUMENTATION RELEASE

Frontline documentation is available in Adobe Acrobat format (pdf) on the Frontline ftp server site. The full set of available documentation can be loaded from the Frontline FTP server and installed for network wide access. The software which is required to view and search through the documentation (Acrobat Reader with Search) is distributed freely from the Adobe web site at (**`http://www.adobe.com`**). Frontline provides a download location of this software on its ftp server as well.

The documentation is used as on-line help in Genesis.

For more details regarding on-line documentation and the ability to extend it with internal documents please refer to Doc. 0103, Documentation Basics.

THE KIT (KEEP IN TOUCH) TOOL

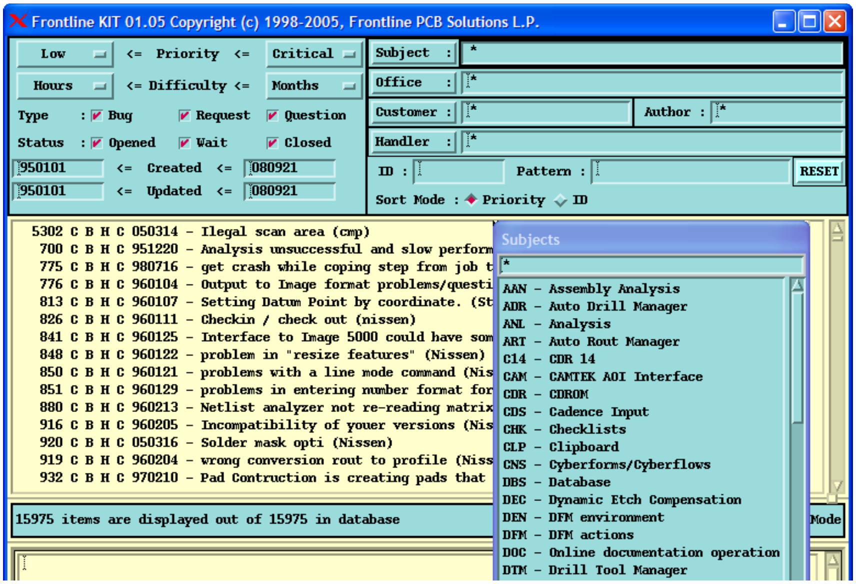
The KIT tool is a unique way of distributing information regarding bugs, requests and questions throughout the Frontline user community. KIT is distributed freely to all Genesis users. The KIT program works on one master file (**`kit_master`**) and several auxiliary files (**`kit_customers`**, **`kit_subjects,...`**). Frontline releases, on a daily basis, an updated set of KIT files, directly downloadable from the Frontline ftp server.

The KIT user has access to all items submitted, whether his/hers or other users'. Each item is assigned a unique ID the moment it is inserted into the kit_master file. This ID is guaranteed to remain constant through the lifetime of the item. Each item is also assigned priority, difficulty, customer, subject, handler, author and more. The KIT program provides easy to use filtering and search capabilities through all the items in the list.

Users can use the KIT program to submit new items. A similar capability (creating KIT items) is available on the Frontline web site (www.frontline-pcb.com). Kit items are received directly by the Frontline R&D team and are inserted into the **`kit_master`** file, so that the user who submitted the item can verify its acknowledgment (on the next **`kit_master`** release).

Users are encouraged to respond to other users' KIT items when they have an important addition or suggested workaround. By referencing the unique ID of the KIT item they respond to, R&D will attach the addition to the original item and create a thread which can be followed by other users.

See Doc. 0203, System Management.



Frontline KIT 01.05 Copyright (c) 1998-2005, Frontline PCB Solutions L.P.

