



Enabling Real-time RIA with iPush® Server

White Paper

<http://www.icetechnology.com>

ICE Technology Corp., April 23, 2007

Doc. version: 2.3

iPush Server is a registered trademark of ICE Technology Corporation. All other trademarks contained herein are the property of their respective owners.

Contents

In the age of RIA.....	3
How iPush® enables real-time RIA.....	4
Easy development of real-time RIA with iPush®	6
Deploy real-time RIA with iPush®	9
iPush® Edition Differentiation.....	12
About ICE Technology Corporation	13
Contact	13

In the age of RIA

RIA, Rich Internet Application, is getting more attentions for service development on the web.

We will not go deep into the explanation of what RIA is, or what benefits RIA brings. Instead, we will directly give you the information about how developers can take advantages of iPush[®] Server to enable their RIAs with real-time data distribution capability.

Note: For learning more about RIA, please visit Wikipedia at http://en.wikipedia.org/wiki/Rich_Internet_Application

The coverage of RIAs here includes applications in Java Applet (.class), ActiveX Control (.ocx), and Flash movie (.swf). All of them can be run in the web browsers with respective run-time environments.

In the age of RIA, users expect the interface of application can bring enhanced experiences: more friendly, more fun, and more timesaving.

In the age of RIA, an on-line service may need to be accessed by variety of devices or applications which are developed with different programming languages.

RIA achieves timesaving by preventing web pages from reloading. But sometimes it is not good enough. In the cases of RIAs have data continually updated, the way of request-and-response (client-pulling) for retrieving newest data is still making lots of inefficiency.

Adopting server-push technique is a wise decision when your RIA needs to receive or send data in real-time fashion. The technique will present lower operation cost in bandwidth, server capacity, and higher service level of data distribution.

How iPush[®] enables real-time RIA

iPush[®] Server implements the server-push technique and designed for serving massive concurrent connections.

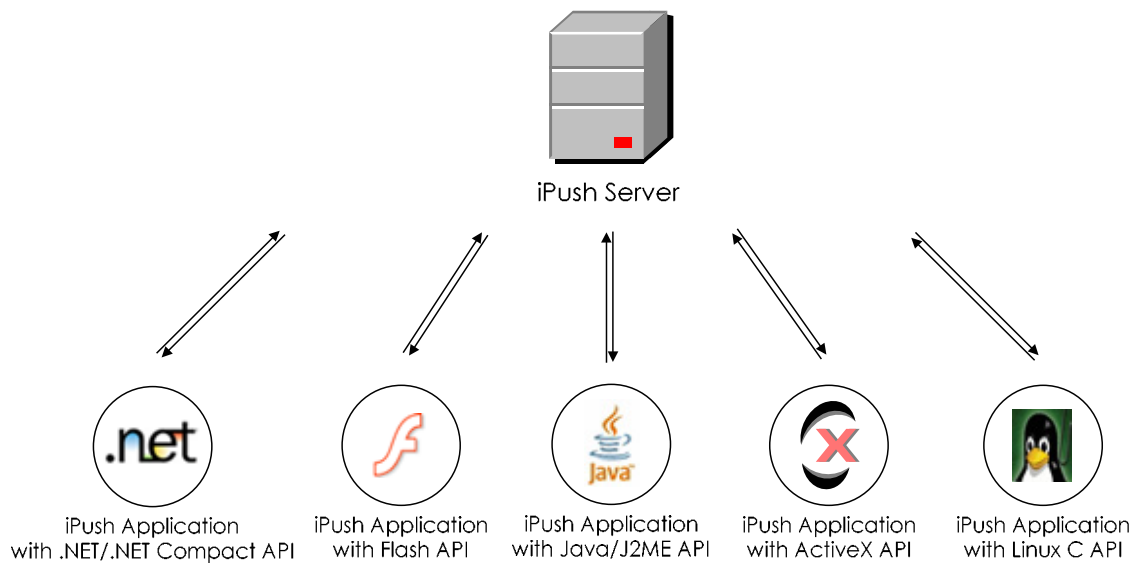


Figure 1. iPush provides variety of client libraries for application development

The native server-push feature of iPush[®] Server allows client application to asynchronously receive streaming data on subscription. Under the same real-time service, client applications, may be developed with different iPush APIs, can receive the subscribed data at the same time.

Every growing on-line service needs a scale-up plan, and the plan should cover both server side and client side.

For client side scaling up, it is quite common to provide both web application and Windows application for accessing the same on-line service. Moreover, porting

applications to support different devices, like from desktop to PDA or mobile phone, would be a good strategy for acquiring new users and providing better service for existing customers.

