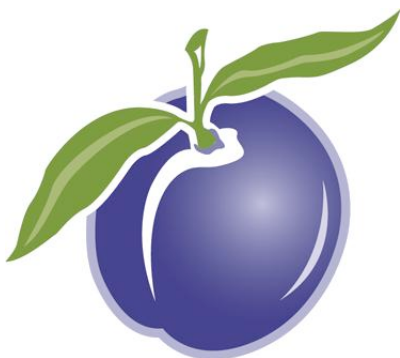


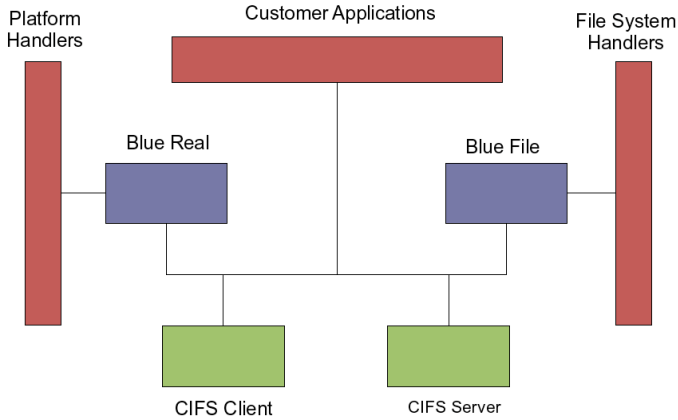
BLUE SHARE™ CIFS FILE SHARING SOFTWARE DEVELOPMENT KIT



The Blue Share™ Software Development Kit by Blue Peach™ provides advanced, portable, and high performance network file sharing compatible with Windows™ Operating Systems (Windows XP™, Windows NT™, Windows CE™, Vista™) and Linux™ or Unix Operating Systems running Samba. Blue Share™ can be configured as a File Sharing Client (accessing files on a remote system) or as a File Sharing Server (systems on a network can access files on a local file system). Blue Share™ is built on top of the Blue Real™ Real-Time Embedded Application Manager. Blue Real™ provides platform abstraction, state handling and configuration to applications. Blue File™ extends

the architecture with a file system multiplexer that presents multiple file system APIs to applications, performs path mapping, and redirects file system operations to platform dependent file system handlers.

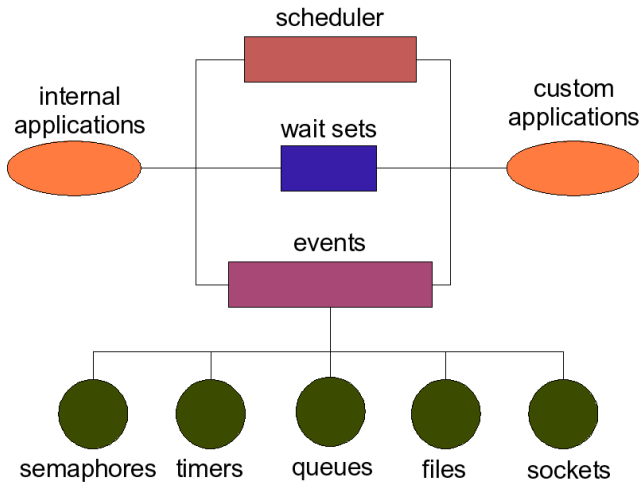
Components of this architecture have been designed to work together. This allows a very compact, yet functional SDK.



Blue Share Architectural Components

BLUE REAL™

Blue Real™ has been designed with the notion that real time applications are state driven. State machines are comprised of states, events, transitions, and actions. A feature of Blue Real™ is the scheduling of applications based on events, and the reliable delivery of those events to the applications.



Blue Real Components

Given this event delivery mechanism, complex state-drive applications can be easily designed and implemented. Customer applications can be implemented as Blue Real applications or as regular platform applications.