Low <= Priority <= Critical
Hours <= Difficulty <= Months
Type : ☑ Bug ☑ Request ☑ Question
Status : ☑ Opened ☑ Wait ☑ Closed
950101 <= Created <= 080921
950101 <= Updated <= 080921

Subject : *
Office : *
Customer : * Author : *
Handler : *
ID : Pattern : RESET
Sort Mode : ● Priority ○ ID

```
 5302 C B H C 050314 - Ilegal scan area (cmp)
  700 C B H C 951220 - Analysis unsuccessful and slow perform
  775 C B H C 980716 - get crash while coping step from job t
  776 C B H C 960104 - Output to Image format problems/questi
  813 C B H C 960107 - Setting Datum Point by coordinate. (St
  826 C B H C 960111 - Checkin / check out (nissen)
  841 C B H C 960125 - Interface to Image 5000 could have som
  848 C B H C 960122 - problem in "resize features" (Nissen)
  850 C B H C 960121 - problems with a line mode command (Nis
  851 C B H C 960129 - problems in entering number format for
  880 C B H C 960213 - Netlist analyzer not re-reading matrix
  916 C B H C 960205 - Incompatibility of youer versions (Nis
  920 C B H C 050316 - Solder mask opti (Nissen)
  919 C B H C 960204 - wrong conversion rout to profile (Niss
  932 C B H C 970210 - Pad Contruction is creating pads that
```

15975 items are displayed out of 15975 in database

Subjects
*

```
AAN - Assembly Analysis
ADR - Auto Drill Manager
ANL - Analysis
ART - Auto Rout Manager
C14 - CDR 14
CAM - CAMTEK AOI Interface
CDR - CDROM
CDS - Cadence Input
CHK - Checklists
CLP - Clipboard
CNS - Cyberforms/Cyberflows
DBS - Database
DEC - Dynamic Etch Compensation
DEN - DFM environment
DFM - DFM actions
DOC - Online documentation operation
DTM - Drill Tool Manager
```

KIT Screen

ACCESSING FRONTLINE ON THE INTERNET

Frontline maintains a centralized Web site:

http://www.frontline-pcb.com

Additional FTP servers available for downloading software and uploading files: ftp://ftp.frontline-pcb.com

Mail can be sent to:

gen_kit@frontline-pcb.com - KIT (Keep In Touch) export items

doc@frontline-pcb.com - documentation feedback

sales@frontline-pcb.com - sales-related issues

Where To Go From Here

Congratulations! You have just completed an extensive walk-through of the Frontline Genesis 2000 system applications. By now, you should have a good idea as to the functionality the system provides. Undoubtedly, if this is your first introduction to the system, you must be somewhat overwhelmed by the task facing you - learning all this vast functionality and putting it into action.

The purpose of this first book is to inform you of what is in store for you. There is no need to implement all the modules at once. You can learn one step at a time, gradually extending system usefulness and improving operating procedures.

If you are the system implementor, you should continue by reading the System Administration group of books (02). Doc.0201 Software Installation and Doc.0203 System Management are of particular importance initially. Doc.0204 Scripts may prove useful in the implementation stages. Doc.0202 ODB++, Doc.0205 The DFM Programming Environment and Doc.0206 Line Mode Commands can be deferred to later stages.

Doc. 0103, Documentation Basics provides a full description of the documentation library, the on-line documentation system and a glossary of common terms and conventions. It also contains a documentation map for further reading. It is advisable for all users of the system to start from this book and continue with the remainder according to specific interests and needs. It is recommended that all other books are read with full access to a working system. This way you can perform examples independently and gain hands-on experience.

Please remember that the system you have just received is dynamic and constantly being updated. Every version will include new functionality and improvements which may fulfill additional requirements. You are encouraged to be involved, submit your questions and requests through KIT, and let us know your needs. We, at Frontline, will do our best to respond in a timely manner and solve the issues as fast as possible.

Enjoy your work!

