EntwicklerCamp 2011

Signierung & Verteilung von Plugins für den Notes Client

René Winkelmeyer

Über mich – René Winkelmeyer

dp consulting | purify it
Senior Consultant
http://www.dpocs.de

Lotus Notes Traveler
- Infrastruktur & Security Workshops
- Traveler Rollouts bei diversen Kunden
- Kundenumgebungen zwischen 200 und 5.000 aktiven Devices (Apple, Nokia, Android, Windows Mobile
- Entwicklung von Tools rund um Traveler (Administration & Apple Device Enrollment)

- RCP und xPages-Development

IBM Notes/Domino NEXT Design Partner

Blog: http://blog.winkelmeyer.com

Agenda

Worum geht es in dieser Session?

Verteilung von Java Extensions für Notes 8.5.x

- Woraus bestehen Extensions (Feature / Plugins / Update Site)?

- Sicheres Deployment und Sicherstellung, das nur gewünschte Extensions installiert werden können

- Vorbereitungen für das Deployment (Signierung)

- Verteilung über Domino Policys & Widgets

Übersicht

Was wird benötigt für das Deployment?

Die Java Extension (a.k.a. Plugins) (File Navigator)

CA-Prozeß und einige Java Tools zur Signierung

Eclipse Updatesite NSF

Widget Catalog NSF

Desktop & Security Policy Einstellungen

Java Extensions – Wat is‘ne Extension?

Was ist eine Extension´?

Eine Extension ist eine Ansammlung von Plugins und Features!

Wichtig ist Plugins und Features zu unterscheiden:

- Plugins bilden die Funktionalität ab & dort spielt die Musik
- Feature referenzieren Plugins und führen diese zusammen.

- Die einfachste Extension benötigt ein Plugin für den Code und ein Feature, um das Plugin zu installieren.

- Nur Features können über im Notes Client installiert werden

Java Extensions - Aufbau

Plugin

- META-INF/MANIFEST.MF
  - Contains plugin version info and dependencies
- plugin.xml
  - Extension points used / exposed
- Java code
  - Well duh!
- Resources
  - Localization, images, properties etc.

Feature

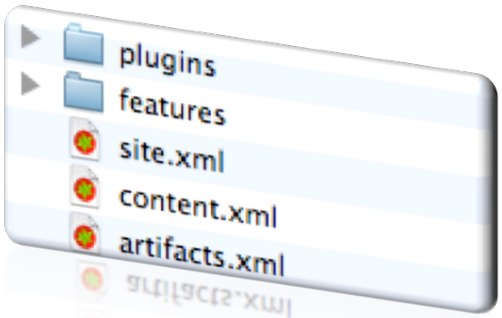
- feature.xml
  - Which plugins make up the feature (incl. which versions)
  - License, copyright etc.
  - Where to go for updates to this feature

Java Extensions - Bereitstellung

Eclipse Update Site

Werden benötigt um neue Feature installieren zu können.
(Wichtig – wir installieren Features und keine Plugins!)

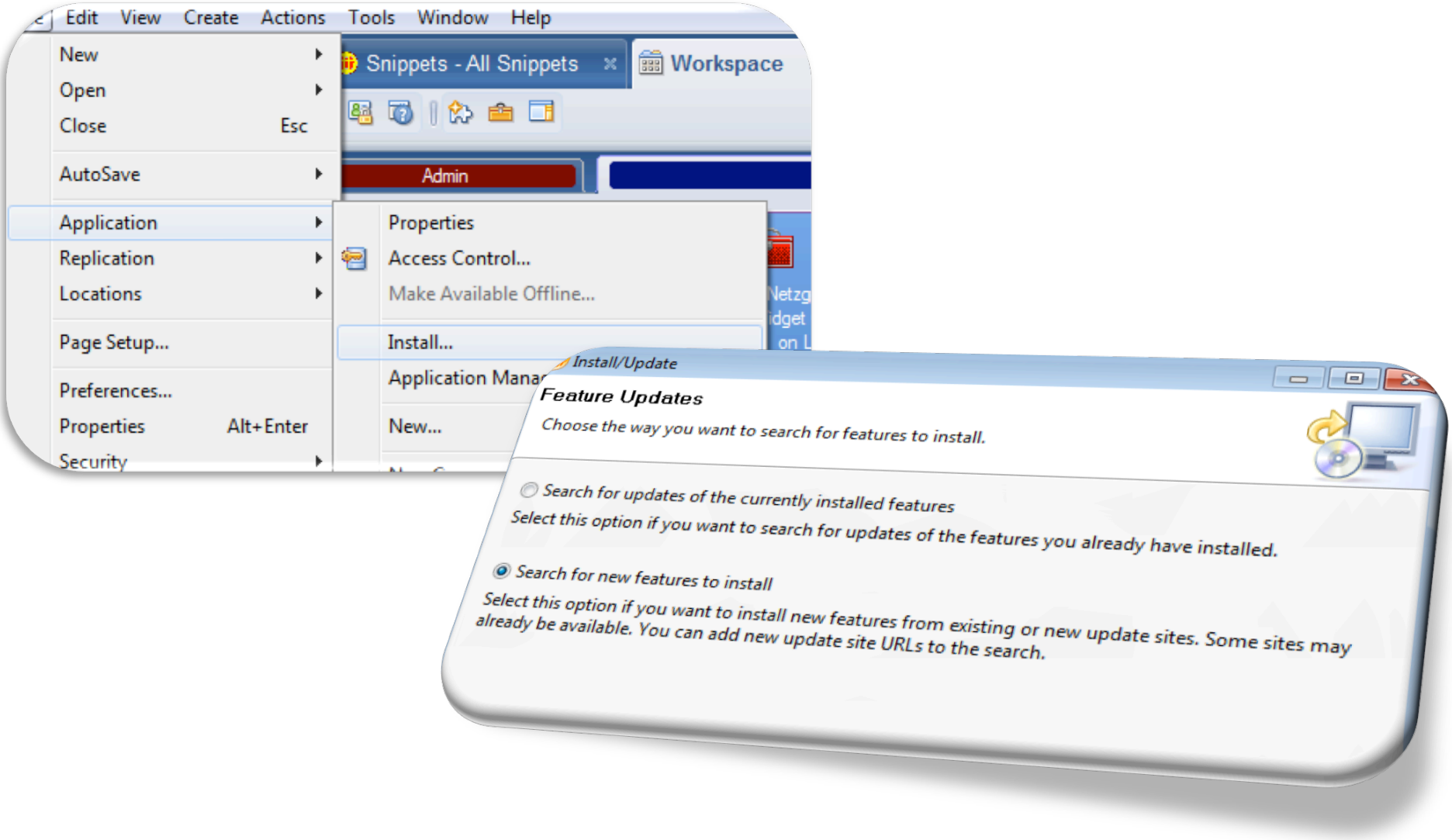
Eine Feature Update Site besteht aus:
- ‚feature'-Verzeichnis
- ‚plugins'-Verzeichnis
- site.xml



Bereitgestellt als ZIP-Datei oder einem Verzeichnis
Lokal oder als Netzressource
(Zugriff per HTTP oder NRPC im Notes Umfeld)

Standard Template ‚Eclipse Update Site' hierfür verwenden

Java Extensions - Bereitstellung



Sicherheit

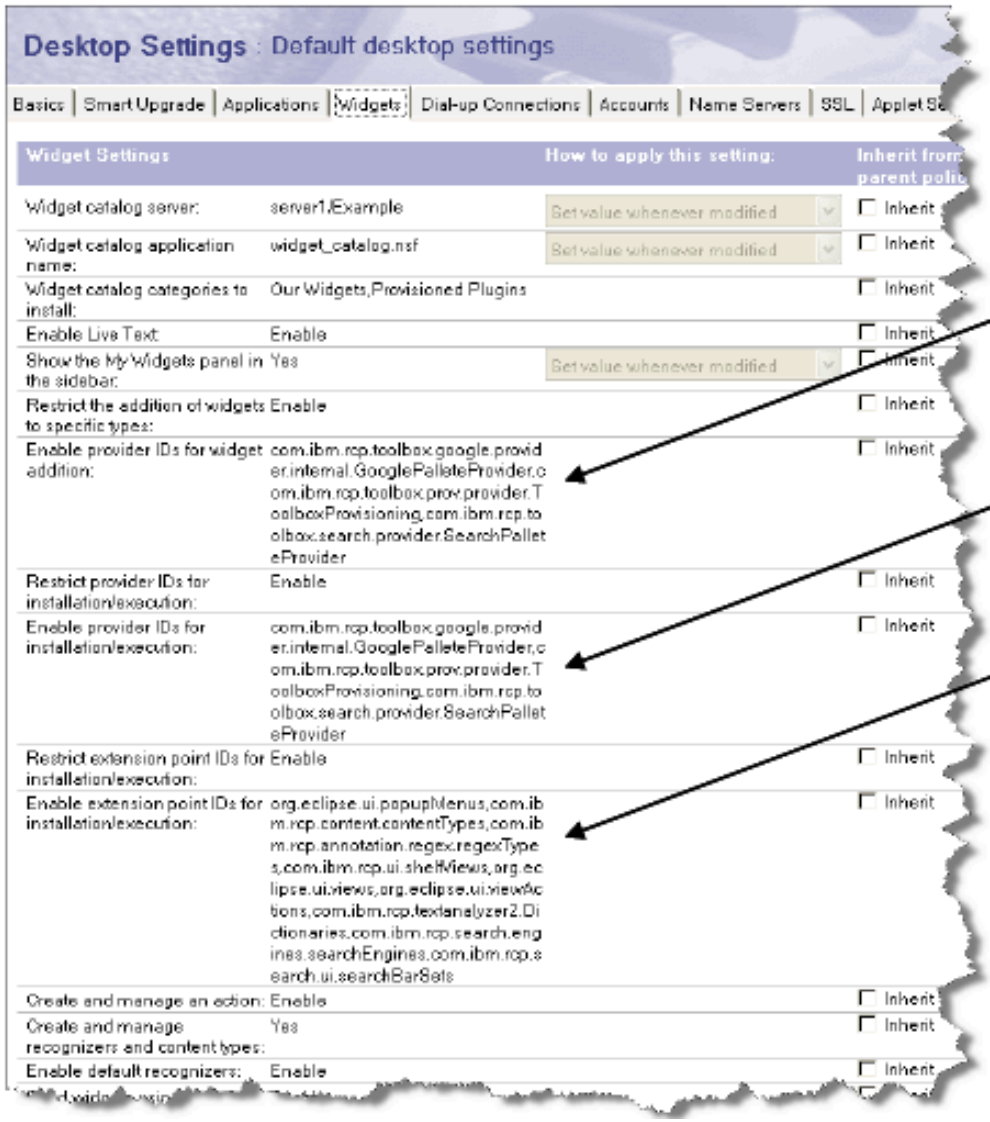
Wie kann ich steuern, welche Extensions ein User installieren und was ein Plugin ausführen darf?

