



IcedTea-Web goes offline and beyond[1]

Fosdem Feb. 2015

Jiří Vaněk

[1] much much much beyond!

Index

- 1) What is ITW
 - 2) Missing features and previous updates
 - 3) Future of applets
 - 4) Security vulnerabilities and new security settings and manifest attributes
 - 5) Time for fun and community help highlights
 - a) android sandbox - just repeated error
 - b) run in sandbox
 - I) demo 1
 - c) Still the future? “-html” switch
 - I) demo 2
 - d) Offline capabilities
 - I) demo 3
 - e) Mixture of above - shortcuts
 - I) demo 4
 - f) Community project based on ITW
 - I) Jogamp
 - II) New docs
 - III) windows java plugin
 - IV) localizations
 - 6) JDK 9
-

What is IcedTea-Web

- Evolved from nineties NETX and gnu plugin
- ITW firstly presented on Fosdem in 2012
- *opensource* and *independent* **java browser plugin** and **javaws client**
 - attempting to run on most jdks/vms
 - 1.5 supports jdk 6, 7, 8 (, 9)
 - head ≥ 7
- <http://icedtea.classpath.org/wiki/IcedTea-Web>
- <http://icedtea.classpath.org/hg/icedtea-web>

Missing features and previous updates

- previous releases were moreover focused to
 - internal refactorings
 - the evolution was complicated
 - and people working on time quite unstable
 - internal fixes
 - “working with Oracle one not with ITW”
 - ...empty codebase?
 - internal bug fixes
 - be feature compliant with specification and/or Oracle implementation
- We are not and we probably never will be
 - no one seems to be missing missing features
 - which can mean more then just they are not needed
 - the pilot implementation is also not specification compliant
- Finally the count of “incompatible” bugs is dropping

Security vulnerabilities and applet security level/new manifest attributes

- java sandbox (security manager), although hacked in past, is good
- much bigger security threat is user clicking “yes” until run
- Oracle security settings:
 - *medium* - all applets will be run after security prompt
 - *high* (minimum recommended) - only applets signed by trusted authority will be run
 - *very high* - only applets signed with a valid certificate are allowed to run

 - It is too easy to do self-signed app!
 - in addition, ITW have
 - **low security** - even unsigned applets are run without asking
 - *disable* - no java plugin will run
- disabling of plugins should be (is) browser's work - but not all have
- nor *medium* nor *high* do not prevent against “yes yes run” issue

Security vulnerabilities and applet security level/new manifest attributes

- Permissions
- Codebase
- Application-Name
- Trusted-Only
- Entry-Point - still missing in ITW
- Caller-Allowable - still missing in ITW
- Trusted-Library - still missing in ITW

- moreover can be waived by user
- over clear security tightens - still no prevention against “yes yes issue”

Future of applets

- Is dark...
 - Oracle is not trusting their own sandbox
 - Google is cutting down NPAPI
 - hopefully, nobody is developing any new ones (or somehow relying on them)
- But still, there is plenty good old applets!
 - allow them run without new attributes
 - allow them run outside browser
 - allow them to be run offline
 - and forbid (signed ones) them to do everything

Allow them to run at all

- So after years of torturing, there is finally time for some fun

Android sandbox - just repeated mistake?

- When you install application from Google play, it simply tells: this app requires net, storage, settings, root (what? Not even user have root!) ... access
 - deal with it, or don't install it
 - Java since beginning had the similar....
 - unsigned - can not access sensitive resources (but may escape the sandbox)
 - signed - can do all (deal with it or don't run it)
- ... but is capable of opposite
- Last steps in pilot implementation (new attributes and security levels) suggests that
 - unsigned applets - which can not harm you are no longer ... recommended.
 - signed, which can - absolute victory of “yes yes run issue”

Run in sandbox - inverted approach

- inverted approach - allow signed applications to do only what necessary
 - why should my tic-tac-toe game be signed?
 - if so, why it should access network and storage?
 - ok, maybe storage is ok
 - well written applications will work, with error dialogue and limited functionality
 - wrong ones, or malicious ones, will fail