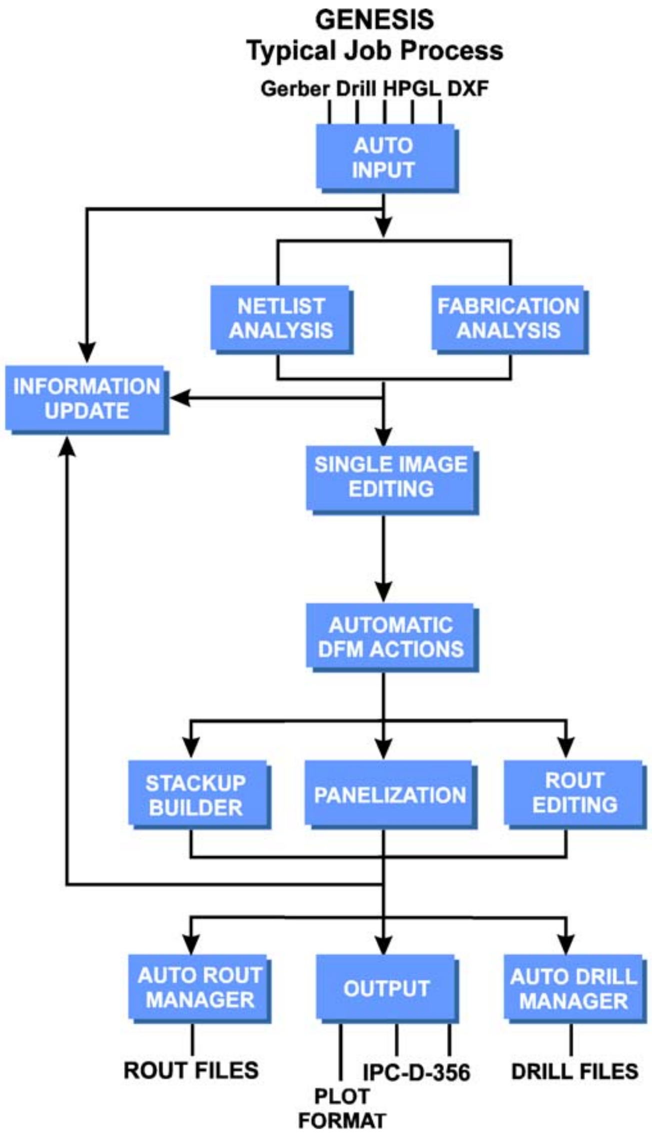
Additional Information

In this section we provide short answers to questions often asked by new customers, including:

- What is a typical job process in Genesis?
- What programming languages can I use for scripting in Genesis?
- How do I archive jobs in Genesis?
- How can I customize Genesis to meet my particular needs?
- Where are the system files located?

We welcome you to the ranks of satisfied Genesis users, and stand ready to assist you in any way we can.

Common Examples



FREQUENTLY ASKED QUESTIONS

1 ***Are all the modules described above a part of the standard offering?***

All the products described above are offered by Frontline. However, the Frontline policy advocates using a highly modular approach. The user only purchases what he needs. Each module can be licensed according to the simultaneous instances of usage (floating license policy). If any of the modules described above are not functional in your system due to insufficient license privileges, please contact your Frontline office for a free trial (send request by email to sales@frontline-pcb.com).

2 ***What if I would like to limit some of the functionality to specific operators only?***

The system implementor can define groups of users and assign privilege levels accordingly. Each user is assigned a privilege level based on his/her user ID (login). For every line mode command in the system, the system implementor can assign the minimum privilege level required.

See Doc. 0203, System Management.

3 ***How do I archive jobs in the system for future use?***

Frontline offers the STAR 1000 product – a complete solution for archiving, remote viewing and revision control. The STAR 1000 has special interfacing hooks with Genesis. It provides long term storage of jobs, including a flexible attribute mechanism and annotation capabilities. For more details, please refer to the Frontline Web site (http://www.frontline-pcb.com).

4 ***Can I use programming languages for scripts other than the ones specified in this document?***

The interface between the running script and the application is based on standard ASCII headers transferred through the **`stdout-stdin`** of the running script. This allows scripts to be written in any programming language, including **`Python, ksh, C`** and **`C++`**. All the user needs is a very simple set of definitions in order to submit line-mode commands to the application and receive the status. For more details, please contact your Frontline support center.