For server side scaling up, consider expanding service capacity painlessly and nonstop, the configure-to-add feature for adding a new host to the serving cluster will be important for the service operator.

iPush[®] Server presents a good scaling up solution for client side and server side respectively. Variety of iPush[®] client APIs help the development of various real-time rich applications that run on different environments. Feature of native message up-linking between servers and clusters makes the expansion of service capacity seamless and painless.

Easy development of real-time RIA with iPush®

When you got iPush® Server installed and run, you can choose any iPush® client library to develop real-time RIA. For the time being (June 2006), ICE Technology provides:

- ❖ ActiveX Control for Windows 32
- ❖ ActiveX Control for Windows CE
- ❖ Java Class for Java Applet or Java Application
- ❖ J2ME Class for Java Midlet (proven on Nokia N70 and SonyEricsson Z800i Java Phone)
- ❖ Flash Component Library for ActionScript 2
- ❖ .NET Framework Library
- ❖ .NET Compact Framework Library
- ❖ Windows DLL
- ❖ Linux C Library
- ❖ uClinux 2.6 C Library
- ❖ JavaScript Wrapper - PJAX

Note: An on-line function list of iPush® V2 client APIs is available at <http://www.icetechnology.com/developer/functions.shtml>

By straightforward function calls, the development of iPush® application for real-time data sending and receiving can be extraordinarily simple.

Figure 2. illustrates the real-time data distribution flow with iPush® Publish/Subscribe messaging model. Table 1. lists the programming skeleton of data sending and receiving with relative function calls.

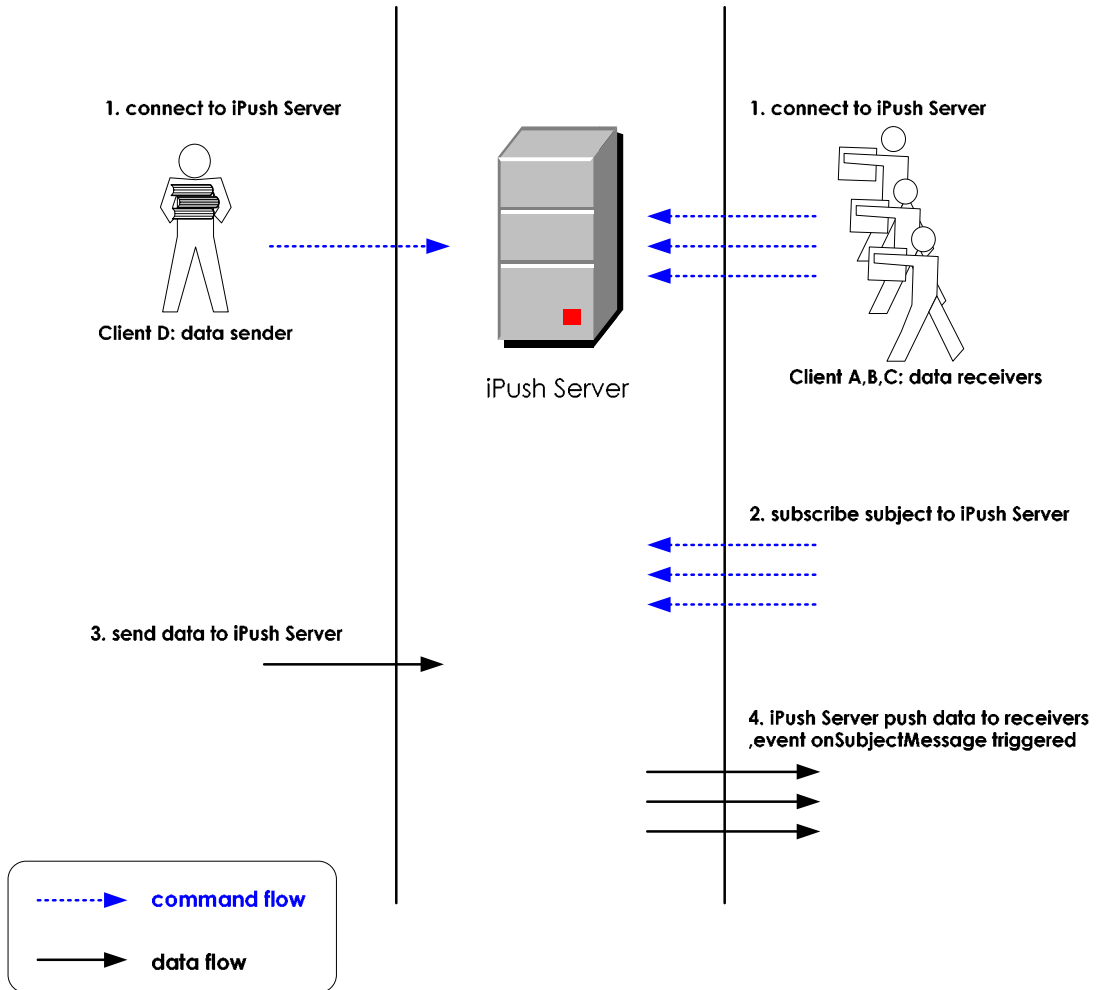


Figure 2. The real-time data distribution flow with iPush Publish/Subscribe messaging model