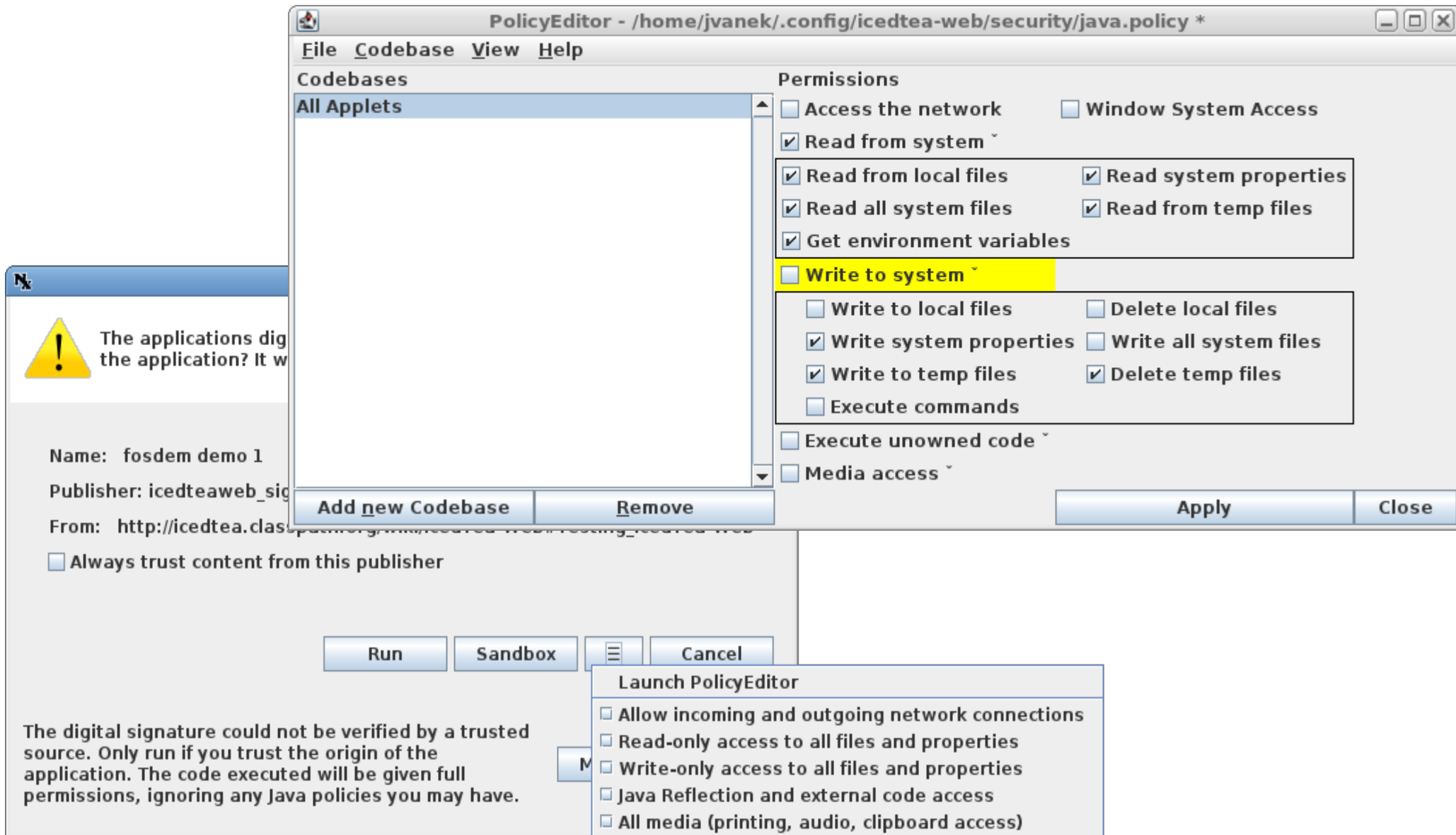
- <http://icedtea.classpath.org/wiki/IcedTea-Web-Custom-Policies>

- pitfalls
 - consider java sandbox as unbreakable
 - at least it is not easy to hack it
 - custom permissions may need a lot of tuning on both ITW side and users' side
 - dying policy tool - policy editor is getting full policy parser

Run in sandbox - inverted approach

- Demo 1!
 - See the **new developer console** in background!
 - Signed x unsigned
 - Javaws x browser
 - Signed x signed in sandbox
 - Signed x signed with custom permissions
 - Good x bad apps
 - Policyeditor x policytool

Run in sandbox - inverted approach



The screenshot shows a Java security warning dialog in the foreground and the PolicyEditor application in the background.

