

## Bestandteile von CA

---

- Laufen im Rich Client (Notes oder Expeditor) oder via Portal Server im Browser
- NSF-Komponenten sind Notes-Designelemente
- Eclipse Komponenten sind Eclipse ViewParts
- lokale oder Websphere Portlets
- für die Nutzung von Domino Komponenten im Portal Server wird das Domino Application Portlet benötigt, allerdings wird der Property Broker nicht unterstützt
- alternativ kann mit der Portlet Factory das Portlet selbst erstellt werden um den Broker zu unterstützen, Interfaces müssen aber selbst implementiert werden
- Pages, Forms, Views, Folders, Framesets und Navigators können in Komponenten genutzt werden

## Erstellung von CA

---

- Komponenten designen – Notes Design
- Komponenten bauen – WSDL hinzufügen mittels Property Broker Editor
- Zusammenstellen und Verbindungen von Komponenten zu CA
- CA verteilen
- Beachte, dass Masken und Seiten in der Größe und Lage variabel sind
- Beachte, dass Aktionen und Eigenschaften beim Wiring Rechenzeit erfordern

## Erstellen von Notes C

---

- Properties festlegen, die veröffentlicht werden
- Aktionen der C definieren
- WSDL erstellen
- WSDL in die NSF importieren
- Designelemente mit Aktionen und Properties verbinden (LS)
- Eigenschaften veröffentlichen
  - Per LS API
  - Ansichten oder Ordner können Spalte mit dem WSDL verbinden (advanced tab)
- Eigenschaften konsumieren
  - Notes Aktion verweist auf eine WSDL Aktion (advanced tab einer aktion)
- Neue LS API
  - NotesSession.GetPropertyBroker
  - NotesPropertyBroker.SetPropertyValue
  - NotesPropertyBroker.Publish
- Eclipse Komponenten
  - Property Broker JAVA API

## Neue LS Klassen

---

*NotesSession*

- **NotesSession.GetPropertyBroker( )** -- Returns NotesPropertyBroker.

## NotesPropertyBroker

- **NotesPropertyBroker.InputPropertyContext** (property) -- Returns an array of input NotesProperties. For current releases, only the first member of the array will be populated.
- **NotesPropertyBroker.Namespace** (property) -- String. Provides the Read/Write default namespace of interest for PropertyBroker.
- **NotesPropertyBroker.HasProperty(Name)** -- Boolean. Indicates whether a new property exists.
- **NotesPropertyBroker.GetProperty(Name)** -- Returns NotesProperty.
- **NotesPropertyBroker.GetPropertyValue(Name)** --Returns the value for a specified property.
- **NotesPropertyBroker.SetPropertyValue(Name, Value)** -- Returns NotesProperty. It creates a backend object and populates it with the value. This method throws an error if the property name is not defined by WSDL..
- **NotesPropertyBroker.ClearProperty** -- A new method that purges any new or modified property, not for input properties.
- **NotesPropertyBroker.Publish** -- A new method that publishes all new and modified properties.

## NotesProperty

- **NotesProperty.Values** (property) - Read/Write for simple properties: Returns a Variant array of type String. Single value properties contain data in first array element. As input, accepts Variant or Array of type String. The Array must be homogeneous. At this time, an Array can contain only one String entry. Input properties cannot be set.
- **NotesProperty.Name** (property) -- Read only. Returns property name as string.
- **NotesProperty.TypeName** (property) -- Read only. Returns type name as string.
- **NotesProperty.NameSpace** (property) -- Read only. Returns namespace as string.
- **NotesProperty.Title** (property) -- Read only. Returns title as string.
- **NotesProperty.isInput** (property) -- Read only. Returns true if this is an input property, otherwise false. Returns Boolean.
- **NotesProperty.Description** (property) -- Read only. Returns description as string.
- **NotesProperty.Publish( )** -- A new method that publishes this property if modified.
- **NotesProperty.Clear( )** -- A new method that purges this property if new or modified.

## Deployment und Testing

---

- CA können in separaten NSF oder gemeinsam mit den C gespeichert sein
- Special Notes URLs
  - notes:///0000000000000E00 - opens the current user's Mail database
  - notes:///0000000000000E01 - opens the current user's Contacts database
- Datenbank Start Option muss auf die CA zeigen

## Debugging property broker

---

The following property broker commands are supported:

- pbsh a  
This show all actions.
- pbsh a <owner>  
This show all actions that belong to that owner.