Table 1. The programming skeleton of data sending and receiving with iPush API

	Data sender programming skeleton	Data receiver programming skeleton
1.	ipushTCPConnect() Connect to iPush Server with TCP.	ipushTCPConnect() Connect to iPush Server with TCP.
2.		ipushSubSubject() Subscribe specific subject to iPush Server.
3.	ipushSendNPSubjectData() Send data to iPush Server with specific subject.	
4.		onSubjectMessage() After receives the data pushed from iPush Server, the client API will trigger this event for data handling in application.

After subscribes specific subject to iPush[®] Server (finish 2.), the data receivers can process their own tasks without waiting for the coming data. When data pushed from iPush[®] Server, the callback function **onSubjectMessage()** will be executed in event-driven fashion. This is why we say the iPush[®] application development would be an asynchronous programming style.

Follow the Publish/Subscribe (Pub/Sub) programming skeleton with iPush[®] to implement a 1-to-N real-time data broadcasting service is quite simple. An experienced developer definitely can work out an iPush[®] application within several hours.

Deploy real-time RIA with iPush®

There are two deployment types for iPush® application:

Type 1: **Standalone**

Type 2: **Embedded in browser (web application)**

A standalone iPush® application means it have to be installed at client side first, then start to run with real-time functionality. The standalone iPush® application communicates with iPush® Server directly, and the iPush® Server can be run on any host and listen to any port, such as port 80 or 443 to be firewall-friendly. Figure 1. has illustrated the deployment of standalone iPush® applications.

An embedded iPush® application (web application) in browser means it have to be downloaded before every time use. The download of iPush® web application is completed by web server and browser through HTTP, then the iPush® web application initialized and starts to communicate with iPush® Server in the respective run-time environment, such as Java virtual machine or Flash Player plug-in.

Under the security restriction of Java run-time environment, the iPush® Server requested to run on the same host with the web server which provides download of iPush® Java Applet. But iPush® web applications for Flash or ActiveX control wouldn't.

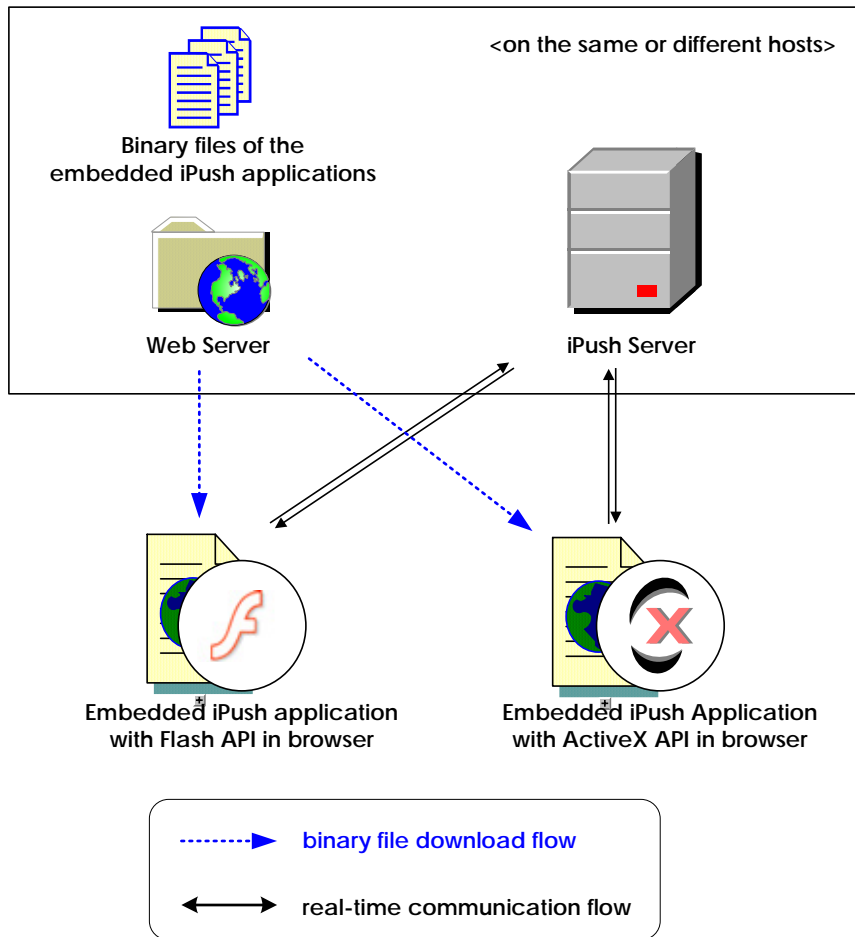


Figure 3. The typical deployment for iPush web applications (Flash and ActiveX)