Da ein Plugin in Java programmiert ist, kann es im Prinzip all das tun, was mit Java machbar ist. (Kompletter Zugriff aufs Dateisystem, Notes & Netzwerk,...)

Es gibt keine ECL‘s für Java Extensions wie sie aus Notes bekannt sind.

Ist das Plugin erst einmal installiert, stehen dem Plugin alle Wege offen.

Sicherheit



Which widget types may the user create using UI

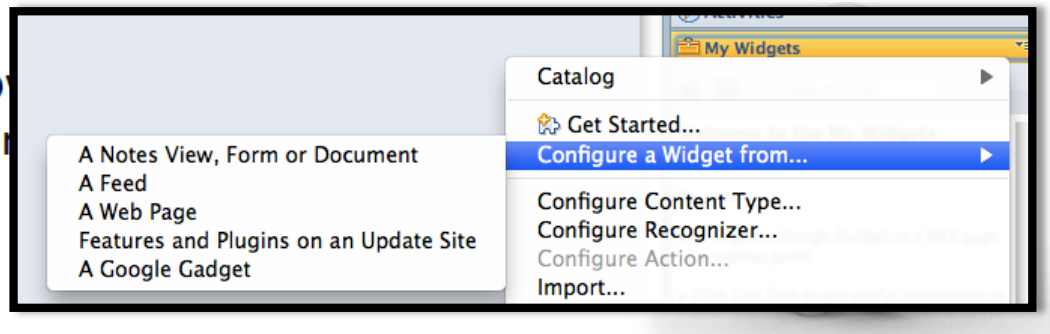
Which widget types may the user install using UI/policies

Which extension points may the user install using UI/policies

consulting purify it

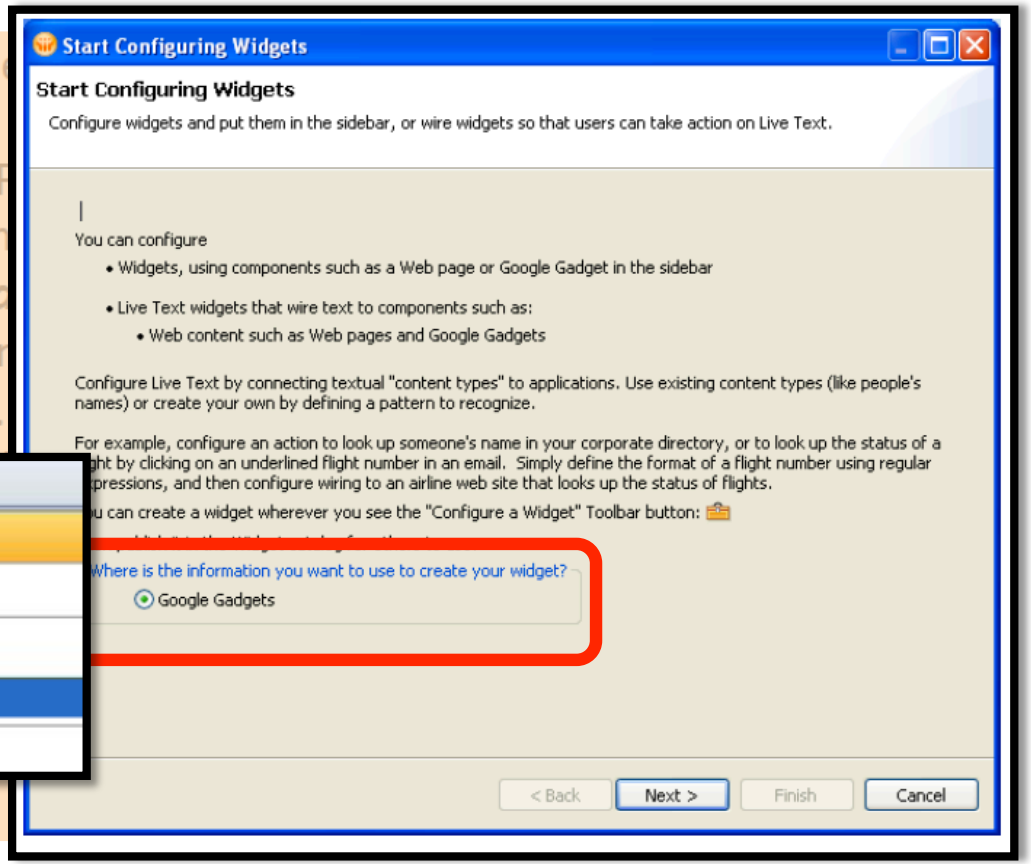
Restrict to widget types

- com.ibm.rcp.toolbox.google.provider.internal.GooglePalleteProvider
  - Allow Google gadgets
- com.ibm.rcp.toolbox.web.provider.WebServicesPalleteProvider
  - Allow widgets using web pages
- com.ibm.rcp.toolbox.feeds.FeedPalleteProvider
  - Allow widgets using feeds (Atom, RSS)
- ***com.ibm.rcp.toolbox.prov.provider.ToolboxProvisioning***
  - Allows provisioning of Java extensions using widget descriptors
- com.ibm.notes.toolbox.provider.NotesViewPalleteProvider
  - Allow Notes views / documents in widgets
- com.ibm.notes.toolbox.provider.NotesFormPalleteProvider
  - Allow Notes forms in widgets
- com.ibm.rcp.toolbox.search.prov
  - Allow widget to go into the sear



Restrict to widget types

- com.ibm.rcp.toolbox.google.provider.internal.GooglePalleteProvider
  - Allow Google gadgets
- com.ibm.rcp.toolbox.web.provid
  - Allow widgets using web pages
- com.ibm.rcp.toolbox.feeds.FeedP
  - Allow widgets using feeds (Atom
- ***com.ibm.rcp.toolbox.prov.provid***
  - Allows provisioning of Java exter
- com.ibm.notes.toolbox.provider.



consulting purify it

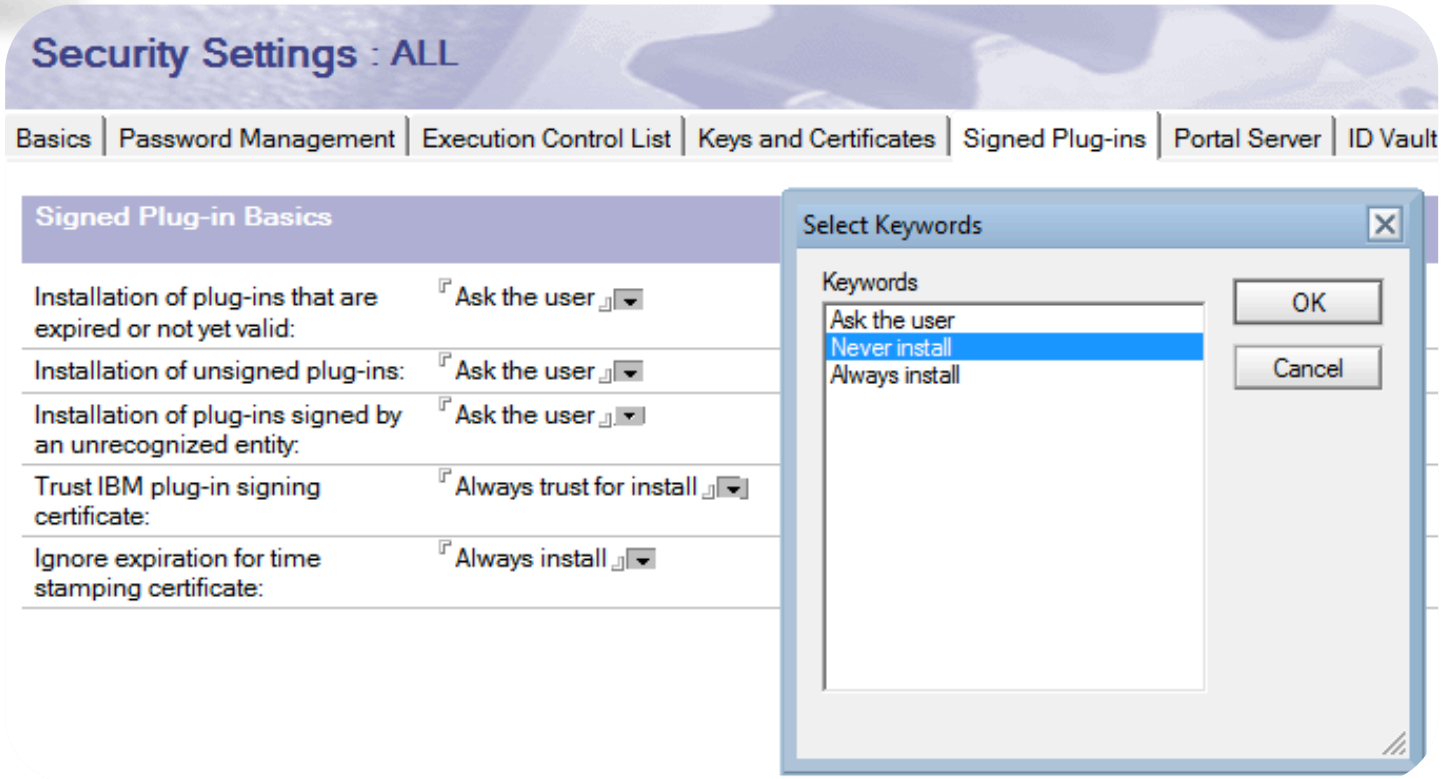
Restrict to extension points

- org.eclipse.ui.popupMenus
- com.ibm.rcp.content.contentTypes
- com.ibm.rcp.annotation.regex.regexTypes
- ***com.ibm.rcp.ui.shelfViews***
- ***org.eclipse.ui.views***
- org.eclipse.ui.viewActions
- com.ibm.rcp.textanalyzer2.Dictionaries
- com.ibm.rcp.search.engines.searchEngines
- com.ibm.rcp.search.ui.searchBarSets

Sicherheit

Wir können nur kontrollieren was installiert wird; aber nicht, was der Code dann auf dem Client macht.