**Note** All commands that accept scoping parameters (<owner> with pbsh a, pbsh o, pbsh p, pbsh w) accept wildcard characters. This can be any regular expression. At a minimum this should be, for example, "XYZZY\*" and "\*XYZZY"

- pbsh aa  
This shows all active actions.
- pbsh o  
This shows all owners.
- pbsh o <owner>  
This shows all properties, actions and wires that belong to that owner.
- pbsh p  
This shows all properties by owner.
- pbsh p <owner>  
This shows all properties for this owner (string owners only).
- pbsh w  
This shows all enabled wires.
- pbsh w <owner>  
This shows all enabled wires belonging to that owner.
- pbsh aw  
This shows all wires.
- pbsh ns  
This shows registered name spaces.
- pbt <Action>  
This traces the path for the specified action. This produces trace messages when a property change results in the watched action being fired.
- pbt <Action> <Page>  
This traces the path for the specified action on a specific page. This produces trace messages when a property change results in the watched action being fired.
- pbut <Action>  
This untraces the specified action and removes the trace.
- pbt <Property>  
This traces the path for the specified property.
- pbt <Property> <Page>  
This traces the path for the specified property on a specific page.
- pbut <Property>  
This untraces the specified property and removes the trace.
- pbt <Wire>  
This traces the path for the specified wire. This produces trace messages when a property change connects up with a watched wire.
- pbt <Wire> <Page>  
This traces the path for the specified wire on a specified page. This produces trace messages when a property change connects up with a watched wire.
- pbut <Wire>  
This untraces the specified wire and removes the trace.

- pblt

This shows a list of the currently traced properties, actions, and wires.

## Wiring

---

- WSDL speichert für jede C zum Nutzen im Wiring
- WSDL importieren im Wiring einer CA
- Merge WSDL – für Websphere sinnvoll da alle Eigenschaften zusammen

## Standard-Komponenten

---

- Inbox
- My Contacts

## PIM – C

---

<b>PIM Component View</b>	<b>Name</b>	<b>Value</b>
Mail Navigator	com.ibm.rcp.csiviews.viewpart.dburl (required)	notes:///000000000000E00/MailFS?OpenFrameset
Mail Mini View	com.ibm.rcp.csiviews.viewpart.dburl (required)	notes:///000000000000E00/Switcher MV Form For Mail?OpenForm
Mail View	com.ibm.rcp.csiviews.viewpart.folderalias (optional)	INBOX ALL_DOCUMENTS
Calendar Navigator	com.ibm.rcp.csiviews.viewpart.dburl (required)	notes:///000000000000E00/CalendarFS?OpenFrameset
Calendar Mini View	com.ibm.rcp.csiviews.viewpart.dburl (required)	notes:///000000000000E00/Switcher MV Form For Calendar?OpenForm
Calendar View	com.ibm.rcp.csiviews.viewpart.folderalias (optional)	CALENDAR
Notes Todo Navigator	com.ibm.rcp.csiviews.viewpart.dburl (required)	notes:///000000000000E00/ToDoFS?OpenFrameset
Notes Todo View	com.ibm.rcp.csiviews.viewpart.folderalias (optional)	notes:///000000000000E00/ToDoFS?OpenFrameset
Contacts Navigator	com.ibm.rcp.csiviews.viewpart.dburl (required)	notes:///000000000000E01/Contacts?OpenFrameset
Contacts View	com.ibm.rcp.csiviews.viewpart.folderalias (optional)	ALL_CONTACTS RECENT_CONTACTS RECENT_COLLABORATIONS

## **Container elements**

---

- Notes View and Notes PIM containers, unterstützen nur das selection event und publishing properties
- Hosted on-demand container  
(<http://publib.boulder.ibm.com/infocenter/hodhelp/v10r0/index.jsp?topic=/com.ibm.hod.doc/doc/beans/com/ibm/eNetwork/beans/HOD/Session.html>.)
- Symphony Spreadsheet container
- Web browser container
- Notes document container