PLATFORM HANDLERS

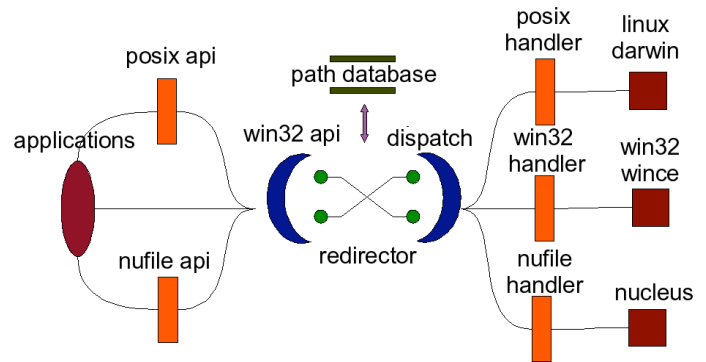
Blue Real™ provides platform abstractions for:

- Windows™ Desktop (WinXP, WinNT, Vista™)
- Linux™ and Darwin
- WinCE™
- Nucleus™ PLUS

Blue Real™ can be ported to other platforms not listed. Since Blue Real™ supports Linux™ and Windows™, applications written to Blue Real™ can be targeted to an embedded RTOS, yet unit, functional, and regression tested on a desktop development machine.

BLUE FILE™

Blue File™ provides generic file handling to embedded applications. It offers multiple file system APIs that transforms operations into a normalized form. File names are mapped through a path database that allows translation to paths relative to target file systems. Operations are then dispatched to target platform file system handlers.



Blue File Components

The normalized file interface is similar to a Win32™ API. This API is richer than other platform APIs and similar to the CIFS protocol. By normalizing around Win32™, Blue File can support most features of CIFS.

Existing file aware applications can be ported to Blue File™ by leveraging optional file APIs. In addition to the Win32™ API, Blue File™ provides APIs for:

- Posix™
- Nucleus™ FILE

Other APIs can be added as long as they conform to the Blue File™ redirector's API.

FILE SYSTEM HANDLERS

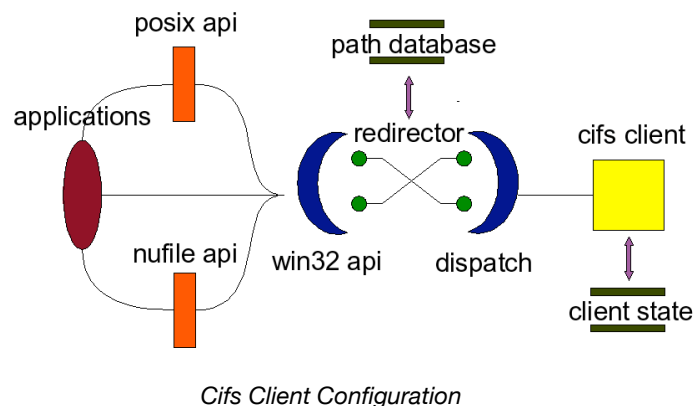
File System handlers provide the back end interface to the native file system interface of the platform. Blue File provides file system abstractions for:

- Posix™ (Linux™ and Darwin)
- Nucleus™ FILE
- Win32™

Other file system abstractions can be added. Some useful abstractions include a ram file store, persistent flash storage, pipes or other virtual file systems.

CIFS CLIENT

Blue Share™'s CIFS Client leverages the functionality provided by Blue Real™ and Blue File™. Architecturally, it attaches to Blue File™ via the redirector's dispatcher.

