

## *New Product Announcement*

# **Power PathPort<sup>tm</sup>**

### **Introduction**

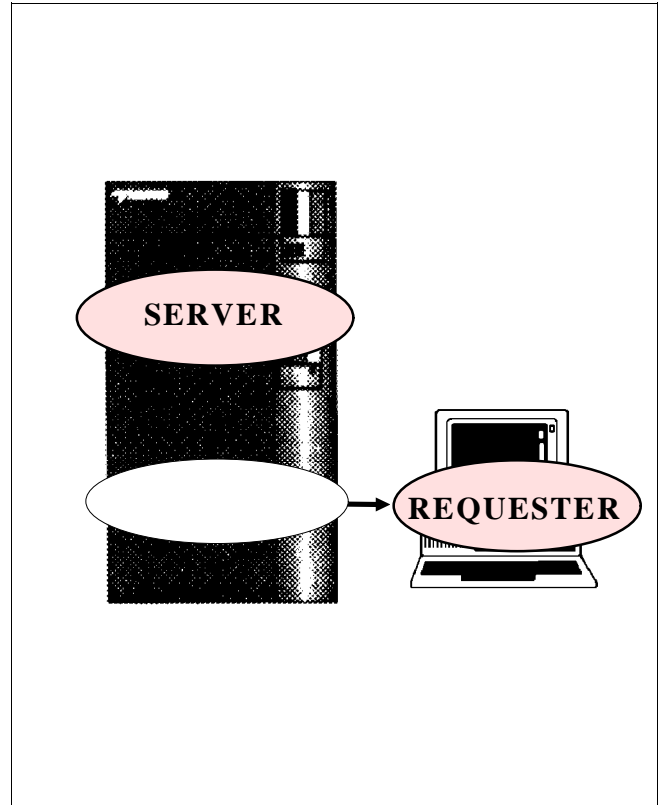
Power PathPort is an innovative new product which converts legacy Pathway applications to client/server architecture and graphical user interface (GUI).

As its name implies, Power PathPort actually ports Screen COBOL Pathway requesters to a combination of PowerBuilder applications and Windows DLL's running on user PC's.

Product highlights include:

- *Ports Pathway Applications to Client/Server*
- *Converts Pathway Screens To PowerBuilder Windows*
- *Generates C++ DLL From SCOBOL Requester*
- *Facilitates Extensive GUI Enhancements*
- *Provides Automated Conversion Process*
- *Complies With Open Systems Standards*
- *Requires No Application Development Or Changes*
- *Protects Existing Application Investment*

Power PathPort was developed in response to numerous customer requests to provide a product that not only automatically performs the conversion of a Pathway application to client/server, but also facilitates the easy and



*Power PathPort converts Pathway applications to client/server architecture and GUI.*

rapid enhancement of the resulting GUI using industry standard tools.

A critically important feature of this product offering is the interface it affords to Powersoft Corporation's PowerBuilder product.

PowerBuilder is the leading Windows GUI development tool in the market today. When used in conjunction with Power PathPort, it provides the capability to quickly and easily revise the application's GUI presentation to the extent desired by the end user customer.

The Power PathPort solution provides not only the capabilities of converting legacy Pathway applications to client/server under a customized GUI, but the ability to do so at a small fraction of the time and cost normally associated with reimplementing.

This document outlines the main features of this new product announcement.