Warning Dialog:

- Icon: Yellow triangle with an exclamation mark.
- Text: "The applications dig the application? It w"
- Name: fosdem demo 1
- Publisher: icedteaweb_sig
- From: http://icedtea.classpath.org/icedtea-web/signing/
- Checkbox: Always trust content from this publisher
- Buttons: Run, Sandbox, Cancel
- Text: "The digital signature could not be verified by a trusted source. Only run if you trust the origin of the application. The code executed will be given full permissions, ignoring any Java policies you may have."

PolicyEditor Application:

- Title: PolicyEditor - /home/jvanek/.config/icedtea-web/security/java.policy *
- Menu: File Codebase View Help
- Codebases: All Applets
- Permissions:

 - Access the network
 - Window System Access
 - Read from system
 - Read from local files
 - Read all system files
 - Get environment variables
 - Read system properties
 - Read from temp files
 - Write to system (highlighted in yellow)
 - Write to local files
 - Write system properties
 - Write to temp files
 - Execute commands
 - Delete local files
 - Write all system files
 - Delete temp files
 - Execute unowned code
 - Media access

- Buttons: Add new Codebase, Remove, Apply, Close

Launch PolicyEditor Dialog:

- Buttons: Run, Sandbox, Cancel
- Text: "Launch PolicyEditor"
- Permissions:

 - Allow incoming and outgoing network connections
 - Read-only access to all files and properties
 - Write-only access to all files and properties
 - Java Reflection and external code access
 - All media (printing, audio, clipboard access)

HTML switch

- The future of applets is still dark
 - Google cutting down NPAPI
 - long way to adapt ITW plugin to PPAPI
 - probably not worthy (if possible at all - volunteers?)
 - appletviewer do not support sandbox (and much more)
- `javaws -html url [applet id]`
 - parse page and lunch applet(s)
 - using regular applet sandbox
 - javascript launchers
 - can be solved by evaluating javascript first
 - `htmlunit` or similar
 - security impacts?
 - `java<->javascript` communication
 - can be faked, but never fully compatible
 - not the target audience

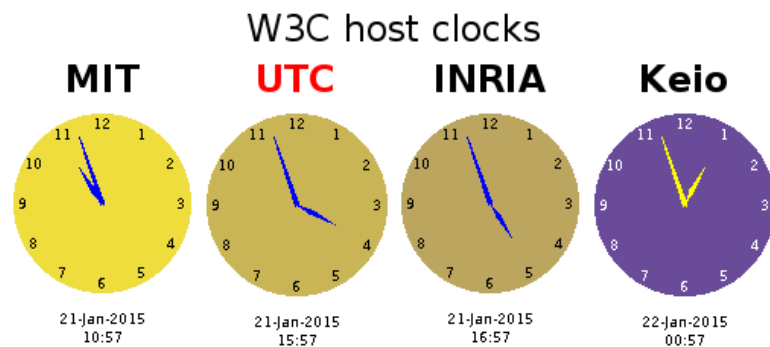
HTML switch

- Demo 2!
 - See the **new developer console** in background O:)
 - Single applet
 - Selected applet
 - All applets
 - Multiple sandboxes
- Online demos:
 - Future of applets:
<http://www.walter-fendt.de/ph14e/cpendula.htm>
 - Selected or all:
<http://www.w3.org/People/mimasa/test/object/java/clock>
 - Complex one:
<http://jogamp.org/deployment/jogamp-next/jogl-applet-runner-newt-ElektronenMultiplizierer-napplet.html>

