How to web-enable a Jade application and start a browser session over the web[1]

Abstract: This document describes how to enable a Jade application to deliver its output over the world wide web to a client web browser. Any web-capable computer may then access the Jade application.

Introduction

When a Jade schema has been created its application(s) may be web-enabled.

For example, having created a Schema called *MicroUniversity*, containing an Application called *MicroU*, we may add to the schema another application called *MicroUWeb* and enable it to publish its output to client web browsers. Both *MicroU* and *MicroUWeb* will share the same schema, that is the same database of objects. Both may deliver the same functionality with the same or similar forms and methods.

To web-enable an existing application use the **Applications Browser** to **Add** the new application, *MicroUWeb*, and for **Application type** select **web-enabled**.

The new application begins with the functionality of the original *MicroU*. It will access the same forms, classes and so on. The existing forms of the original application will be available under the new application name and will be accessible by any standard web browser. Any computing device in the world that can run such a web browser then becomes capable of being a remote client able to run the Jade application, on your local server. That server's Jade system, acting in concert with a web-server program, will generate the forms as HTML code and send them across the internet to the remote browser on demand.

At the same time as remote clients are accessing the schema via *MicroUWeb* other local, non-web, LAN-based clients may be accessing it via *MicroU*. As the database is updated by one user (local or remote) the forms displayed to another user (remote or local) user will update accordingly.

Limitations of present web browsers

Unfortunately, current web browsers themselves have limitations which must be taken into account in the design or redesign of the forms to maximise the effectiveness of the application.

Therefore, the Jade forms displayed by the web browser will be more attractive and more functional if they were originally made with the web in mind or are re-made as a cloned set adjusted for web presentation. Note that . . .

1. The following Jade controls are currently not supported by a browser:
   - *Browse button* controls,
   - *Horizontal and vertical scroll bar* controls,
   - *Progress bar* controls
   - *OLE* controls, and
   - *Multimedia* controls (unless the media file is local to the client web-browser machine).
2. **Click** is the only control method supported. (**dragOver** and so on are not supported). Moreover the following controls do not execute their click methods: **CheckBox**, **OptionButton**, **TextBox** and **ComboBox**. These are serious limitations, imposed by browsers, not by Jade.
3. In the nature of HTML web pages there has never been any *modal* support for forms.[2] For example, a **MessageBox** is not a small modal pop-up; it occupies a full web page. The **.show**

[1] This appendix is the file *WebAwareJadeApplications.doc*.

and **.showModal** methods behave identically—all forms are displayed as web pages and web pages are just not modal.

4. A request received from a web browser for a fresh web page is simply a context-free URL.[3] In a receiving web server there is normally no knowledge of the previous request—the context of the session.[4] To address this problem Jade provides a unique *session ID*, which is included on every form that is sent to the remote client. The Jade code can examine this session ID in returning URLs to determine the context of a request for a page.
5. Try to ensure that every mouse click does not generate a complete screen refresh, with all the internet traffic that that entails.

Form design: When a new form is written in an application the **Painter** provides three options for the **Form style**: **Screen**, **Printer** and **Web**. The **Web** option does not of itself create forms unique to the web.[5] Its purpose is merely to assist you to design screen forms that will look more satisfactory when viewed with a browser. For example, a coarser grid is employed. Avoid the controls and control events that browsers do not support. For example, rather than display a *CheckBox* control employ a button and use its click event to alter a confirming text field or label field.

The *Web Server* and the *Jade Server*

Any web browser can access a remote Jade application if that application has been web-enabled.

The harder set-up task lies at the server end. Web browsers communicate with a *web server*. This computer and computer program combination manages the internet traffic by sending web pages back in return for URLs received.[6] The web server for a Jade application must run under Windows NT or Windows 2000.

Since the material which the web server is accessing is (in the *MicroUWeb* case under discussion) sourced from a Jade application we will speak of that Jade application as the *Jade server*. The Jade server delivers Jade output to the web server which in turn delivers it over the web to the ultimate client.