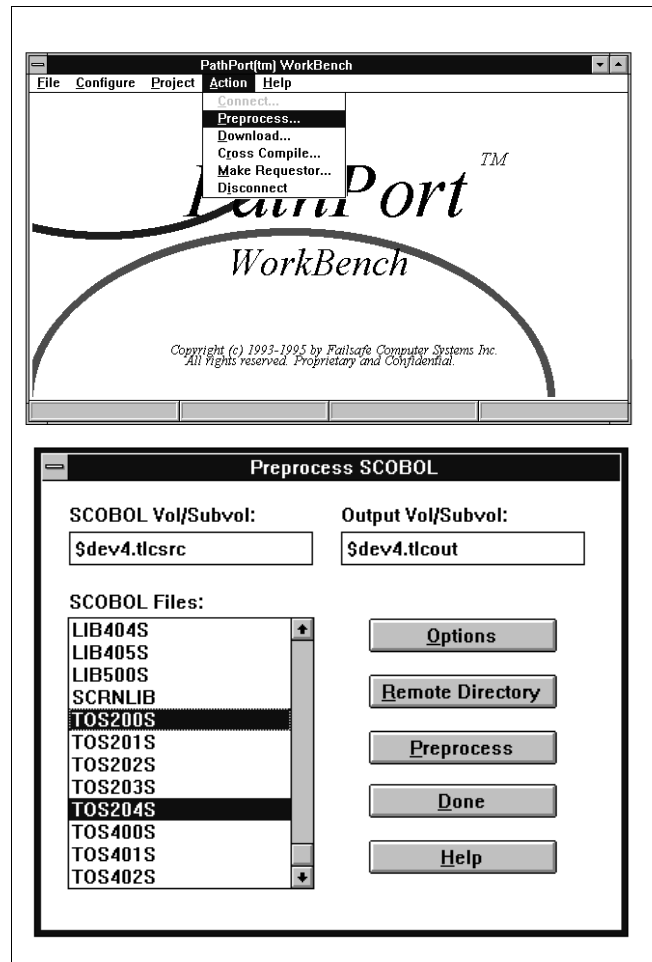
### The PathPort Process

A key feature of the Power PathPort approach is that the porting process is entirely automated and normally requires no new development work at all. It is implemented via the PathPort Workbench.

The Workbench provides a GUI-based point and click interface to the whole porting process. It is implemented as a Windows application running on a PC workstation designated as the development workstation. The Workbench communicates over a network to a Tandem host resident server process (the Workbench server) which accesses the Pathway application's source code library.

Using the simple point and click user interface, individual Screen COBOL modules are selected for inclusion in the new Windows requester, preprocessed on the host to incorporate copy libraries, downloaded to the Workbench, and then cross compiled. The result of cross compilation is C++ source code and PowerBuilder source code.

