

Whitepaper: Wavelink Studio

How It Works

Revised 9/10/02



INTRODUCTION

Many organizations find that the benefits of a wireless network are too numerous to ignore. The ability to create a mobile workforce that builds upon an existing network is just one of many incentives that compel many organizations to “go wireless.” However, most organizations still face a significant hurdle when attempting to integrate wireless technologies: the costs associated with designing, deploying, and maintaining wireless applications. Consequently, there is a strong desire to incorporate wireless solutions that leverage the existing capabilities of IT managers, developers, and other network personnel.

Wavelink Corporation addresses this dilemma by offering Wavelink Studio —a powerful collection of development libraries, server software, and client applications. Studio allows organizations to rapidly deploy and develop wireless solutions that are either extensions of existing applications, or newly created ones.

OVERVIEW OF STUDIO COMPONENTS

Wavelink Studio is a comprehensive set of tools that allow organizations to migrate existing solutions into a wireless environment. These tools fall into three main categories: development libraries, server components, and clients.

Wavelink Studio development libraries are collections of objects and methods designed to handle the intricacies of wireless networks and mobile devices. The following libraries are available: COM (which supports Visual Basic and other languages) and Java. Because Wavelink Studio contains multiple libraries, developers can use a programming language of their choice and be assured that there is a Wavelink library that can help them handle the various wireless network and mobile device functions.

A number of server components are included with Wavelink Studio, with each component specializing in a specific aspect of wireless application management. These components are designed to work seamlessly with each other to ensure that wireless applications are managed correctly. While several of these components are described in this document, more complete information can be found in the Wavelink Studio product documentation.

The third component of Wavelink Studio is its collection of Wavelink Clients. These clients act as a bridge between server-side applications and an end-user. Each client is designed specifically for a type of mobile device while simultaneously supporting Wavelink Studio development libraries. As a result, application developers can focus more on functionality and less on accommodating the various characteristics of mobile devices. With Wavelink Clients, a single application can operate on multiple mobile devices.

INCORPORATING WAVELINK LIBRARIES INTO APPLICATIONS

Wavelink libraries reduce the time it takes to create a wireless application by allowing developers to separate the logic of an application from the presentation of that application. The logic of an application is defined as the underlying functionality that

allows it to accomplish its designed tasks. These functions can include communicating with databases, compiling reports, sending information to a printer—anything that helps the application solve a user’s need.

An application’s presentation focuses on the ways in which information is retrieved from and displayed to an end user. Parts of an application’s presentation include the buttons, fields, message boxes, and other components that help the user work with application data. An application’s presentation typically does not control the core functions of the application as a whole; however, it does interact closely with the application’s logic.

Wavelink libraries allow developers to retain the logic of their applications. Without Wavelink Studio, developers must entirely re-design an application to work on a mobile device. By incorporating a Wavelink library into an application, developers only need to create a new presentation component for use with mobile devices. Further, the developers can design the new presentation to work with multiple mobile devices in a way that takes full advantage of device-specific capabilities.

Incorporating Wavelink libraries into an application is a straightforward process. Developers need only to open a new or existing development project, using the programming tools of their choice. The developers can use their programming tool to reference the Wavelink library, located in the `\lib` folder of the Wavelink Studio working directory, to gain access to its objects and methods.

WORKING WITH WAVELINK CLIENTS

One of the challenges of a wireless network is the lack of standards for mobile device functionality. Instead, each hardware manufacturer designs mobile devices differently. These variations in design can hinder the deployment of a wireless solution, because developers must consider how the application will operate with different mobile devices.

Wavelink Clients remove this obstacle because each client supports the objects and methods of Wavelink Studio development libraries. Developers writing code with one of these libraries can be assured that their applications will work on any mobile device on the network. With a constantly expanding collection of Wavelink Clients, Wavelink Corporation assures organizations that a client is available for almost any mobile device an organization uses.

Configuring Wavelink Clients is also a straight-forward process. Because Wavelink Clients integrate with the built-in operating systems of mobile devices, organizations can use the existing features of those systems to configure the devices for their network. While the configuration process varies for different device types, most devices require changing the following parameters:

- network identifier (such as an ESSID)
- device IP address
- network subnet mask
- network gateway (or router) IP address

Wavelink Clients include two additional parameters to this configuration process: a host IP address and a port number. These parameters direct the mobile device to connect to a specific location on the network. Organizations can configure these parameters using the Wavelink Client. Note that, with Wavelink Studio, these parameters are not necessary when organizations employ the Wavelink Studio AutoDiscovery feature. This feature enables the display of available applications on the mobile device and eliminates the need to “point” the device to a specific host IP address and port number. The Wavelink AutoDiscovery section of this document describes this feature in more detail.

