Opening Issue and Bug Links in JIRA Client and Deskzilla

The Goal

Make JIRA Client and Deskzilla catch links to issues/bugs from e-mails, web pages, etc., and open them instead of the browser.
I'll refer to JIRA Client for readability, but everything is the same for Deskzilla.

URL Examples:
- **http[s]://jira.mycompany.com/browse/FOO-123** (JIRA Issue)
- **http[s]://bugzilla.mycompany.com/show_bug.cgi?id=123** (Bugzilla bug)

These are normal HTTP[S] URLs, that are generally handled by the default web browser.

URL Scheme Handlers

Operating systems generally decide which application to use for an URL from the URL scheme alone.
Instructions on registering URL scheme handlers: Mac OS X, Windows, Gnome, KDE.

Mac OS X invokes the handler by sending it the "open location" Apple Event.
Windows executes a shell command with the URL as parameter, or sends a message through DDE.
Gnome and KDE execute shell commands.

~~JIRA Client as the Default HTTP[S] Handler~~

The idea is to make JC catch all http:// and https:// links and forward the non-issue ones to the browser. This is achievable one way or another, but very harsh: JC is launched for any URL whatsoever, it must know or figure out the user's preferred browser. Not humble.

Further investigation:
- Is it possible in any OS to associate URL handlers with more elaborate patterns, not just URL schemes?
- Is it possible to make a chain of responsibility, when several handlers are consulted for an URL, and whoever likes it best, wins?

Faux Protocols

The currently investigated solution is to make our own URL schemes and register JC/DZ to handle them.
Then we'd need to rewrite issue http:// URLs into faux:// URLs, and the OS would pick JC as the handler.
JC would then do the inverse rewrite and show the issue.

The schemes:
- **almworks-jc-http, almworks-jc-https** (JIRA Client HTTP/HTTPS)
- **almworks-dz-http, almworks-dz-https** (Deskzilla HTTP/HTTPS)

The URL http://jira.mycompany.com/browse/FOO-123 becomes almworks-jc-http://jira.mycompany.com/browse/FOO-123, and so on.

URL Rewriting Extensions

The rewriting is done in the user's web browser by a special extension that listens for "open URL" events, checks the URL, rewrites it if necessary, and navigates the browser to the faux URL.
This means we need one extension for every (popular) browser out there. But the same extensions would work across all operating systems (where their respective browsers work).

Available extensions:
- Google Chrome -- ALM Works Connector for Chrome (0.2.1)
- Mozilla Firefox -- ALM Works Connector for Firefox (0.1, functionality of Chrome's 0.2.1)
- Apple Safari -- ALM Works Connector for Safari (0.1 prototype)

Needed extensions:
- Internet Explorer (terra incognita)

Details on Connector for Chrome

Open-source the code and make a link here?
NB: there's some secret stuff involved -- the private key for signing the bundle. The same will be true for Safari (Apple Safari Developer certificate).

Opening Issue URLs on Mac OS X

The preferred method is sending the "open location" Apple Event. That's what Mac OS X uses itself.
JC 2.3.4 hotfix 234, DZ 3.0 and JC 3.0 (in development) register a listener for this event via Apple's extensions to the JRE.
This only works if you have *Java for Mac OS X 10.6 Update 3, Java for Mac OS X 10.5 Update 8,* or later updates.

You can do this in your AppleScript:

```
tell application "JIRA Client"
    activate
    open location "http://jira.mycompany.com/browse/FOO-123"
end tell
```

It seems like Mac OS X sends the "open location" to the URL handler, but not the "activate". This means that when JC and Chrome are in different Spaces, JC would open the issue, but the space wouldn't switch.
~~The lab-tested workaround is to use a separate tiny AppleScript application as the faux URL handler, that would also send the "activate" to JC.~~
Fixed in JC 2.3.4 build 5825.236. It sends "activate" to itself through the AppleScript engine for javax.scripting API (requires Snow Leopard with Java Update 3),

You can also use the command line:

```
open "/Applications/JIRA Client.app" --args --open=http://jira.mycompany.com/browse/FOO-123
```

But this will not work if JC is already running. If you really need command line access on OS X, invoke AppleScript via **osascript -e**.

Opening Issue URLs on Windows

JC 2.3.4 hotfix 234, DZ 3.0, JC 3.0 support the **--open=<URL>** command-line parameter (with "real" and "faux" URLs):

```
jiraclient.exe --open=http://jira.mycompany.com/browse/FOO-123
```

If JC is already running, another instance starts, passes the URL via XML-RPC to the running instance, and quits.
This scheme leads to a problem: Windows won't let the "old" JC come to foreground, all it can do is blink in the task bar. Annoying.

Further investigation:
- Can registering JC as a DDE server solve the above problem?
- If yes, look at Java/DDE integration options -- one, two, three, more?
- If no, then what else can be done?

Opening Issue URL on Linux

JC 2.3.4 hotfix 234, DZ 3.0, JC 3.0 support the **--open=<URL>** command-line parameter (with "real" and "faux" URLs):

```
./jiraclient.sh --open=http://jira.mycompany.com/browse/FOO-123
```

Further investigation:
- Assemble a working solution for Gnome and KDE -- write scripts, or, at least, instructions.