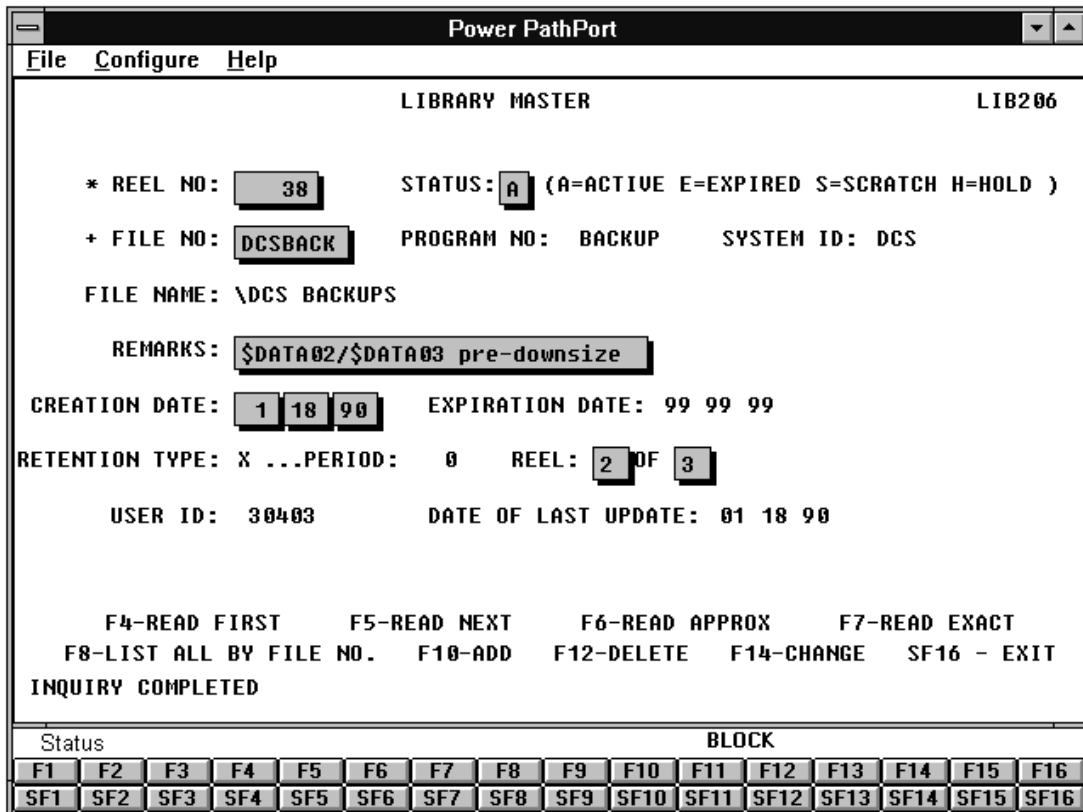
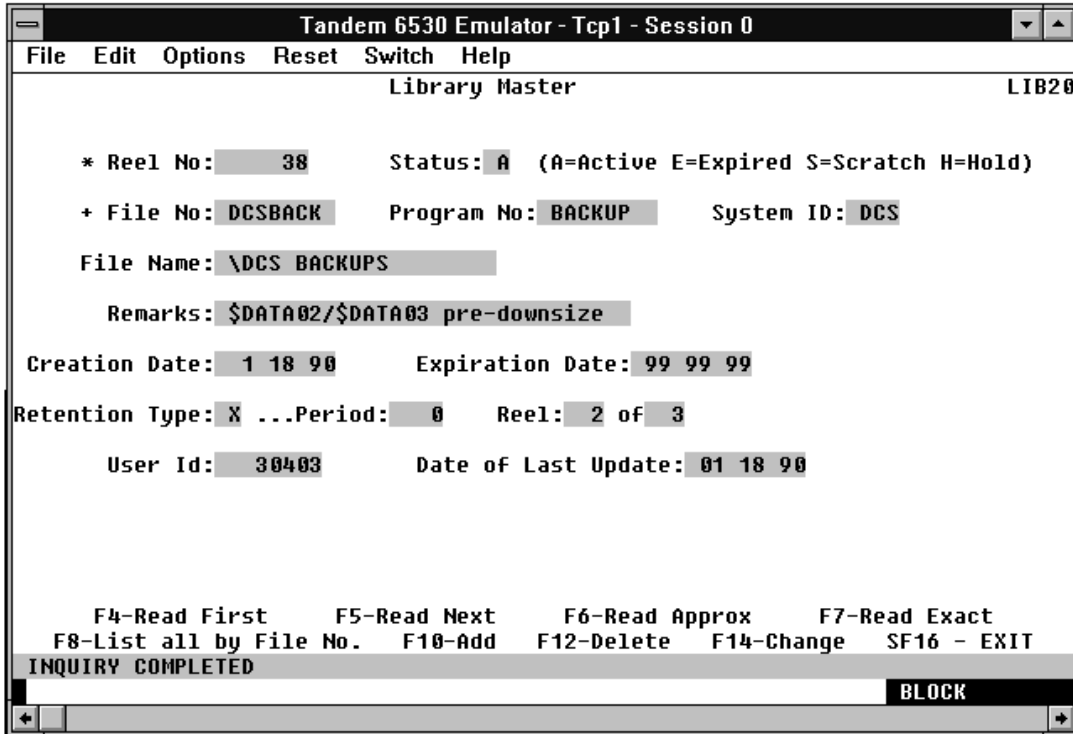
The C++ source code implements all requester logic, excluding screen presentation. Another simple Workbench mouse click causes the C++ source code to be compiled, yielding a ready to run Windows DLL.



Porting is accomplished via a simple point and click interface.

The PowerBuilder source code implements all GUI screen presentation functions. It is imported into the PowerBuilder product, where screens can be previewed. The PathPort provided Style dialog permits the customer to separately control the colors, size, font, and border for all text, input fields, buttons, and boxes. After style selections are made, the screen presentation is converted to executable form by the PowerBuilder compiler.

This combination of the DLL and PowerBuilder executable represents the new Windows requester.



Power PathPort automatically converts the legacy terminal presentation (top) to a default GUI presentation (bottom).

Note that the PathPort WorkBench has been designed to integrate the application's existing host resident source code library with the entire porting process. This eliminates concerns regarding the existence of multiple copies of the source code on different machines.

Additionally, the maintainability of the Windows requester is easily accomplished. Should any SCOBOL module be subsequently modified, the new module can be easily incorporated into a new Windows requester using the Workbench. The revised module is simply ported again using the same Workbench point and click interface and linked into a new Windows requester.

## GUI Enhancement And Customization

The Power PathPort product provides automated conversion of SCOBOL screen presentations that are significantly improved over the legacy block mode terminal style that is inherent in a Pathway application.