Die Kontrolle erfolgt über digitale Zertifikate und der Verwendung digitaler Signaturen, um die Code Herkunft zu verifizieren.



Sicherheit – Zertifikate & Signaturen

Warum digitale Signaturen oder warum habe ich einen Personalausweis?

Der Zollbeamte vertraut meiner Identität, weil diese so in meinem Pass hinterlegt ist.

Der Beamte vertraut nicht mir, sondern dem Innenministerium, die über die Bürgerbüros die Pässe für Ihre Bürger ausgeben.

Da ich einen deutschen Ausweis habe, können bei Vorlage Beamte darauf vertrauen, das ich deutscher Staatsbürger bin und meine Identität gültig ist.

Sicherheit – Zertifikate & Signaturen

Zertifikate sind die Ausweise für Software.

Digitale Zertifikate werden benötigt um Vertrauen (Trust) zwischen Softwareumgebungen aufzubauen.

Verwendet wird eine Public Key Verschlüsselung

Ein Zertifikat wird von einer vertrauenswürdigen Person/ Instanz ausgestellt.

Ich kann auch ein eigenes Zertifikat erzeugen (a.k.a. selbstsigniertes Zertifikat).
Dies führt dann aber dazu, das ich mir im Prinzip einen eigenen Ausweis ausstelle.

Eventuell werden Andere diesem Zertifikat insbesondere dem Aussteller nicht vertrauen.

Sicherheit – Zertifikate & Signaturen

In Notes haben wir zwei Arten von Zertifikaten

- Notes Zertifikate
- Internet Zertifikate

Wir benutzen Internet Zertifikate für

- Sicheren Webzugriff (HTTPS)
- Sichere Mail (S/MIME)

Ein Internet Zertifikat wird in der ID-Datei des Benutzers gespeichert. Auf dem Server wird das Zertifikat in einer Key-Ring-Datei abgelegt.

Genau so wie Notes Zertifikate benötigt werden, um die Identität des Benutzers sicherzustellen, werden Internet Zertifikate verwendet.

Damit Benutzer einem Internet Zertifikat vertrauen, muss es querzugelassen werden und im privaten Adressbuch verfügbar sein.
Ab 8.5.1 kann dies per Policy automatisiert verteilt werden

Sicherheit – Signieren von Extensions

Java Extensions (Plugins) bestehen aus JAR-Dateien

JAR-Dateien können digital signiert werden.

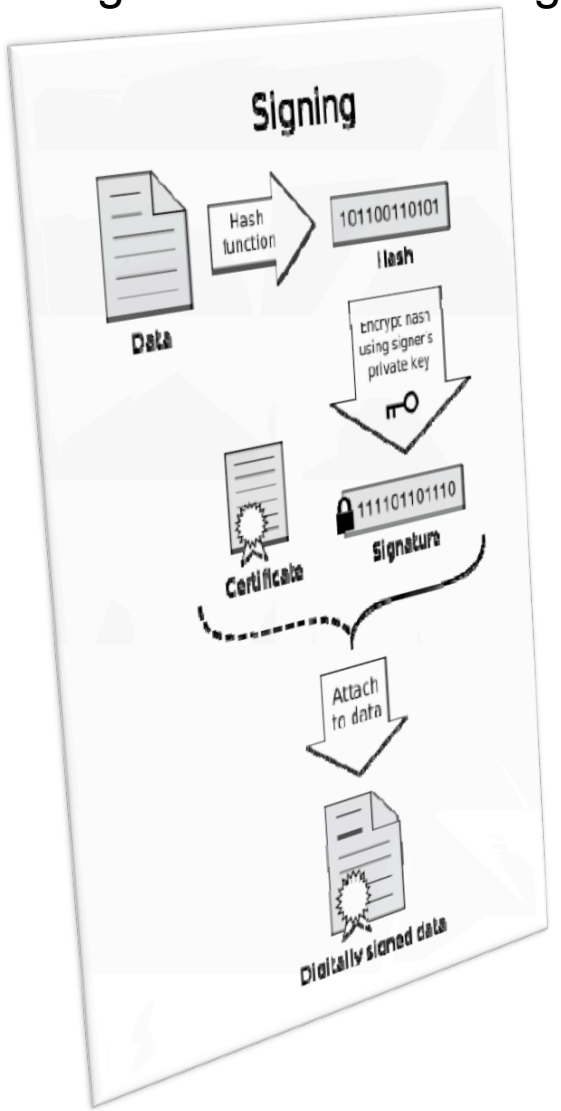
Dies gewährleistet, das

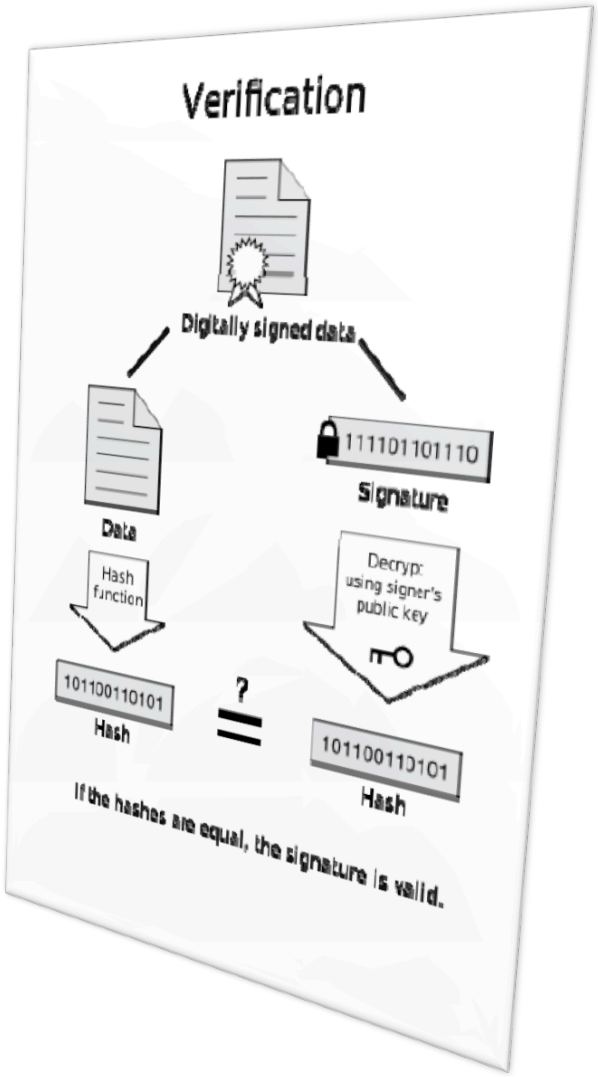
- Die Datei nicht verändert wurde
- Das der Hersteller der JAR-Datei verifiziert werden kann.

Die Signatur wird neben dem eigentlichen Code mit in der JAR-Datei gespeichert.

Sicherheit – Signieren von Extensions

Was geschieht bei der Signierung bzw. bei der Verfikation?





Sicherheit – Signieren von Extensions

Variante I: Schritte um eine JAR-Datei zu signieren:

1. Internet Zertifikat erstellen (Wir gehen von einem Selbstsignierten aus!!!)
    1. CA-Prozeß einrichten
    2. Neuen Internet Zertifizierer (ca/netzgoetter/net) erstellen und im CA-Prozeß bereitstellen

2. Für einen Benutzer (Code Signer/ca/netzgoetter/net) einen Internet Zertifikat ausstellen lassen.

3. Öffentlichen Schlüssel des Internet Zertifikats querzulassen.