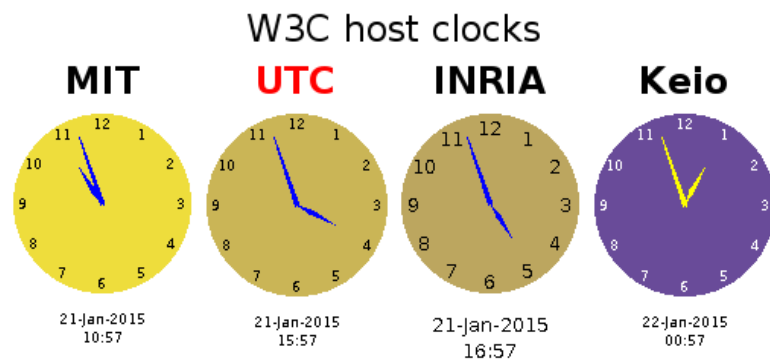
HTML switch

Java applet test with `applet` and `object`

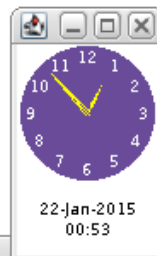
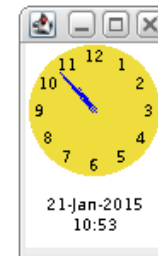
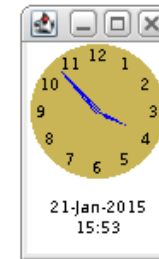
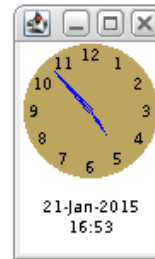
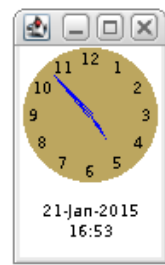
Java applets with `applet`



Java applets with `object`



HTML switch



```
jvanek@jvanek:/run/media/jvanek/6977-9723/fosdem/runInSandbox/signed
```

File Edit View Search Terminal Help

```
[jvanek@jvanek signed]$ ~/icedtea-web-image/bin/javaws -html http://www.w3.org/People/mimasa/test/object/java/clock all
```

Offline capabilities

- `<offline-allowed>` and `<shortcut online="">` tags are often misused or simply forgotten
- What about applets?
- What if I don't want to update my jnlp app?
- Offline detection x `-Xoffline` switch

- ITW tries to run **any** application or applet even if environment is offline and/or machine is offline and/or `offline-allowed` is missing
- ITW may pretend it is offline by adding **-Xoffline** to `javaws` command
 - offline capabilities are fully depending on cache
 - it do not prevent client applications to use network

Offline capabilities

- Demo 3!
- Online demos:
 - You may still be watching **new developer console** in background
 - Complex ones to watch the payload with/without **-Xoffline**:
 - <http://jogamp.org/deployment/jogamp-next/jogl-applet-runner-newt-ElektronenMultiplizierer-napplet.html>
 - <http://www.sweethome3d.com/SweetHome3D.jnlp>
 - No network, see jnlp and applet to work
 - <http://www.sweethome3d.com/SweetHome3D.jnlp>
 - <http://www.walter-fendt.de/ph14e/cpendula.htm>
 - Not surprising heavy cache dependence