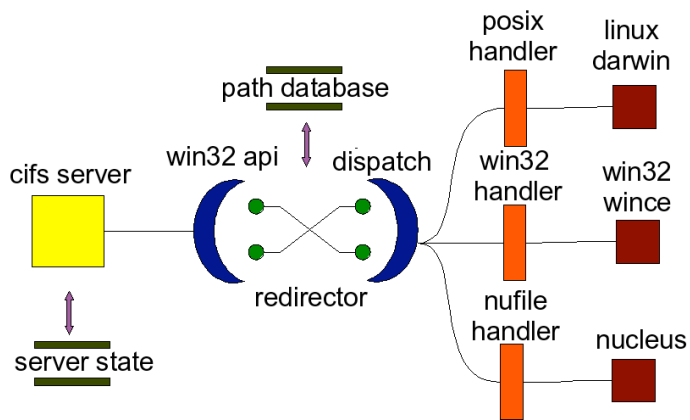


Since the CIFS client is layered below the redirector, any supported file system API can be used to interact with a remote CIFS server. File naming of CIFS resources use an extended Universal Naming Convention (UNC) or SMB URL format. These naming conventions embed server, share, and authentication information for the remote. The path mapping capability of the Blue File™ Redirector can be configured to translate drive letters or mount points to UNC or SMB URLs. Therefore, not only can an application access files over CIFS through any supported file system API, it can access these files transparently making simple ports of file aware applications to network file aware applications possible.

CIFS SERVER

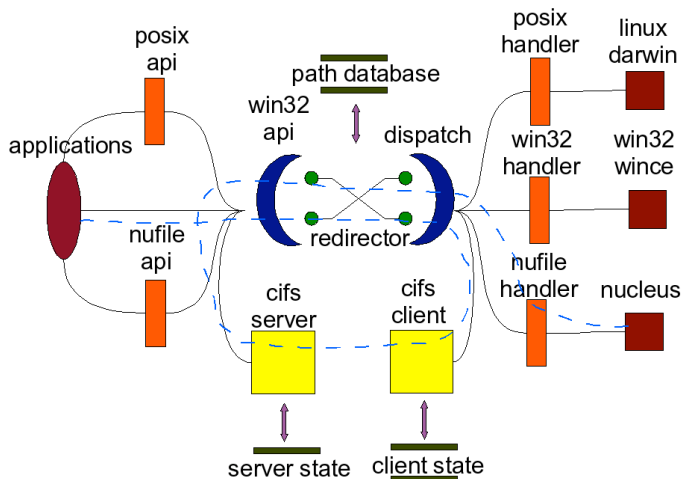
Blue Share™'s CIFS Server also leverages the Blue Real™ and Blue File™ components. It interfaces with the Blue File™ Redirector as another API.

File operations from the CIFS server can also be mapped through the redirector and targeted to any platform supported file system.



Cifs Server Configuration

One configuration that illustrates the flexibility of this redirector architecture is the case where one of the target file systems is a CIFS client. This would allow remote share to itself be shared through the CIFS server, essentially configuring Blue Share™ as a CIFS proxy.



Loopback Cifs Configuration

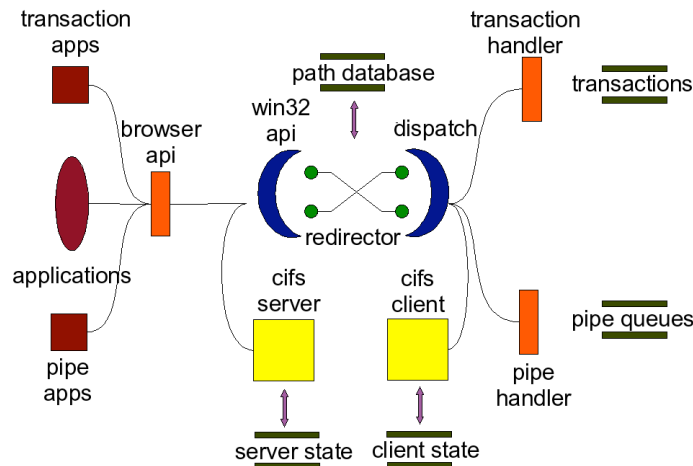
Another configuration useful for discussion is the loopback configuration. In a loopback configuration, the CIFS client accesses the local CIFS server through a network loopback.