The Jade server and the web server may be separate computers or the same computer. For a proof of concept there is no problem having both on the same hardware. But as demand grows it is a good idea to allocate a separate computer to the web server. This makes it easier to add resources in order to scale up the responsiveness of the web traffic part of the system.

In Figure 1 below we show the web traffic at the left managed by the *web server* and the applications at the right (including any being accessed by remote web clients) managed by the *Jade server* .

- The Jade server is controlled by the standard **Jade.exe** program.
- The web server handles the communication between Jade and the web via an ISAPI program (**MS Internet Information Server (IIS)**.[7] TCPIP is used to send the messages between the programs. We will assume that the Web Server and the Jade Server are on the same machine with IP address 'localhost' and that port number 5555 will be used. When a URL is received the web server program calls **Jadehttp.dll** directly. Jadehttp.dll is an executable program in the

---

2 (A modal form is one that seizes the focus—until it is closed not other form can be clicked on. Dialog boxes are the most familiar examples.)

3 URL—Universal Resource Locator: the familiar addresses like http://www.bigBusiness.com\pageLateInfo

4 Hence one of the reasons for the invention of 'cookies'.

5 The actual HTML code for presenting a form to a web client is not generated until a client browser requests that form from an application that has been web-enabled.

6 For demonstration purposes it is possible for the (Jade server+web server) to be on the same computer as the client web browser. The URL will involve the address /localhost/.

7 Supplied with Windows NT and later.

Jade\bin directory. Part of the task of setting up the system is to make it easy for a web user to tell the web server where to find Jade\bin.