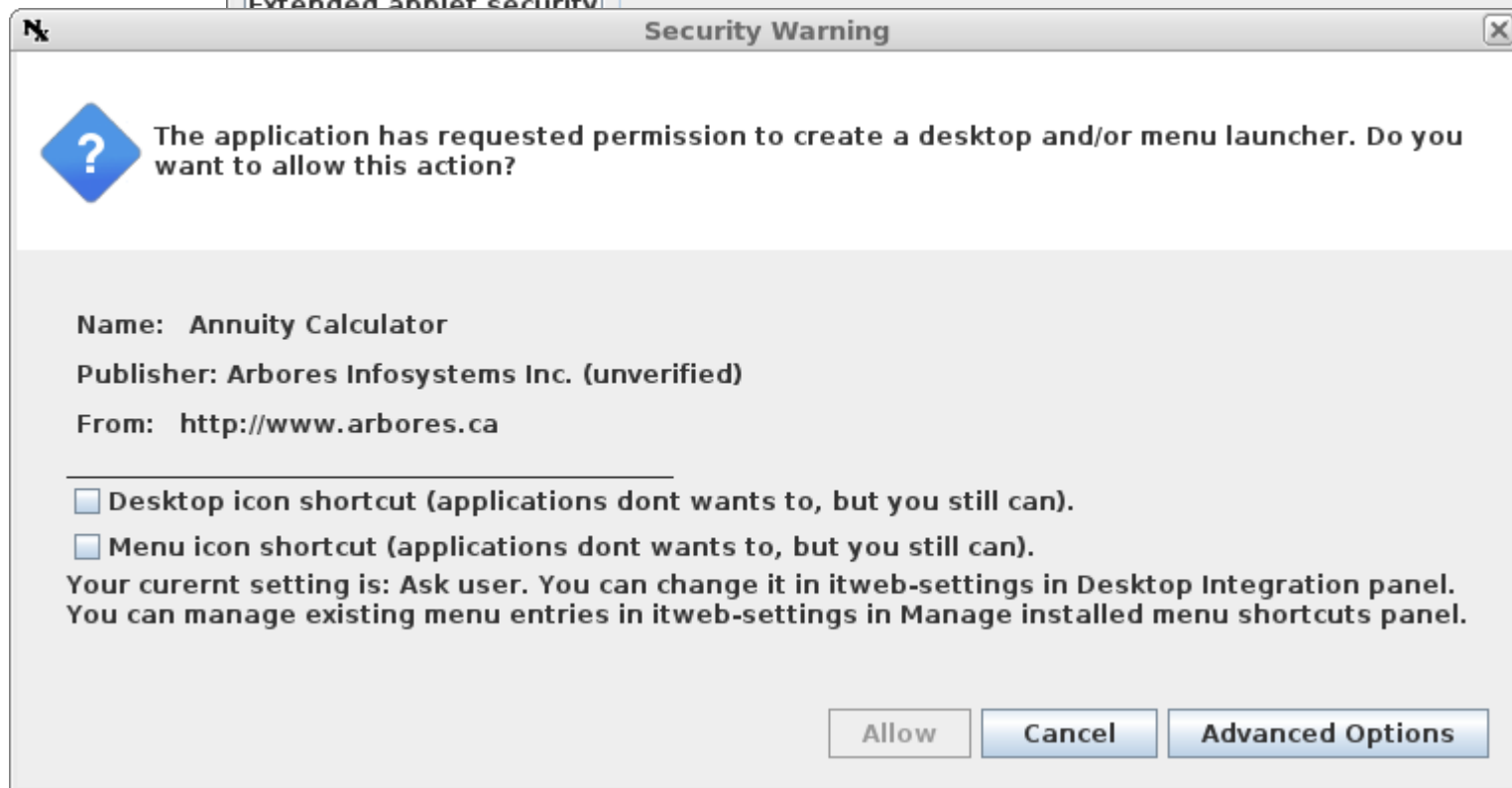
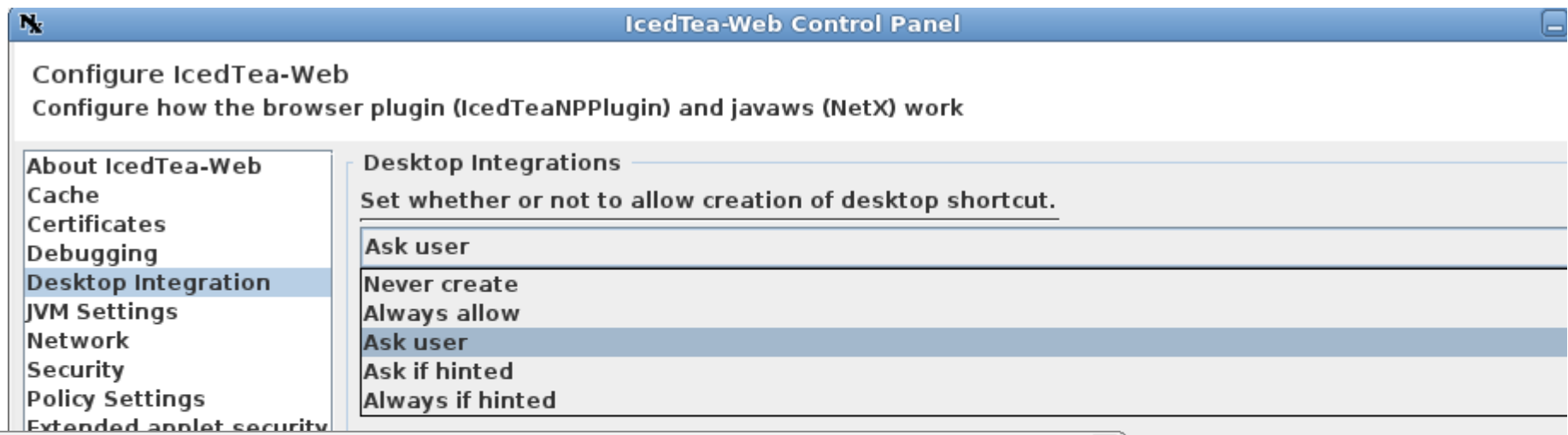
Shortcuts