4. Internet Zertifikat exportieren und in Java-Code-Store-Format konvertieren.

5. JAR-Dateien mit JAR-Signer-Tool signieren

Detailanleitung: http://bit.ly/9aViOB von Mikkel Heisterberg

Sicherheit – Signieren von Extensions

Variante II: Schritte um eine JAR-Datei zu signieren:

1. Internet Zertifikat erstellen mit JDK Keytool erstellen
2. Öffentlichen Schlüssel des Internet Zertifikats querzulassen und an die Clients verteilen
3. JAR-Dateien mit JAR-Signer-Tool signieren
4. Eclipse Update Site erstellen

1. Internet Zertifikat erstellen

   (a) Installation einer aktuellen Java JDKs und setzen der JAVA_HOME Umgebungsvariablen.

   set JAVA_HOME=C:\java\jdk1.5.0_05

1. Internet Zertifikat erstellen

(b) Erstellen eins Keystores und Erzeugung eines Schlüsselpaars:

```
%JAVA_HOME%\bin\keytool
    -genkey -dname "cn=signer, ou=ca, o=netzgoetter, c=DE“
    -alias “Code Signer"
    -keypass mypassword
    -keystore C:\work\mykeystore
    -storepass geheim
    -keyalg "RSA"
    -validity 360
```

1. Internet Zertifikat erstellen

(c) Zur Kontrolle einmal prüfen, ob das Zertifikat auch im Store ist:

```
%JAVA_HOME%\bin\keytool
    -list -v
    -alias “Code Signer"
    -keystore C:\work\mykeystore
    -storepass geheim
```

2. Öffentlichen Schlüssel des Internet Zertifikats querzulassen und an die Clients verteilen

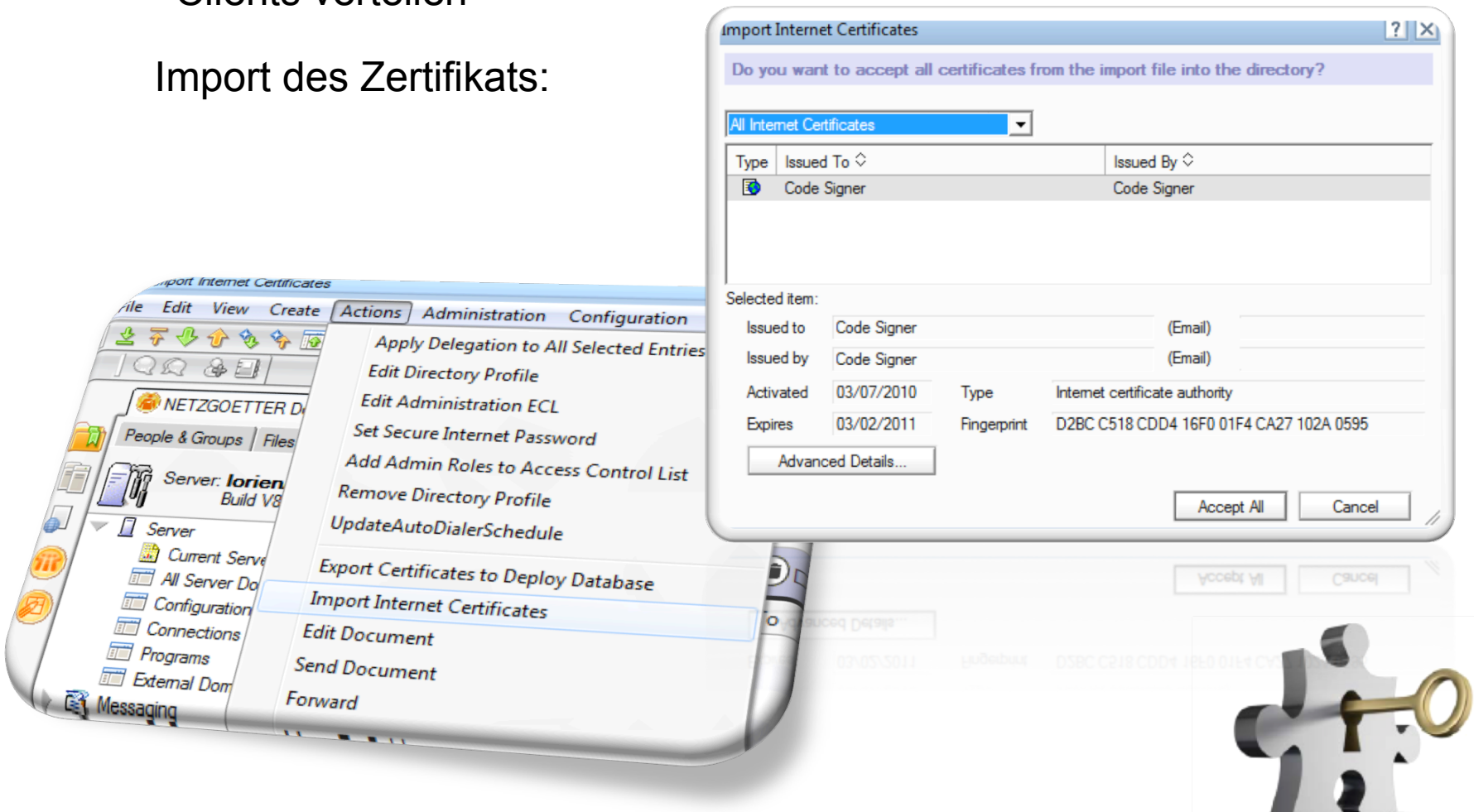
Export des Zertifikats:

```
%JAVA_HOME%\bin\keytool
    -export
    -alias “Code Signer"
    -file  myselfsignedcert.cer
    -keystore C:\work\mykeystore
    -storepass geheim
```

Sicherheit – Signieren von Extensions

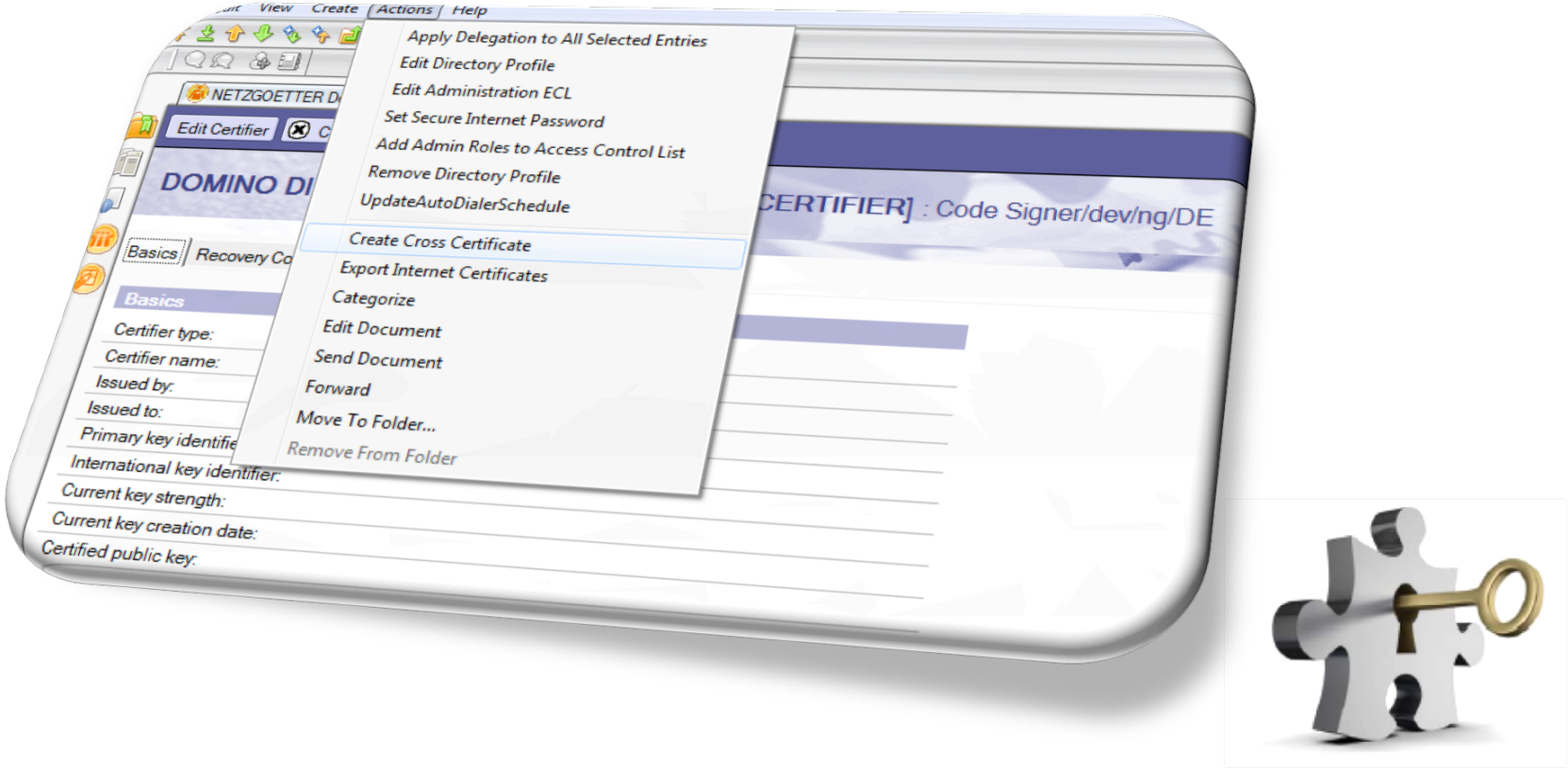
2. Öffentlichen Schlüssel des Internet Zertifikats querzulassen und an die Clients verteilen

Import des Zertifikats:



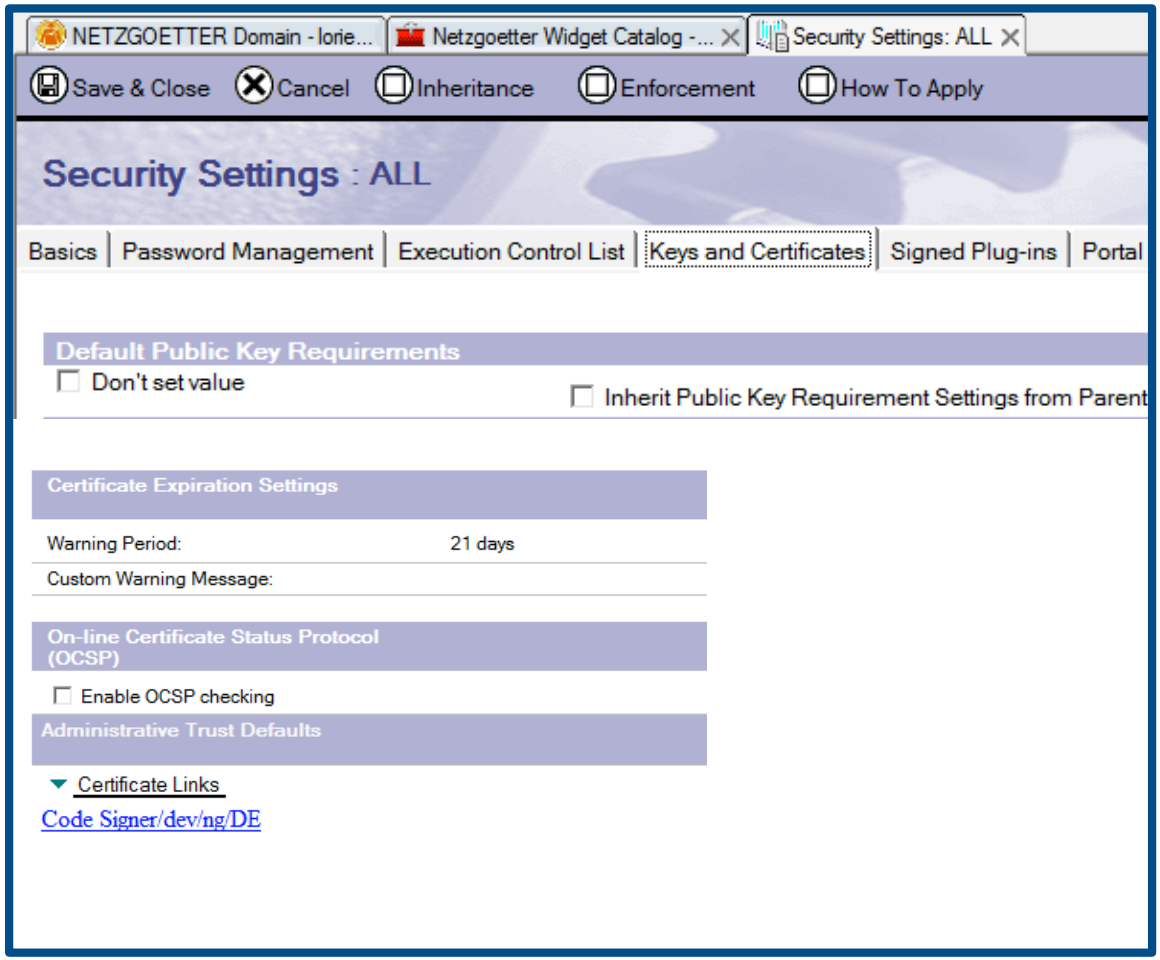
2. Öffentlichen Schlüssel des Internet Zertifikats querzulassen und an die Clients verteilen

Querzulassung des Zertifikats:



Sicherheit – Signieren von Extensions

2. Öffentlichen Schlüssel des Internet Zertifikats querzulassen und an die Clients verteilen

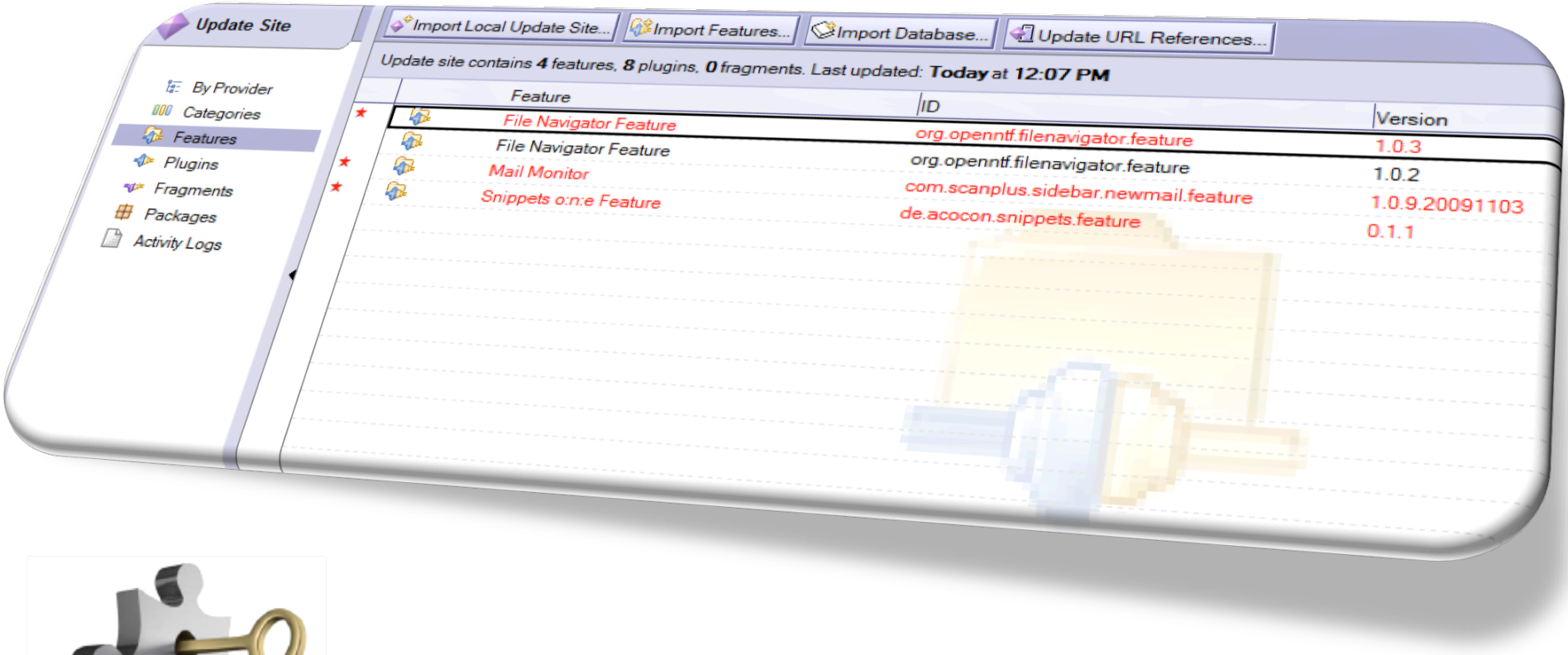


3. JAR-Dateien mit JAR-Signer-Tool signieren

Für die gewünschten Feature die JAR Dateien der Updatesite (sowohl im Features als auch im Plugins Ordner) mit dem Self-Signed Zertifikat/ Schlüsselpaar signieren.

```
%JAVA_HOME%\bin\jarsigner -verbose
    -keystore C:\sign-plugin\abx\mykeystore
    -storepass geheim
    -keypass geheim
    C:\work\Updatesite\feature\net.netzgoetter.feature.jar
    “Code Signer"
```

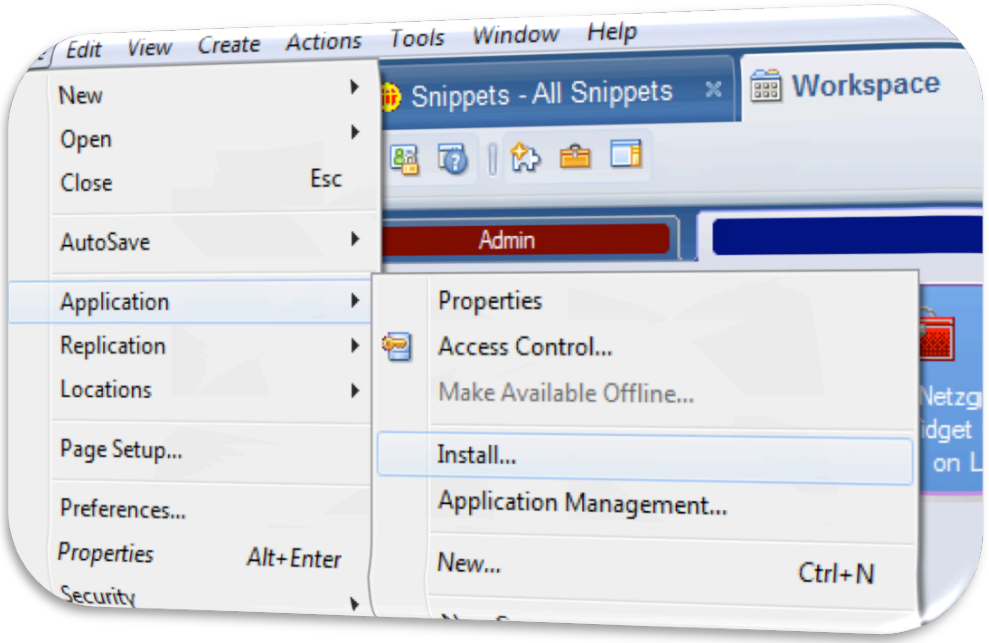
4. Eclipse Update Site erstellen und signierte lokale Updatesite importieren



Verteilung - Manuell

Es gibt drei Möglichkeiten:

1. Der User installiert das neue Feature selbst

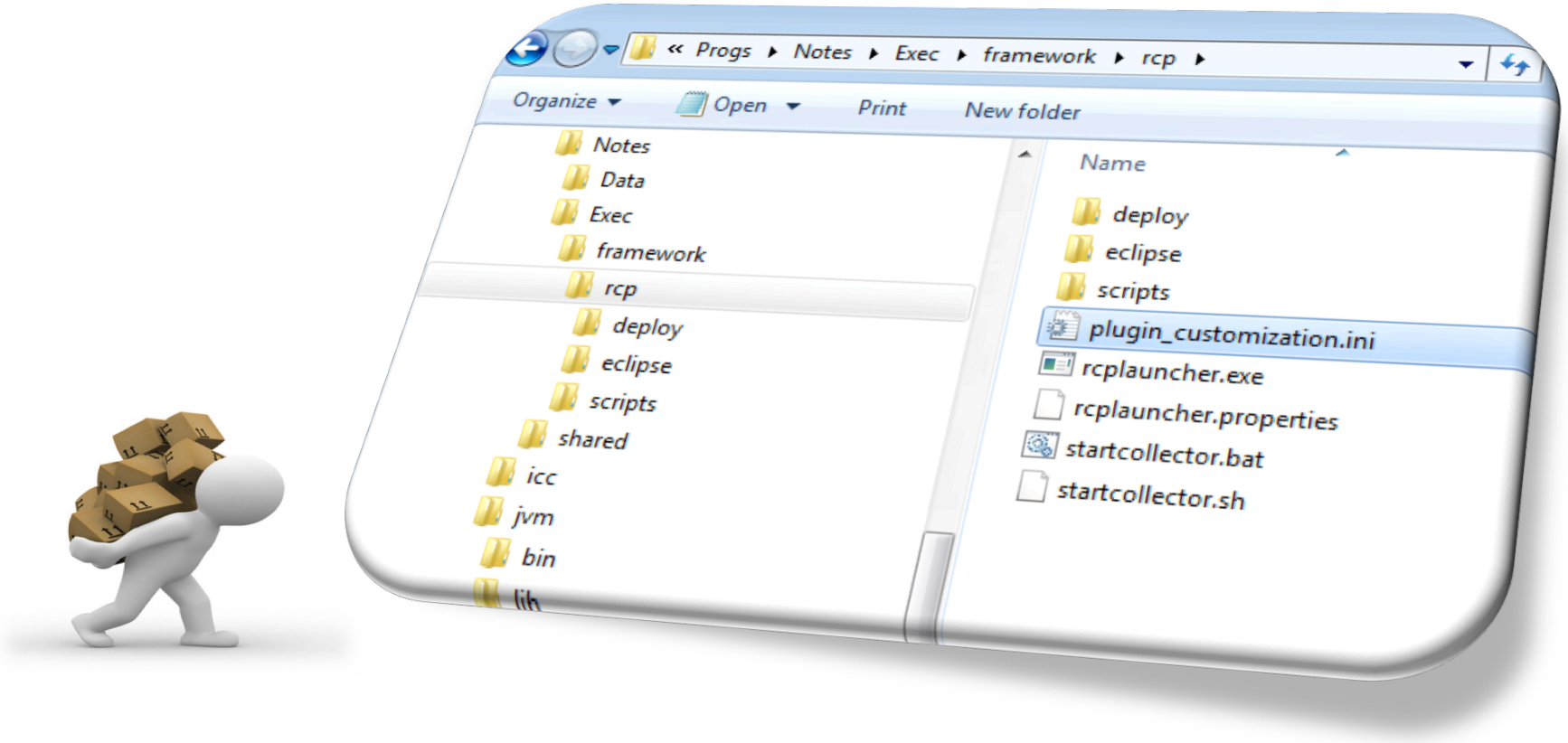


Verteilung - Manuell

Es gibt drei Möglichkeiten:

1. Der User installiert das neue Feature selbst

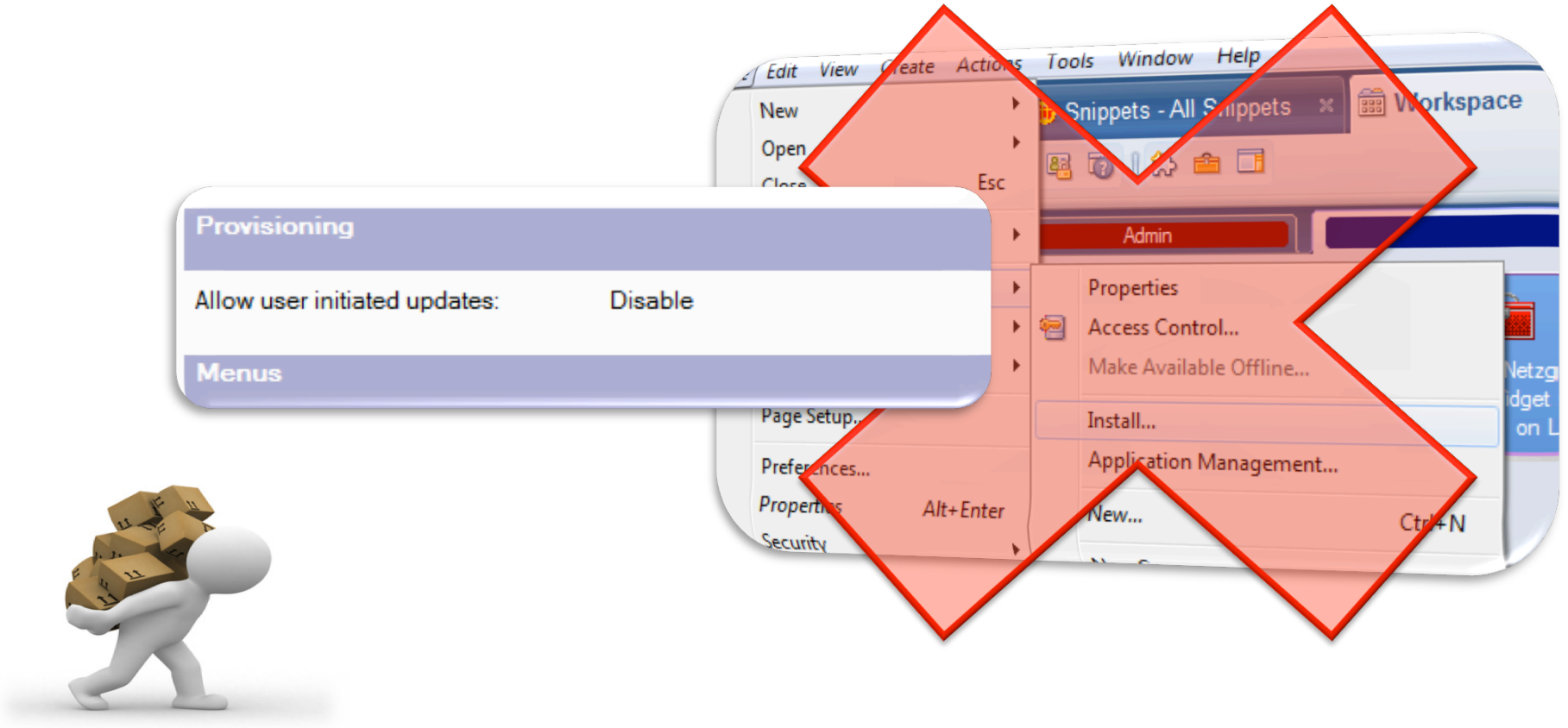
   Plugin_customization.ini erweitern:
   com.ibm.notes.branding/enable.update.ui=true



Verteilung - Manuell

Es gibt drei Möglichkeiten:

1. Der User installiert das neue Feature selbst

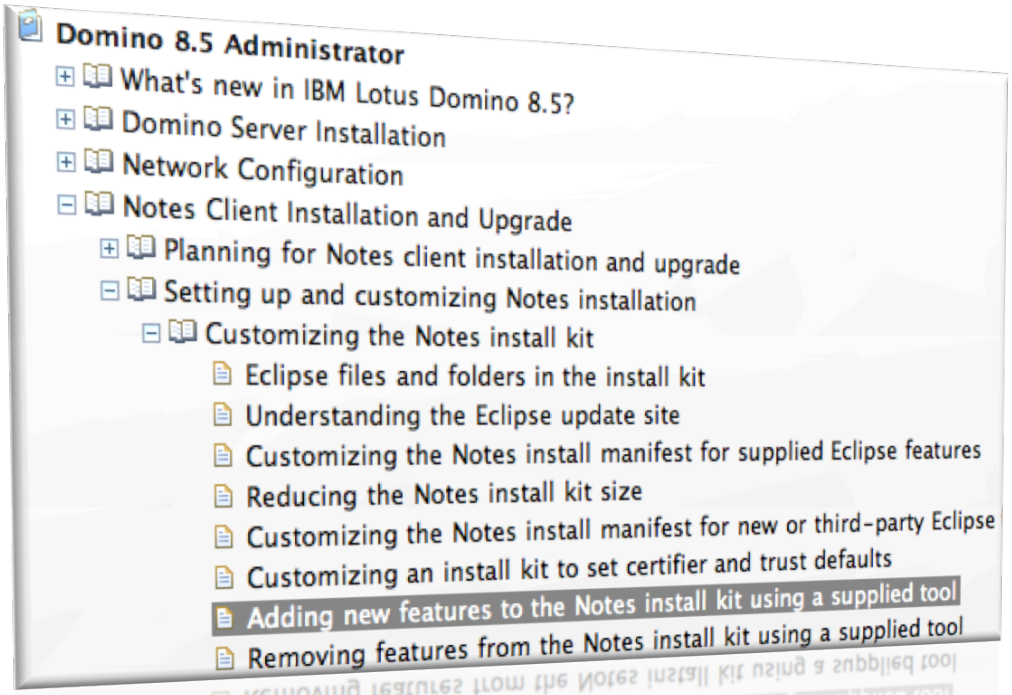


Verteilung - Paketierung

Es gibt drei Möglichkeiten:

1. Der User installiert das neue Feature selbst
2. Anpassung des Installationspackets

Siehe Notes Admin Hilfe



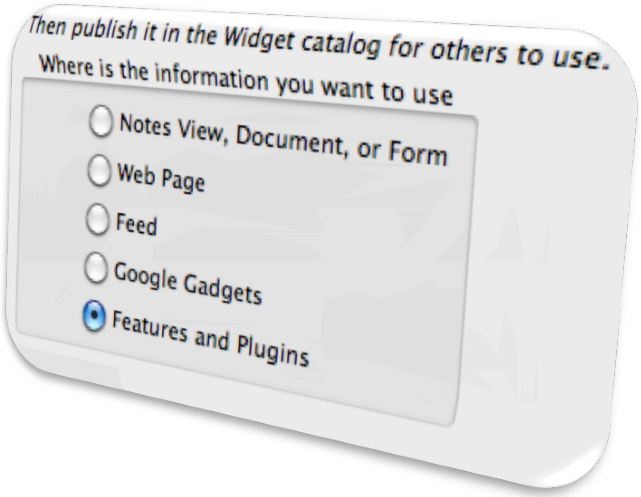
Verteilung - Widget

Verteilung über MyWidgets und Widget Katalog

Ein Widget ist ein XML-Dokument, welches auch Widget Descriptor genannt wird.