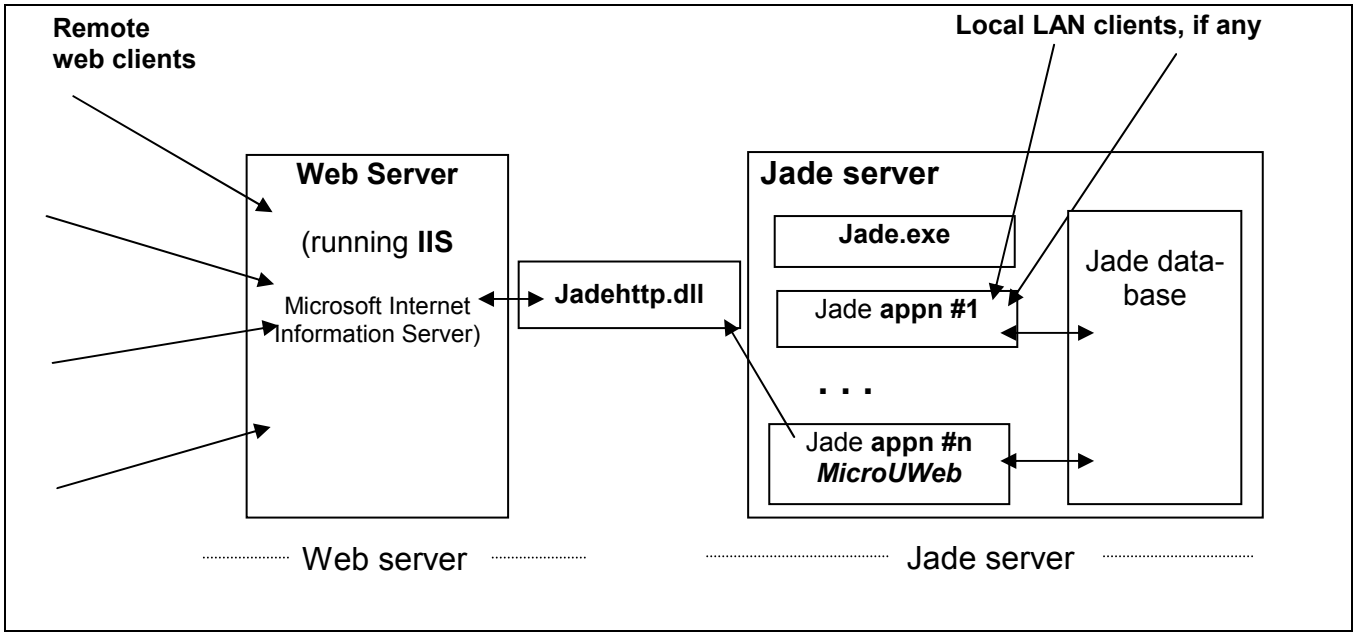


Figure 1 Web server and Jade server

Setting up a Web Server/Jade server link

1. **Web-enable the Jade application . . .**
    a) Enter Jade and **Add** a new Application to the schema, such as the *MicroUWeb* discused above. Select the web-enabled **Application type**.
    b) On the Web Options sheet, enter **localhost:5555** as the Connection Name. (**localhost** is the IP address of the web server and **5555** is the port number we will use for communication between the Web Server and the Jade Server.)
2. **Set up the Web server and create an alias and an .ini file for jadehttp.dll . . .**
    a) Install Internet Information Services
        i. **Control panel | Add/Remove Windows Componenets**.
        ii. Add **Internet Information Services**. Point to the location of the program or supply on disk. [Ours was last found at C:\i386\]. Click **OK**.
        iii. **Services** setup. Check at least the top two services. Answer **Yes** to *Create directory?* The publishing directory can be c:\inetPub\wwwroot. (This directory is where www pages will live.) Click on **OK**.
    b) Create an alias for Jade\bin
        i. From **Start**, run **Control Panel | Administrative Tools | Internet Information Services**. You will see the local network's name for the server machine, e.g. bevan2pc. Double-click on it.
        ii. Click the **Directories** tab. We will **Add** a new directory reference—we have to tell Peer Web Services about the Jade\bin directory.
        iii. **Enter the full path to Jade\bin**. This may be a long path—too long, too private and too prone to change for any remote user to enter into a URL. So we may give it an alias, such as **/microU**.

iv. **Check the Execute box**. We check this because this is not just a passive web page; we are pointing to a directory with *executable* programs.

v. **OK**. Right-click on bevan2PC and click **Stop**. Again and click **Start**.

c) Create an ini file

i. Create a directory called **bin_jadehttp** under **C:\Jade**

ii. Under **bin_jadehttp** create a directory called **ini**.

iii. In the **ini** directory create a file in **Notepad** called **jadehttp.ini**.

iv. Add the **following** new section to the end of the **jadehttp.ini** file.

```
[MicroUWeb]
ApplicationType=WebEnabledForms
TcpConnection=localhost
TcpPort=5555
ConnectionGroup=All
```

3. **This completes the one-time set-up work.** Your system now has a web server service and a web-enabled Jade application.

Some time later . . .

4. Start the Jade server . . .

For a demonstration you may run both the Jade server and the Web server on the same machine.

Start Jade and run the Jade application *MicroUWeb*. You will see a **Jade Web Application Monitor** program launched. This program's window will assist you to monitor web traffic requests.

5. Start a Browser . . .

Start your browser. Normally you might enter into it a URL like www.JadeWorld.com or www.pres.whitehouse.com. But in this illustration both servers are on the same computer. So instead, at this point enter the URL . . . http://localhost/microU/jadehttp.dll?MicroWeb

In this URL . . .

http:// designates the internet protocol to be used

localhost/ signifies that the web server is the same machine that the browser is running on. Otherwise it would designate another machine on the LAN or, more normally, a remote internet address.

microU/ is the alias we created earlier—in this demonstration it was a shortcut for the location of \bin, which was actually c:\Jades\Jade5014/bin when this document was written.

jadehttp.dll is the program we wish to run. This is the Jade-supplied manager of Jade's end of the internet access to an application.

?MicroWeb is the parameter we pass to jadehttp.dll. It names the application that this particular caller wants to run remotely.

You should now see the web version of your *MicroUniversity* application running in the web browser.

Consider: in order to develop a web-commerce application with other web languages you may need to join together several tools (such as cgi, Perl, C++, Java and a database package).

With Jade you have database, business logic, screen design and web interface in one package.

Acknowledgement:

In obtaining this information we were greatly assisted by Arjan van Hasselt and Geoff Davies of Jade from whose excellent **Advanced Jade Course** much invaluable material has been learned.

Bevan J Clarke