- Getting mixture of offline, sandbox and -html
 - ITW can serve as secure (as java sandbox allows) desktop deployment environment
- Also applets/applications without <menu> or <desktop> shortcut elements may be “bookmarked”
- Advanced bookmark options should allow
 - “Bookmark” any jnlp app or applet
 - Desktop shortcut
 - Menu shortcut
 - Any applet
 - In browser
 - Via -html
 - Via generated jnlp
- Handling of icons is not depending on cache
- Itweb-settings management tool will be needed

Shortcuts


- Demo 4!
 - Jnlp desktop & menu shortcuts
 - <http://www.arbores.ca/AnnuityCalc.jnlp>
 - <http://www.sweethome3d.com/SweetHome3D.jnlp>
 - Applet desktop and menu shortcuts
 - <http://jogamp.org/deployment/jogamp-next/jogl-applet-runner-newt-gears-normal-napplet.html>
 - Browser
 - -html
 - Generated jnlp
 - jnlphref

Shortcuts



Shortcuts

Security Warning

 The application has requested permission to create a desktop and/or menu launcher. Do you want to allow this action?


Name: <no associated certificate>
Publisher: <no associated certificate>
From: www.walter-fendt.de

Desktop icon shortcut (applications dont wants to, but you still can).
 browser desktop item jnlp generated jnlp href javaws html fix issues in jnlp

Menu icon shortcut (applications dont wants to, but you still can).
 browser desktop item jnlp generated jnlp href javaws html fix issues in jnlp

Your curenrt setting is: Ask user. You can change it in itweb-settings in Desktop Integration panel.
You can manage existing menu entries in itweb-settings in Manage installed menu shortcuts panel.

Remember by application Remember by domain Dont remember

 The application has requ want to allow this action

Name: Sweet Home 3D
Publisher: eTeks (unverified)
From: http://www.sweethome

Desktop icon shortcut (applications want to).
 Menu icon shortcut (applications will try to include to submenu - eTeks Sweet Home 3D).
Your curenrt setting is: Ask user. You can change it in itweb-settings in Desktop Integration panel.
You can manage existing menu entries in itweb-settings in Manage installed menu shortcuts panel.

Remember by application Remember by domain Dont remember

Community projects

- Translations
 - CZ
 - PL + DE
 - With finally revisited and fixed documentation
 - CZ, PL, and DE man pages, html and run-time docs
 - Rumors about FR one
- ITW plugin for Windows
 - No IE right now
 - Still in beginnings, but on good way
 - 11/9/2014 - <http://mail.openjdk.java.net/pipermail/distro-pkg-dev/2014-September/029509.html>
- Awtless plugin
 - Presented at Fosdem 2014
 - Probably eternal fork :(

JDK 9

- Currently ITW works fine with jdk9
- We are not sure how ITW will survive modules
 - ITW currently depends on few private fields and methods
 - <http://icedtea.classpath.org/hg/icedtea-web/rev/c6591d36d68a#I6.1>
 - patches posted upstream
 - no positive reaction
 - see “make life easier for general java plugin” thread
 - <http://mail.openjdk.java.net/pipermail/awt-dev/2014-May/007828.html>
 - unless public API is designed inside jdk9
 - after several months still no full answer from Oracle (still the same thread, half year later)
 - or jar compatibility kept
 - ITW will be a bit doomed
- How will custom applications work on jdk9?
 - modular application in javaws/plugin
 - modular jdk and legacy jnlp-apps/applets
 - modular javaws/plugin and legacy apps
 - legacy javaws/plugin on modular JDK
- There is a lot of similarity between JNLP specification and jigsaw.
 - solution should be achievable

Good live demo to try it all

- <http://caff.de/applettest/Signed.html>

Questions? Opinions?