As described above, Power PathPort automatically generates a GUI version of the old Pathway screen, and allows customer control over the display of the resulting objects.

However, many customers desire to further enhance the GUI presentation to their applica-

The screenshot shows a window titled "Library Master" with a menu bar containing "File", "Configure", and "Help". The window is divided into several sections:

- Reel:** Number: 38, 2 of 3.
- Status:** Radio buttons for Active (selected), Expired, Scratch, and Hold.
- File Information:** File Number: DCSBACK, System ID: DCS, Program Number: BACKUP, User ID: 30403, File Name: \DCS BACKUPS, Remarks: \$DATA02/\$DATA03 pre-downsize.
- Date:** Creation: 01/18/90, Expiration: 00/00/00, Last Update: 01/18/90.
- Retention:** Period: 0, Type: Permanent (dropdown).
- Read:** Buttons for First, Next, Approx, and Exact.
- Maintain:** Buttons for Add, Delete, Change, and List All.
- Exit:** A button at the bottom left.

The status bar at the bottom of the window displays "Status" and "BLOCK".

*Customers may rapidly and easily customize the default GUI presentation automatically generated by Power PathPort.*

tions. A key feature of the Power PathPort approach is its ability to quickly accommodate these enhancements using the industry standard PowerBuilder GUI development tool.

The default GUI presentation of the SCOBOL screen generated by Power PathPort acts as the starting point. Using the standard PowerBuilder Window Painter, the screen presentation may be enhanced to provide the user a customized GUI to the application.

Both protected and unprotected fields may be rearranged. Drop down list boxes may be added to any input field. Input fields may be converted to radio buttons. Function keys may be reimplemented as buttons or even icons.

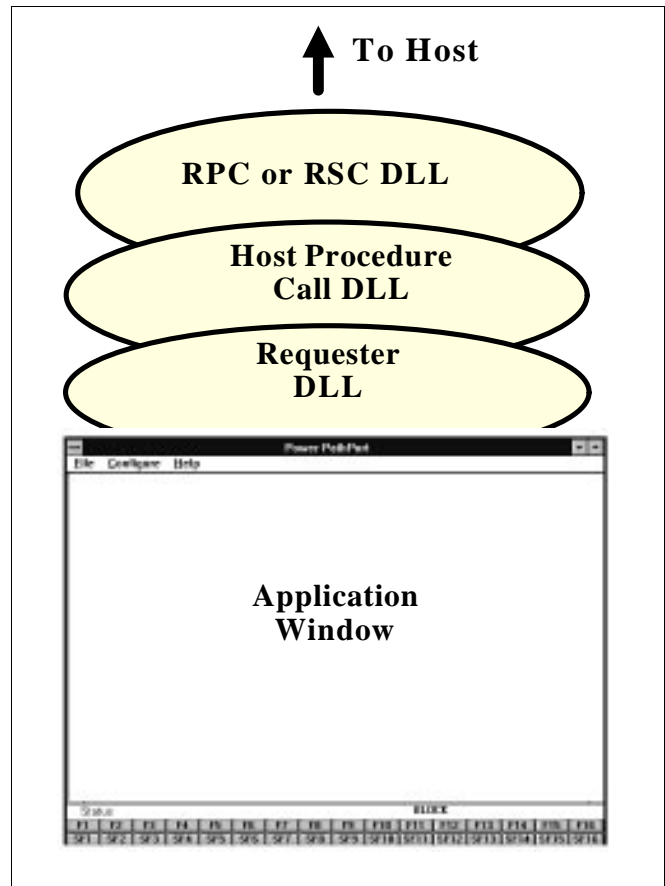
The entire appearance of the application can not only be revised as desired, but done so easily and rapidly.

## The Run Time Environment

After the porting process and optional GUI customization has been completed, the new Windows based application requester must be installed on user workstations along with PathPort run time components that support host communications from the new requester.

The PathPort run time environment has been carefully designed to provide complete environmental transparency.

Both workstation and host resident PathPort run time components cooperate to provide this entirely seamless interface between the new requester and existing application servers. As a result, the application never requires modification.



*Workstation run time components provide a seamless interface to the host servers.*

### *Workstation Run Time Environment*

The run time environment is comprised of a number of components that provide all required support for the newly created GUI requester.