Der Widget Descriptor kann von Hand erzeugt.

Ab 8.5.1 gibt es im New Widget Dialog auch die Option
„Features & Plugins“



Verteilung - Widget

```
<?xml version="1.0" encoding="UTF-8"?>
<webcontextConfiguration version="1.1">
<palleteItem contributeToSideshelfOnStartup="false" doubleClickCommandId=""
        id="177388592" imageUrl="" modified="false" title="Mail Monitor"
        providerId="com.ibm.rcp.toolbox.prov.provider.ToolboxProvisioning"
        url="nrpc://lorien.netzgoetter.de/__c1257670004dbd9b">
        <data>
        <installManifest>
        <install>
                <installfeature id="com.scanplus.sidebar.newmail.feature"
                name="Mail Monitor" version="1.0.9.20091103">
                        <requirements>
                                <feature id="com.scanplus.sidebar.newmail.feature"
                                version="1.0.9.20091103" match="perfect" />
                        </requirements>
                </installfeature>
        </install>
        </installManifest>
        </data>
</palleteItem>
</webcontextConfiguration>
```

Verteilung - Widget

```
<?xml version="1.0" encoding="UTF-8"?>
<webcontextConfiguration version="1.1">
<palleteItem contributeToSideshelfOnStartup="false" doubleClickCommandId=""
        id="177388592" imageUrl="" modified="false" title="Mail Monitor"
        providerId="com.ibm.rcp.toolbox.prov.provider.ToolboxProvisioning"
        url="nrpc://dom.netzgoetter.de/__c12576702345dbd1c">
        <data>
        <installManifest>
        <install>
                <installfeature id="com.scanplus.sidebar.newmail.feature"
                 name="Mail Monitor" version="1.0.9.20091103">
                        <requirements>
                                <feature id="com.scanplus.sidebar.newmail.feature"
                                 version="1.0.9.20091103" match="perfect" />
                        </requirements>
                </installfeature>
        </install>
        </installManifest>
        </data>
</palleteItem>
</webcontextConfiguration>
```

Verteilung - Widget

```
<?xml version="1.0" encoding="UTF-8"?>
<webcontextConfiguration version="1.1">
<palleteItem contributeToSideshelfOnStartup="false" doubleClickCommandId=""
        id="177388592" imageUrl="" modified="false" title="Mail Monitor"
        providerId="com.ibm.rcp.toolbox.prov.provider.ToolboxProvisioning"
        url="nrpc://dom.netzgoetter.de/__c12576702345dbd1c">
        <data>
        <installManifest>
        <install>
                <installfeature id="com.scanplus.sidebar.newmail.feature"
                 name="Mail Monitor" version="1.0.9.20091103">
                        <requirements>
                                <feature id="com.scanplus.sidebar.newmail.feature"
                                 version="1.0.9.20091103" match="perfect" />
                        </requirements>
                </installfeature>
        </install>
        </installManifest>
        </data>
</palleteItem>
</webcontextConfiguration>
```

Verteilung - Widget

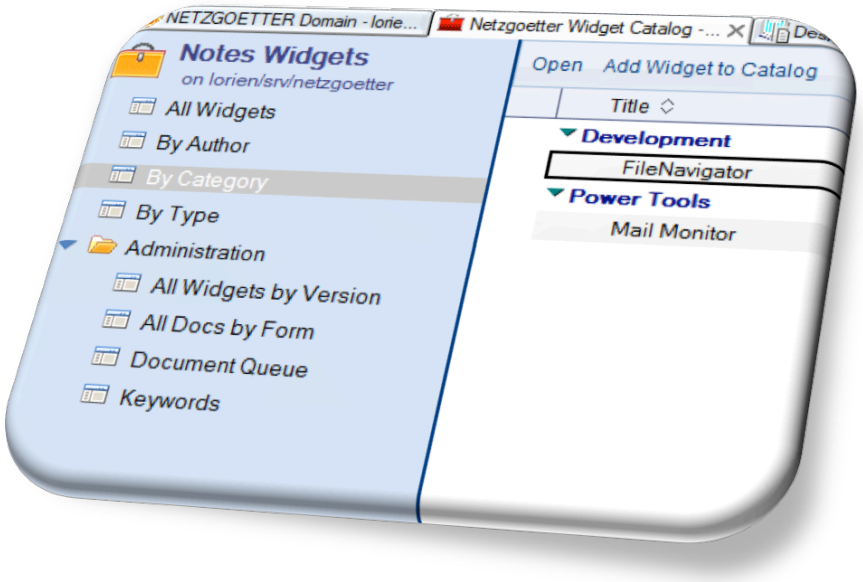
```
<?xml version="1.0" encoding="UTF-8"?>
<webcontextConfiguration version="1.1">
<palleteItem contributeToSideshelfOnStartup="false" doubleClickCommandId=""
        id="177388592" imageUrl="" modified="false" title="Mail Monitor"
        providerId="com.ibm.rcp.toolbox.prov.provider.ToolboxProvisioning"
        url="nrpc://dom.netzgoetter.de/__c12576702345dbd1c">
        <data>
        <installManifest>
        <install>
                <installfeature id="com.scanplus.sidebar.newmail.feature"
                 name="Mail Monitor" version="1.0.9.20091103">
                        <requirements>
                                <feature id="com.scanplus.sidebar.newmail.feature"
                                 version="1.0.9.20091103" match="perfect" />
                        </requirements>
                </installfeature>
        </install>
        </installManifest>
        </data>
</palleteItem>
</webcontextConfiguration>
```

Verteilung - Widget

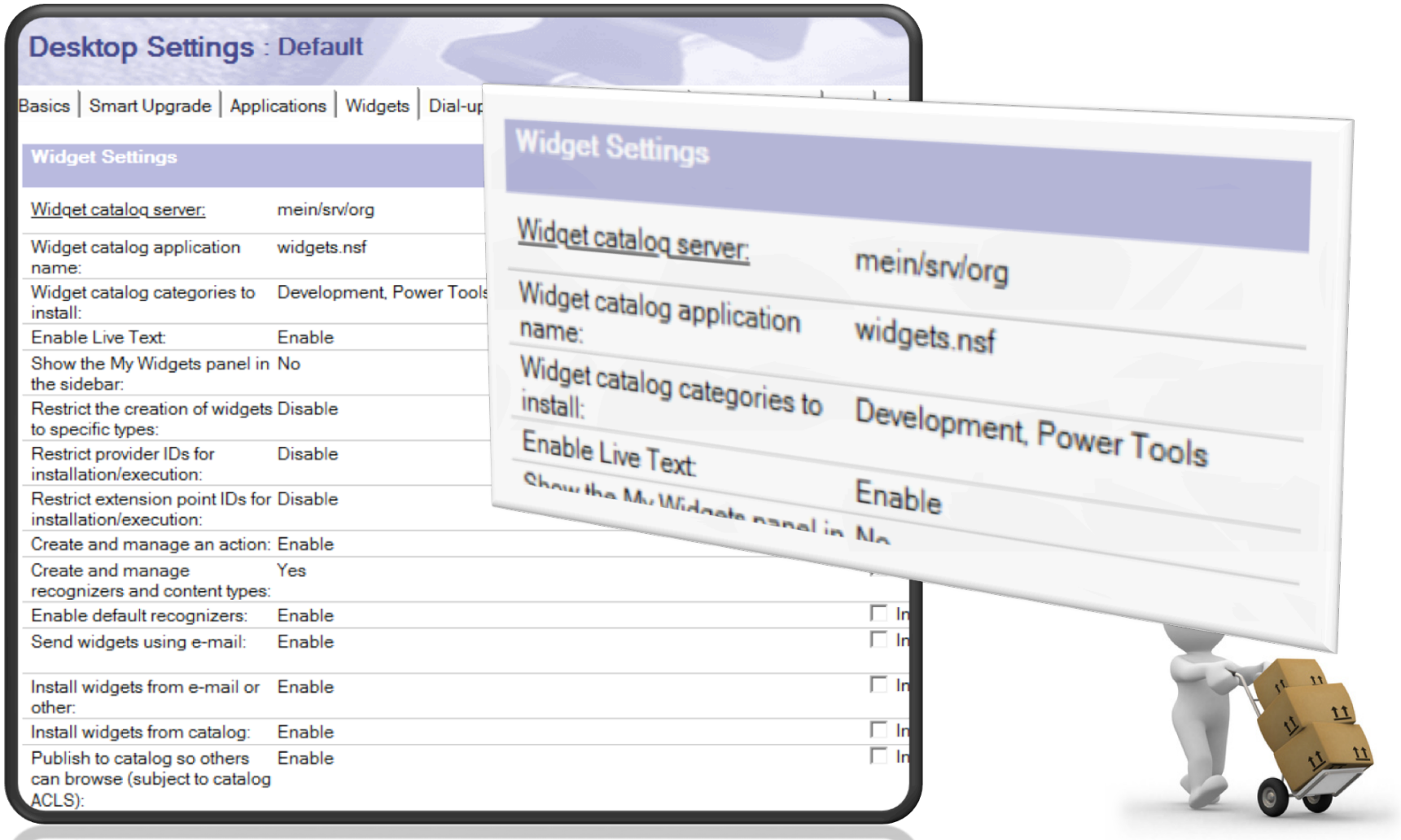
Das so erzeugte Widget wird auf einem Client unter MyWidgets deployed und in den Widget Katalog übertragen.

Wenn noch nicht vorhanden, muß der Katalog basierend auf dem toolbox.ntf Template auf einem Domino Server erstellt werden.