The flow of interactions in this loopback configuration begins with an application accessing a file with a Nucleus™ FILE call. The NUFFile API translates the operation to a Win32™ operation and passes it on to the Blue File™ Redirector. The redirector maps the path to an SMB URL and determines that the file is on a remote system. The server, share, and authentication information is provided to the CIFS client through the SMB URL. The CIFS communicates to the remote CIFS server through a network connection (TCP, NetBIOS). In a loopback configuration, the server name in the SMB URL species the localhost. The server receives the share name and authenticates the user. The server maps the share name and file path to a logical location using information on that share in the server state. This logical location is passed to the File Share™ redirector. The path is

mapped to a location within a Nucleus™ FILE disk on the local system. Blue File™ passes the operation to the NUFile Handler. This handler receives the operation through a Win32™ dispatch, translates the operation to a NUFile operation and issues the call to NUFile.

NETWORK BROWSING AND MANAGEMENT

Network Browsing and Management in a CIFS network uses applications layered above CIFS to communicate to remote CIFS applications with an RPC mechanism referred to as DCE. DCE utilizes networked pipes and transactions. Pipes appear as files on special shares in a CIFS server. Transactions provide a packet based communications facility between applications.



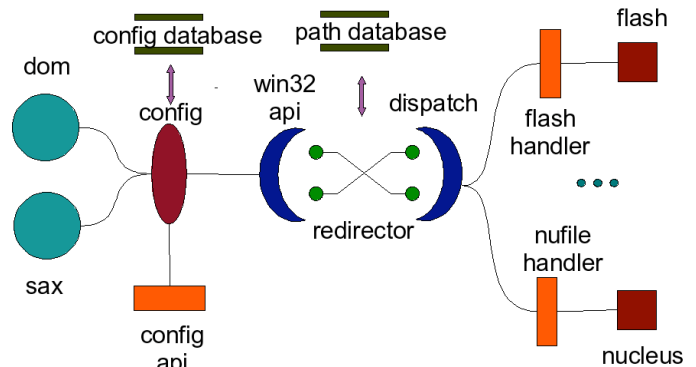
Browser Configuration

Since browsing is layered above CIFS, browsing operations look very similar to any other CIFS operations. An application issues RPC calls or transactions to the browser API. These operations are translated into Blue File™ operations and passed to the Blue File™ redirector where they are mapped to a URL that specifies a network pipe to be handled by the CIFS Client. The CIFS Client and Server treat the operation as an opaque file operation. The server takes the operation, translates it to a logical path and passes it to the redirector which remaps it to a local pipe. The handlers buffer transactions and pipe data and manages reads and writes on the channels to provide the RPC mechanisms.

The Blue Share™ CIFS Server provides applications to provide browsing and management services.

CONFIGURATION

Blue Share™ offers an extension for Blue File™ that manages extensible configuration information. This configuration subsystem utilizes a model based on stored XML. The XML schema is application specific although Blue Real™, Blue File™, and the CIFS Client and Server have defined schemas for their own persistent data.



Persistent Configuration

The configuration application accesses the xml file through the Blue File™ redirector. The file can reside on a platform supported file system, or can be stored in flash or some other non-volatile location. In the case of non-volatile storage, an NVRAM handler can be added that will receive Win32™ dispatches from the redirector and translate them to NVRAM operations.

The XML data is parsed and stored as a DOM data structure. Applications can access this configuration data using the DOM API, or can access it through an API that accepts named elements similar to a registry. For instance, the local hosts IP address may be stored in the configuration database as:

`/BlueShare/BlueNet/IP`

The IP address can be retrieved by simply requesting the value using the path and name.

Copyright © 2008 Blue Peach Solutions, Incorporated. All rights reserved. Blue Peach, the Blue Peach Logo, Blue Share, Blue Real, and Blue File are trademarks or registered trademarks of Blue Peach Solutions, Inc or its affiliates in the U.S. and other countries. Win32, Windows XP, Windows NT, Windows CE are trademarks of Microsoft Corporation in the United States, other countries, or both. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Nucleus is a registered trademark of Mentor Graphics in the U.S. and other countries. Posix is a registered trademark of the Institute of Electrical and Electronic Engineers in the U.S. and other countries



Blue Peach
 2 Wyeth Circle
 Southborough, MA 01772
 857-205-9315 (857-BLU-PEACH)
info@bluepeach.com
www.bluepeach.com