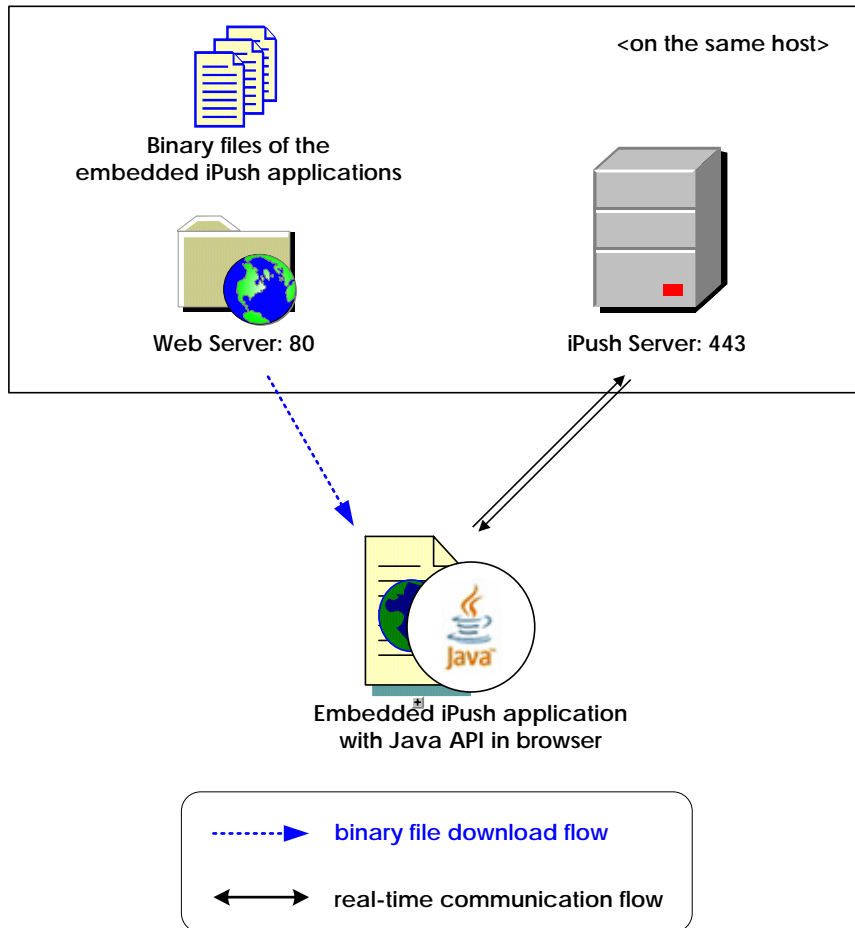


Figure 4. The typical deployment for iPush Java Applet

As Figure 4. shows, usually, the web server will listen to port 80 by default and iPush[®] Server can be configured to listen to port 443 on the same host to keep firewall-friendly. When web server and iPush Server are run on different hosts, both can be configured to listen to port 80 to keep firewall-friendly.

iPush® Edition Differentiation

There are two standard editions of iPush® Server V2:

1. iPush Server Enterprise
2. iPush Server Millions

The major differences between these two standard editions are list in Table 2. below.

Table 2. iPush edition differentiation

	iPush Server Enterprise	iPush Server Millions
Guaranteed delivery	Enabled	Disable
Cluster edition	Provided by request	Native

C.C. = Concurrent Connections

Guaranteed delivery is a feature that allows offline subscribers to receive the stored data (in server) when they go back on-line. This feature also introduces the acknowledgment of transported persistent message between sender-to-server and server-to-receiver; it gets the result of providing higher level of data distribution service.

We may say that the standard license of **iPush Server Enterprise** is aimed at the real-time data distribution services among tens to hundreds of concurrent users and they require guaranteed delivery; **iPush Server Millions** is positioned for on-line data services with thousands or more concurrent users and it is a clustering edition.

About ICE Technology Corporation

Founded in April 2000, ICE Technology is a leading provider of real-time information distribution software and relative solutions for real-time services.

ICE Technology delivers the most cost-effective information distribution server family, iPush Server, and provides solutions for various vertical markets' needs, such as financial service, sports, gaming, mobile service, and industrial automation.

Please visit <http://www.icetechnology.com> for more corporation details.

Contact

Headquarter

15F-1, No. 81, Sec. 2, Chengde Rd., Taipei, Taiwan 103

Tel: +886-2-2558-6101

Fax: +886-2-2558-6102

Business contact mail: sales@icetechnology.com

Support contact mail: support@icetechnology.com