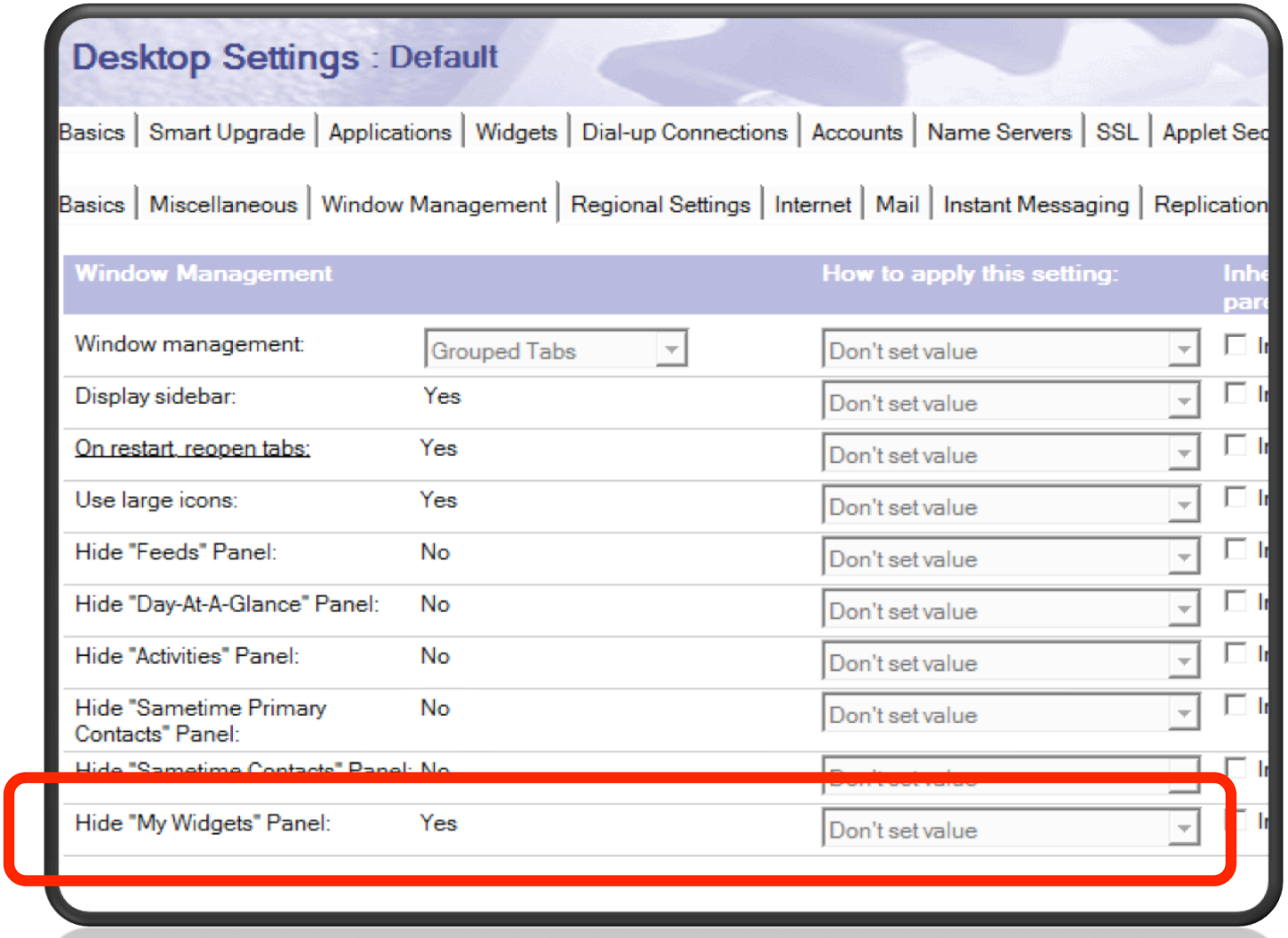
Der Widget Katalog wird den Benutzern per Desktop Policy Setting zugewiesen.



Verteilung - Widget



Verteilung - Widget



Verteilung - Widgets

Wird den Usern über die Policy der Widget Catalog und somit das Widget zugewiesen. Erfolgt die Installation automatisiert und ohne Benutzerinteraktion.

Nach dem Deployment der Extension wird der User aufgefordert den Client einmal neu zu starten.

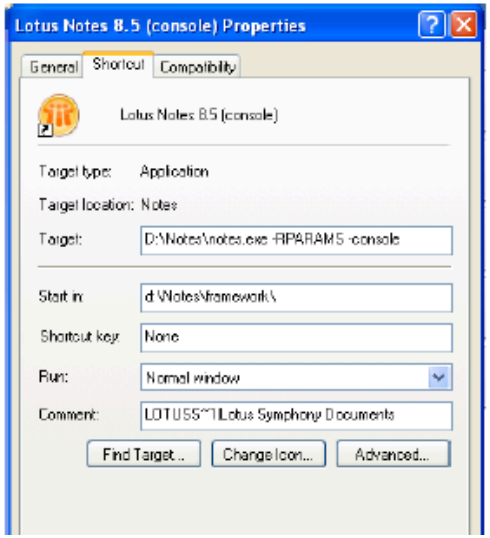
Wird ein Widget später aus dem Katalog entfernt, wird automatisch auf den Clients die Extension deinstalliert.

Debugging using OSGi console

`<notes>/data/workspace/.config/rcpinstall.properties`

- com.ibm.rcp.provisioning.level=FINEST
- com.ibm.rcp.toolbox.level=FINEST

`<notes>/notes.exe -RPARAMS -console`



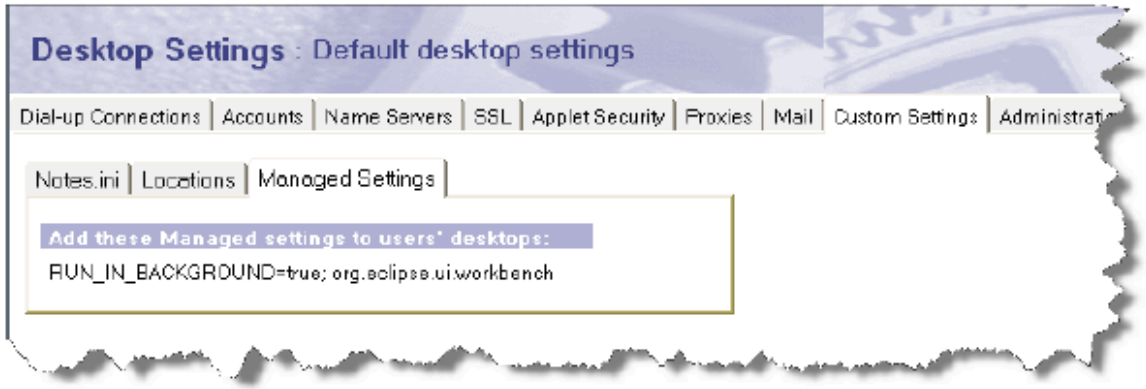
Tips

```
2009/10/12 10:03:14.046 SEVERE
CLFWW0017E: Cannot install extensions
for org.eclipse.ui.views because the
administrator has restricted
extension installation.
::class.method=com.ibm.rcp.dynamic.ex
tensions.DynamicExtensionRegistry.loa
dStore() ::thread=Worker-4
::loggername=com.ibm.rcp.dynamic.exte
nsions
```

Interesting Eclipse settings

<notes>/data/workspace/.metadata/.plugins/
org.eclipse.core.runtime/.settings

- org.eclipse.ui.workbench.prefs
  - RUN_IN_BACKGROUND=true



Verteilung - Widgets

Beachtenswert:

- Mehrere Widget-Kategorien werden in den Desktop Settings per Komma getrennt!!! (Kein Mehrfachwertefeld)

- Es gibt mehre Stellen die das Verhalten von Widgets steuern.

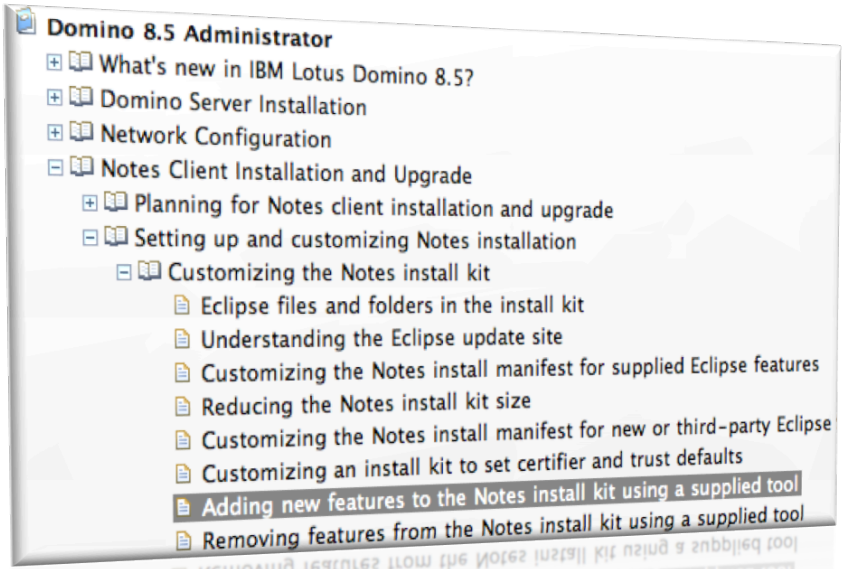
- Immer nur signierte Plugins beim Roll Out verteilen.

- Am besten ein selbstsigniertes Zertifikat verwenden und über Security Settings verteilen.

How To‘s & Dokumentation:

- Notes Admin Hilfe
  - Internet Zertifikate
  - Widget Deployment

- Domino Wiki



- Mikkel Heisterberg Blog:
  http://lekkimworld.com

- Planet Lotus / Blogs

That‘s it .... Vielen Dank

Resources

Plugin development RedWiki
http://bit.ly/dXfz8C

Plugin deployment (ILUG 2010 slides)
http://bit.ly/ilug2010slides