■ WAVELINK AUTODISCOVERY

One of the most versatile features of Wavelink Studio is AutoDiscovery. AutoDiscovery allows devices to connect to one or more applications without knowing the IP address and port number of the host system.

As mentioned in previous sections, a standard way of connecting Wavelink Clients to a network is to assign them a host IP address and port number. This method, however, is not ideal for organizations whose users need to connect to multiple wireless applications. Also, if large numbers of mobile devices are deployed within an organization, changing the host IP address and port number becomes a challenging administrative task.

Wavelink AutoDiscovery bypasses the restrictions of direct connections by replacing the host IP address and port number with a Discovery Token. When a Wavelink Client attempts to connect to a network, it sends a UDP broadcast that contains this Discovery Token. These tokens are also assigned to appropriate wireless applications on the network.

To receive AutoDiscovery broadcasts, Wavelink Studio employs a server component called a Discovery Server. These server applications reside on each network segment that contains a wireless component. When a Discovery Server receives a UDP broadcast that contains a Discovery Token, it responds with a collection of wireless applications that contain the same Discovery Token. Users can then select and activate one of these applications from this list.

When organizations add applications to Wavelink Studio, they determine if the application is anonymous or requires a signon. Anonymous applications are sent to Wavelink Clients during an AutoDiscovery process. Signon applications are not sent to a client unless the end-user supplies a valid user name and password. This feature of AutoDiscovery prevents users from accessing certain applications without proper authorization. Consequently, organizations can use AutoDiscovery to reduce maintenance costs associated with direct connection configurations, while reserving specific applications for authorized use only.

A further advantage to AutoDiscovery is that it operates on an enterprise-wide level. Discovery Servers provide this capability because they can locate applications outside the local network segment. This capability allows organizations to create repositories of applications that can be centrally managed and maintained.

LOAD-BALANCING

A common concern among organizations is controlling wireless traffic to prevent network components from overloading. To address this need, Wavelink Studio includes a process that helps spread wireless connections across a network.

For Wavelink Studio load-balancing to be effective, organizations must take the initiative to provide redundancy for their wireless applications. This redundancy includes the use of multiple host systems for highly-used applications. Once these multiple installations are completed, Wavelink Studio automatically tracks the number of current connections to each IP address and port number on the network. When a Wavelink Client attempts to connect to an application, the Discovery Server checks to see which installation application has the fewest connections and sends the IP address and port number for that application to the Wavelink Client.

The load-balancing feature of Wavelink Studio is only available if an organization uses the AutoDiscovery feature.

SESSION MANAGEMENT

Another feature offered by Wavelink Studio is the ability to re-connect users prematurely dropped from the network. These accidental disconnects can occur for a variety of reasons, such as device hardware failure or a mobile device re-boot, and can have a negative effect on the productivity of an organization's end-users.

Wavelink Studio handles session management by assigning a unique session ID to each active connection on the network. When an end-user ends a connection, the Wavelink Client sends a specific broadcast to inform the network. Once this broadcast is received, both Wavelink Studio and the Wavelink Client set the session ID for that connection to zero.

If a connection is lost prematurely, Wavelink Studio keeps the session ID active until the mobile device re-connects to the network. When the device attempts to re-connect, the Wavelink Client re-sends the session ID to the network. This ID is checked against existing session IDs to find a match. When a match is found, the Wavelink Client is connected to the same application to which it was connected earlier. As a result, the end-user is presented with the exact application screen displayed prior to the connection loss.

Note: This feature is not available on all clients.

CONCLUSION

Wavelink Studio is an extensive, versatile wireless solution that allows organizations to develop, deploy, and manage wireless applications on a multiple-server environment. The Wavelink Studio components include development libraries, server components, and Wavelink Clients—each of which work together to provide a comprehensive solution to wireless application development, deployment, and management.

By deploying Wavelink Studio, organizations can increase the effectiveness of their wireless networks while decreasing the costs of designing, deploying, and maintaining those solutions.



Wavelink Corporation
11332 NE 122nd Way, Suite 300
Kirkland, WA 98034
Main: 425.823.0111

www.wavelink.com