The Application Window is the PowerBuilder generated executable that implements the screen presentation functions. It may be either the default presentation directly generated by Power PathPort or a customized and enhanced version.

The Requester DLL implements the requester logic. It is automatically generated by Power PathPort during the porting process as discussed above.

The Host Procedure Call DLL implements all logical interactions with the host. This component provides support for the sending of transactions to the Pathway server, receiving responses from the server, and all TMF transaction protection.

The Remote Procedure Call DLL provides the actual low level remote procedure call implementation. It may be either Open Systems Standard RPC/XDR (provided by Failsafe) or Remote Server Call.

Multiple applications may be concurrently executed on the same user workstation to either the same or different Pathway applications.

Since the PathPort solution heavily utilizes DLL's, all application Windows share the same copy of all DLL's, conserving workstation memory.

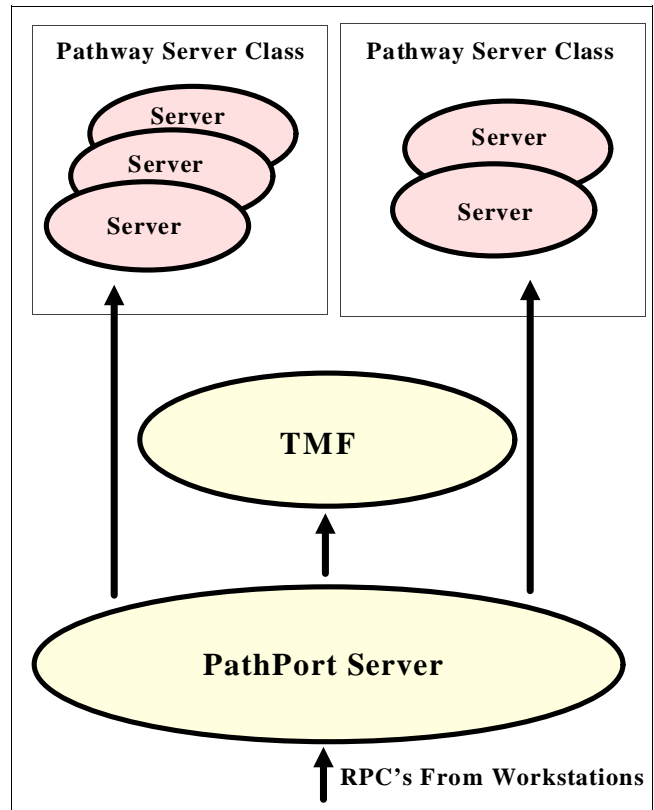
### *Host Run Time Environment*

The host run time environment provides complete transparency for the application's Pathway servers.

The principal component of the host run time environment is the PathPort Server, which is the recipient of the individual workstations' requests for host services.

These requests may be presented to the PathPort Server using either Open Systems Standard RPC/XDR or Remote Server Call (RSC).

The PathPort Server dispatches all transaction requests to the appropriate Pathway server class and returns the responses to the workstation based requester.



*The PathPort Server provides RPC support for workstation initiated Guardian procedure calls.*

Additionally, if the application requires TMF transaction protection, the PathPort Server automatically performs the required TMF operations.

All interprocess messages between the servers and workstation based requesters retain the same format as before conversion. The servers are completely unaware of the fact that the requester is now executing as a Windows application.

An additional feature of the PathPort Server is that it also provides support for the SCOBOL print screen request. When this verb is employed in the requester, the screen image is passed to the PathPort Server, which in turn sends it to the Spooler for actual printing.

The system also features configuration analogous to Pathway terminal configuration. The end user customer may specify for each client workstation items such as initial program name, default printer, and default PATHMON name.

## Open Systems Standards

PathPort was initially implemented using the Open Systems Standard TCP/IP protocol suite. TCP/IP was chosen as a result of the fact that as a true standard it allows complete connectivity between differing computer systems.

The ability to perform remote procedure calls is of critical importance to the PathPort solution. The Open Systems Standard ONC RPC/XDR was chosen for this purpose. It is the leading remote procedure call mechanism in use over TCP/IP networks. The combination of its technical and commercial maturity as well as its wide adoption and deployment assures complete interoperability with a wide variety of hosts, workstations, and PC's.

