

Applino

<http://www.applino.com/>

Edit

Table of Contents

- Applino
 - Introduction
 - Terminology
 - Installation
 - Running Applini
 - The Applino Container
 - Developing an Applino
 - Building Applino

Introduction

- An Applino is to the desktop what a servlet is to the server.
- Applino provides an environment for running multiple Java desktop applications within a single JVM.
- The Applino environment runs as a taskbar icon and starts when the user logs into their computer.
- Applini (Applino applications) are similar to servlets in that they contain runtime descriptors and have their life cycles dictated by the Applino container.

Most Java developers realise that it is inefficient to instantiate a separate JVM for each Java application running on a user's computer. This involves a significant extra start-up time and is wasteful of memory. Unfortunately, due to the need to segregate the effects of one application from another there hasn't been any choice.

On the server side however, much progress has been made in the area of Java Application Servers which are able to run multiple web applications within the same JVM. The level of intra application protection is so good that several hosting companies provide shared hosting where users host their web applications in the same JVM as other users.

The idea behind this project is simple: take the compartmentalisation technology which has been developed for servlets and reapply it to the desktop space. This has proven to be exceedingly simple as concepts such as context life cycles and hierarchical class loaders map perfectly to the requirements of a mono JVM desktop environment.

Typically *porting* an existing application to the Applino environment is trivial. It is simply a matter of ensuring that it is properly started when instructed and properly terminated. The initial window for the hosted application will also be provided by the Applino container (so that it can control events such as window closing).

As a demonstration of how easy it is to port existing applications, we have already provided Applino versions of all the JFC example applications from the JDK.

Edit

Terminology

Applino Container / Runner	The main "container" application running in the taskbar.
Applino <small>(singular)</small> Applini <small>(plural)</small>	The hosted applications. This is similar to an applet or servlet.

Edit

Installation

Unzip the Applino distribution and run `install.bat`. This will copy the required files to `%ProgramFiles%\Applino` and create a link in the "Start Menu\Programs\Startup" causing the Applino container to be run at startup. The container is visible in the task tray (as a green dot).

Edit

Running Applini

Simply drag the Applino over the `Applini` folder on your desktop (i.e. to move the Applino to the correct location). It should now appear in the list of available Applini in the task tray icon.

Edit

The Applino Container

As mentioned previously, Applini are in many ways similar to servlets. For this reason it was decided to base the Applino container on an existing servlet container. The `Jetty` container was chosen because of the fact that it is both lightweight and programmatically configurable (i.e. embeddable).

The `ApplinoContainer` class is simply a container class which configures Jetty and starts the container running in the task tray.

The `ApplinoDeployer` class is the most involved modification to the servlet container in that it scans the `applini` sub-directory for Applini (files with a `.aar` suffix: i.e. Application Archive) and creates a new `ApplinoContext` for each Applino which is found. Expanded `.aar` archives are also supported. When a new `ApplinoContext` is instantiated or destroyed (by deleting the Applino file) the corresponding entry is removed from the task tray icon.

When a user selects the name of an Applino from the task tray icon, the corresponding Applino is started by calling `startApplino()`.

An `ApplinoContext` is simply an extension of a Jetty `WebApplicationContext` which also deals with updating the task tray icon when Applini are started and stopped. In addition it frees any resources associated with an Applino when it is stopped and creates an `ApplinoFrame` for the Applino to use for displaying any graphical output.

[Edit](#)

Developing an Applino

Most Java applications or applets can be easily converted to Applini. The main things to note are:

- Adapt program lifecycle to support the **`contextInitialized`** and **`contextDestroyed`** methods. Typically, code from the main method of an application or the `init` method of an applet will go in the `contextInitialized` method. `contextDestroyed` should be used to free up any resources such as files. Other objects attached to the main Applino class will be garbage collected when the Applino is destroyed.
- Make use of the supplied **`ApplinoFrame`**. The main presentation window for your application can be retrieved using the following code:

```
ApplinoFrame frame = (ApplinoFrame)sc.getAttribute("frame");
```

An `ApplinoFrame` is effectively just a `JFrame` (with an overridden window close method). As most Applications and all Applets use `JFrames` (or a subclass) for display purposes, adapting the program to use the supplied `JFrame` should be trivial.

In any case, we have provided a vast array of examples to get you started and show you clearly how easy it is to transform an existing application. Look in particular at the `/examples` and `/3rdparty` sub-directories.

An example Applino is presented below:

```
package com.applino.example;

import com.applino.ApplinoFrame;
import javax.servlet.*;

public class SimpleApplino implements ServletContextListener
{
    public void contextInitialized(ServletContextEvent event)
    {
        ServletContext sc = event.getServletContext();

        try
        {
            ApplinoFrame frame = (ApplinoFrame)sc.getAttribute("frame");
            frame.setSize(300, 300);
            frame.center();
            frame.setVisible(true);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

```
public void contextDestroyed(ServletContextEvent event)
{
    ServletContext sc = event.getServletContext();
    System.out.println("Context destroyed...");
}
```

[Edit](#)

Building Applino

Three separate build files are provided with the Applino distribution (in the /ant sub-directory):

- build.xml is the main Ant file for building the actual Applino container.
- examples.xml is used for building the bundled (core) example Applini.
- 3rdparty.xml is used for building the 3rd party (Sun) Applini (converted from the JFC example applications bundled with the JDK).

[Edit](#)

applinodocumentation.txt · Last modified: 2008/03/26 14:54 by 15.219.233.71

[Edit this page](#)[Old revisions](#)[Login](#)[Index](#)[Back to top](#)[RSS](#)[XML FEED](#)[CC](#)[LICENSED](#)[\\$1](#)[DONATE](#)[PHP](#)[POWERED](#)[W3C](#)[XHTML 1.0](#)[W3C](#)[CSS](#)[DOKUWIKI](#)