

# Desktop Integration

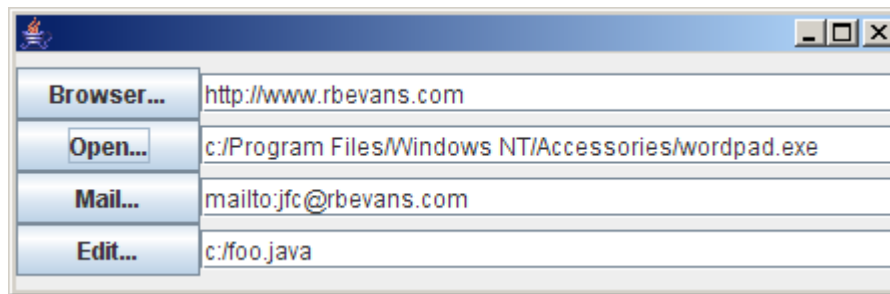
- Starting in JDK 1.5 and more so in JDK 1.6
  - Support for desktop integration (if host OS supports)
- Supported operations include:
  - launching the user-default browser to show a specified URI;
  - launching the user-default mail client with an optional mailto URI;
  - launching a registered application to open, edit or print a specified file.
- Passed object to OS for the Desktop to handle

# Desktop Methods

- `void browse(Uri uri)`
  - Launches the user default browser to display a URI if this is an URI the browser can handle.
- `void edit(File file)`
  - Launches the associated editor application and opens a file for editing.
- `static Desktop getDesktop()`
  - Returns the Desktop instance of the current browser context.
- `static boolean isDesktopSupported()`
  - Tests whether this class is supported on the current platform.
- `boolean isSupported(Desktop.Action action)`
  - Tests whether an action is supported on the current platform.

# Desktop API continued

- `void mail()`
  - Launches the mail composing window of the user default mail client.
- `void mail(Uri mailtoUri)`
  - Launches the mail composing window of the user default mail client, filling the message fields specified by a mailto: URI.
- `void open(File file)`
  - Launches the associated application to open the file.
- `void print(File file)`
  - Prints a file with the native desktop printing facility, using the associated application's print command



# Network Interface Info

New support in JDK 1.6 for getting

- Broadcast address
- Subnet mask
- Mac/hardware address
- MTU size
- State (up/down)
- Know if network interface supports multicasting
- Know if network interface is the loopback interface
- Know if address is dynamic
- Enumerate logical (virtual) interfaces instead of physical interfaces

# System Tray Support

- The SystemTray class represents the system tray for a desktop. On Microsoft Windows it is referred to as the "Taskbar Status Area", on Gnome it is referred to as the "Notification Area", on KDE it is referred to as the "System Tray". The system tray is shared by all applications running on the desktop.
- On some platforms the system tray may not be present or may not be supported, in this case `getSystemTray()` throws `UnsupportedOperationException`. To detect whether the system tray is supported, use `isSupported()`.

## System Tray Support Cont'd

- The SystemTray may contain one or more TrayIcons, which are added to the tray using the add(TrayIcon) method, and removed when no longer needed, using the remove(TrayIcon). TrayIcon consists of an image, a popup menu and a set of associated listeners. Please see the TrayIcon class for details.
- Every Java application has a single SystemTray instance that allows the app to interface with the system tray of the desktop while the app is running. The SystemTray instance can be obtained from the getSystemTray() method. An application may not create its own instance of SystemTray.

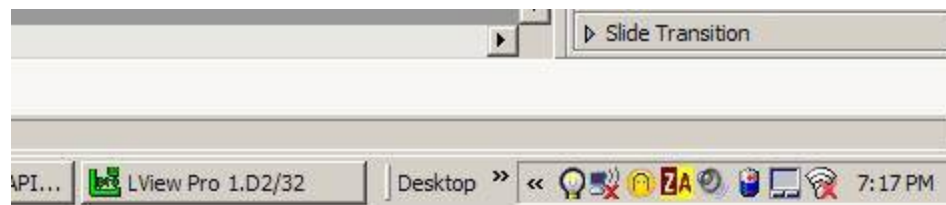
# System Tray Methods

- Here's the methods:

```
void add(TrayIcon trayIcon)
static SystemTray getSystemTray()
TrayIcon[] getTrayIcons()
Dimension getTrayIconSize()
static boolean isSupported()
void remove(TrayIcon trayIcon)
```

- TrayExample.java

💡 ← Used this gif



To make this in  
System tray

Right Mouse Click brings  
up Popup Menu