RPC/XDR is a recognized and formal industry standard, employing no proprietary or nonstandard technologies. It provides a true peer to peer design, where any system may act as a client, a server, or both. This is an important consideration with respect to the handling of Pathway unsolicited messages, where the host must act as a client instead of its normal role as a server.

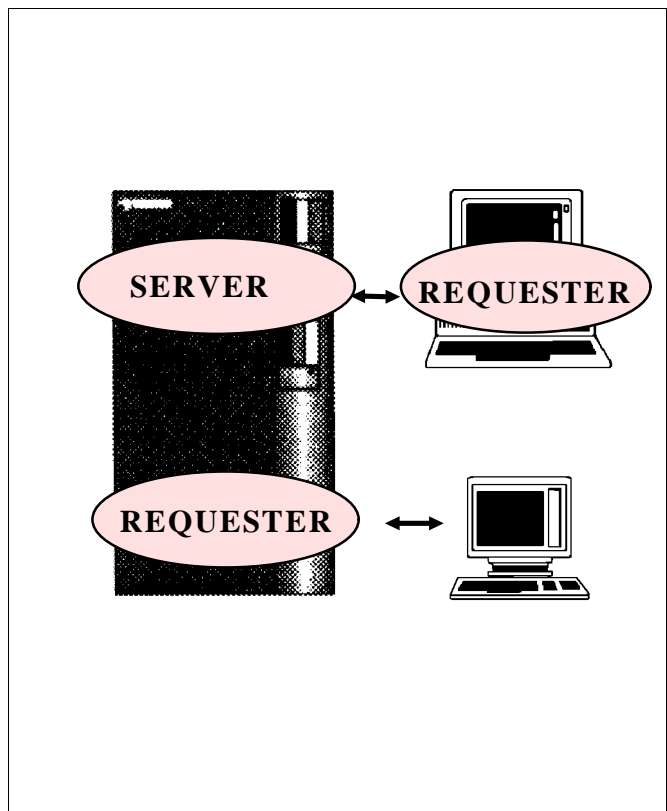
PathPort is also available in a version which uses proprietary Remote Server Call (RSC), which confines the host to a server role. The RSC version is an appropriate choice where there is no emphasis on Open Systems Standards, or where the underlying protocol stack is not a standard such as NETBIOS local area

networks or asynchronous point to point connections.

## Product Benefits Summary

Power PathPort represents a true breakthrough in client/server technology for Tandem users. It offers benefits summarized below.

User productivity is significantly improved through use of a GUI instead of the legacy block mode terminal style of interface. The product allows customer control over its precise design. The Power PathPort generated default GUI may be employed as the first step and subsequently enhanced by the customer to the extent desired.



*Modern PathPort workstations may concurrently utilize the same servers as legacy terminals.*

Existing application investment is fully protected. All currently employed SCOBOL source code is fully utilized; it is merely converted to a more usable form employing modern client/server and GUI technologies. Pathway server source and object code remain in their current form and require absolutely no modification or translation. In fact, the product also protects any existing investment in dumb terminals. Since no server changes are required, modern PathPort workstations may concurrently utilize the same servers as legacy terminals, permitting gradual migration over time.

Greater economy of operation and improved response time are achieved in a variety of ways. Host cpu consumption is reduced by the elimination of the cpu consumptive task of

SCOBOL pseudo-code interpretation by the Pathway TCP. In addition, both the conversion between terminal and internal data formats and terminal protocol support are also eliminated.

Application maintainability is enhanced since the product complies with both formal and defacto standards in protocols, remote procedure call technology, programming language, and windowing system. Proprietary legacy technologies are obsoleted.

Exceptional return on investment is provided by the Power Pathport solution. Legacy Pathway applications may be converted to client/server and GUI at a mere fraction of the time and cost associated with a reimplementation strategy.

PathPort and Power PathPort are trademarks of Failsafe Computer Systems, Inc.  
PowerBuilder is a trademark of Powersoft Corporation.

All other brand and product names mentioned are trademarks or registered trademarks of their respective companies.



**FAILSAFE SYSTEMS**

*"Excellence in Action"*



9950 W. Lawrence Ave., Suite 202, Schiller Park, Illinois 60176  
Phone (847) 671-5700 FAX (847) 671-2988  
info@failsafe-systems.com <http://www.failsafe-systems.com>