5 *Should I start using scripts and the DFM Programming Environment right after the system installation?*

The answer is ‘yes’ for scripts and ‘no’ for the DFM Programming Environment. Scripts is a relatively simple tool, easy to use and carries significant benefits for the general users (elimination of mundane tasks, avoiding procedure errors,...).

The DFM Programming Environment, on the other hand, requires deeper knowledge of the system and understanding of the basic constructs of the database. After basic implementation is complete, you may find that you have some specific needs, especially in the analysis and DFM areas, which are not solvable using scripts. This is the time to delve into the DFM Environment literature and explore its possibilities.

6 *What is the recommended mode of transfer from Enterprise to Genesis?*

Enterprise (from Valor) and Genflex share a common database format: ODB++. It is highly recommended to use a simple compressed **tar** file of the job as the medium of transfer between systems. It assures that data intelligence will be retained.

7 *I looked at some files in the database (features.Z) and tried to view them. The editor showed a collection of non-comprehensible characters. Isn’t the database made of readable ASCII files?*

Some files are compressed due to their size. To decompress, type from the shell:

```
uncompress features.Z
```

Then look at the file called ‘**features**’. The system deals with both compressed and decompressed files. You can save on disk space by compressing files. Type:

```
compress features
```

where **features** is the name of the file to compress.

Error Messages

Error messages in the system are displayed in a light red popup window consisting of the error number, a 5 or 6 digit number, and the message. Some of the errors are due to operator error while others may be caused by environmental conditions and/or system bugs.

When reporting such an error, it is important to include in the report:

- The steps which lead to the error and the error number and text.
- A portion of the system log file (the last 50 last lines, if possible).

Send the report to your local Frontline office, or see "On-Line Documentation Release" on page 45.

To view the system log file:

- Open the Engineering Toolkit window.
- Select Actions>View Log... from the menu bar.

Appendix C, which is found in many of the manuals in the documentation library contain explanations of potential errors and their causes.

SYSTEM ADMINISTRATOR NOTES

The following notes are basic for understanding some of the system aspects of Genesis. This material is covered in more detail in Doc.0201 Software Installation and Doc.0203 System Management. However, some important definitions. are worth repeating:

Root directory of the Enterprise / Genesis:

`/genesis` (can be overridden by the **`$GENESIS_DIR`** environment variable)

Location of the software:

```
$GENESIS_DIR/e<xx>
```

where xx is the major/minor number of the version (e.g. e70 for version 7.00)

Note: *This directory is overridden in new releases/patches.*
NO CHANGES SHOULD BE PERFORMED HERE BY ANY USER.

Location of the user definitions/customization:

```
$GENESIS_DIR/sys
```

Location of the jobs database:

The jobs database can be distributed between systems. The paths to these databases can be found in the file:

```
$GENESIS_DIR/sys/dblist
```

The logical database for each job can be found in the file:

```
$GENESIS_DIR/share/joblist
```

Location of system scripts:

```
$GENESIS_DIR/sys/scripts
```

Location of user local definitions and scripts:

`~/.genesis` where ~ is the home directory of the currently logged in UNIX user (not the Genesis user)

Location of the system's main daemon which controls the licenses, locking and inter-process messages:

```
$GENESIS_DIR/e<xx>/gnd/gnd
```

Location of the Cyberlink daemon:

```
$GENESIS_DIR/e<xx>/cns/cns
```

Frontline PCB Solutions
Headquarters
4 Faran St. , Ashtrum Park
Northern Industrial Zone, Yavne
P.O. Box 680, 81106, Israel
Tel: (972)-8-932-2183
Fax: (972)-8-932-2186
Email: info@frontline-pcb.com

Distributors:
USA - Orbotech Inc.
Tel: 1(978)-667-6037
Europe - Orbotech S.A.
Tel: (32)-2-727-4811
Far East - Orbotech Pacific Ltd.
Tel: (852)-2827-6688
Japan - Orbotech Japan Ltd.
Tel: (81)-3-3280-1300

www.frontline-pcb.com

Frontline
PCB SOLUTIONS
An Orbotech Valor Company

FLGFXoverview 0065 8/2008