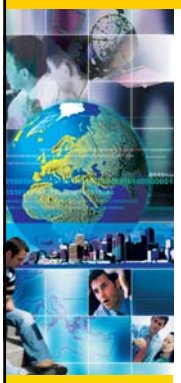

Workgroup Computing Praktikum

Using Actions to Automate Tasks

University of Paderborn
Business Computing 2 – Information Management & Office Systems
Faculty of Business Administration, Business Computing & Economics
Prof. Dr. Ludwig Nastansky
Warburger Str. 100, D-33098 Paderborn
Tel.: +49--5251--60-3368
http://gcc.upb.de

- **Eine Action ist ein zu einer Page, Form, Subform, Folder oder View zugehöriges Programm zum Automatisieren von Anfragen und Aufgaben des Benutzers**
- **Anwendungsgebiete**
  - Sich wiederholende Aufgaben beschleunigen
  - Aktualisieren der dargestellten Informationen
  - Datenbanknutzung vereinfachen
  - Komplexe Berechnungen und Datenmodifikationen durchführen
  - Benutzereingaben auf Fehler überprüfen
- **Beispiele**
  - Menübefehle über einen Action-Button aufrufen
    - Dokument erstellen, editieren und/oder speichern
  - Das aktuelle Dokument als Mail verschicken
  - Vom Benutzer ausgewählte Dokumente in einen Ordner verschieben

Vorüberlegungen

- **Welche Funktion soll die Action erfüllen?**
- **Welche Programmiersprache eignet sich hierfür am besten?**
- **Wie und wo soll die Action angezeigt werden?**
- **Wie soll die Action formatiert und dargestellt werden?**

Actions erstellen

- **Von Actions unterstützte Programmiersprachen**
  - Simple Action
    - Können nicht angepasst werden und werden im **Web** nicht unterstützt
    - Stellen Standardfunktionalitäten für Dokumentenmanagement / -manipulation (copy to Folder, mark Document Unread, …) zur Verfügung
  - @Formulas
  - LotusScript
  - (Common) JavaScript
- **Actions können auf Pages, Forms, Subforms, Folders, und Views eingesetzt werden**
  - Actions können in der Action Bar am oberen Rand des Designelements und/oder im Menü „Actions“ angezeigt werden
- **Actions können dynamisch mittels der Hide-When-Properties ein- und ausgeblendet werden**
  - Bsp.: Eine Action, die ein Dokument im Editiermodus öffnet, soll nicht angezeigt werden, wenn sich das Dokument schon im Editiermodus befindet.

System Actions

- **Neu erstellte Forms und Views enthalten noch keine Actions**
- **Es gibt einige vordefinierte System Actions, die der Form/View hinzugefügt werden können**
  - Categorize
  - Edit Document
  - Send Document
  - Forward
  - Move to Folder
  - Remove from Folder
- **System Actions können nicht verändert werden**
- **System Actions werden im Web nicht unterstützt**
  - Stattdessen eigene Actions erstellen, die die gleichen Funktionen realisieren.

@Commands I

- **@Commands ermöglichen dem Programmierer, im Programmcode auf Menübefehle des Clients zuzugreifen**
  - Viele @Commands funktionieren nicht im Web (siehe Domino Designer Hilfe)

@Command	Funktion	Beispiel
Compose	Erstellt ein neues Dokument mit der angegebenen Form	@Command([Compose]; “Response“)
FileSave	Speichert das aktuelle Dokument	@Command([FileSave])
FileCloseWindow	Schließt das aktuell geöffnete Fenster	@Command([CloseWindow])
EditDocument	Öffnet das aktuell ausgewählte Dokument im Editiermodus	@Command([EditDocument])

@Commands II

- **Um das Benutzen aller Buttons im Web zu ermöglichen ist es notwendig, in den Database Properties die Option „Web access: Use JavaScript when generating pages“ auszuwählen**
  - Ohne diese Einstellung erkennt Domino nur den ersten Action Button und behandelt ihn als Submit Button (Speichern und Schließen)
- **Reihenfolge der Ausführung von Formula-Statements**
  - Von links nach rechts
  - Von oben nach unten
  - Jeder Befehl wird beendet, bevor der nächste ausgeführt wird

@Commands III

- **Es gibt @Commands, die erst nach allen anderen @Functions ausgeführt werden, z.B. FileCloseWindow**
- **Mit @PostedCommand kann man @Commands auch nach allen anderen @Functions ausführen lassen**

Shared Actions

- **Shared Actions werden als „Shared Code“ zum Einsatz auf mehreren Designelementen der Datenbank erstellt**
  - Code kann wieder verwendet werden
  - Einfachere Wartung
  - Kein Unterschied zu „einzelnen“ Actions auf einem Designelement
- **Eine individuelle Konfiguration der Action für jedes einzelne Designelement ist nicht möglich**
- **Die Position einer Shared Action in der jeweiligen Action Bar des Designelements kann individuell konfiguriert werden**

In 10 Schritten zur Shared Action

- **Erstellen einer Shared Action:**
  1. Datenbank in Designer öffnen
  2. Im Design Pane auf Shared Code → Actions klicken
  3. Klick auf new shared Code (Property Box öffnet sich)
  4. Shared Action programmiern
  5. Shared Action speichern
- **Shared Action zu einer Form oder View hinzufügen:**
  6. Form oder View im Designer öffnen, die die Action enthalten soll
  7. Bei einer View: Create → Insert Shared Action vom Menü auswählen
     Bei einer Form: Create → Action → Insert Shared Action
  8. Die Shared Action auswählen und auf Insert klicken
  9. Klick auf Done
  10. In der Notes Vorschau ansehen

Sub Actions

- **Sub Actions ermöglichen das Gruppieren mehrerer (Shared) Actions unter einem einzelnen Menüpunkt**
  - Sub Actions enthalten keinen eigenen Code
- **Sub Actions sind für den Benutzer hilfreich, um bei einer großen Anzahl von Actions in einem Designelement die Übersicht zu bewahren**
  - z. B. Kategorisierung ähnlicher Actions unter einem gemeinsamen Oberbegriff: Siehe gruppierte Einzel-actions unter „Reply“ in der Actionbar des Ordners Inbox der eigenen Mail-Datenbank

Action Bar konfigurieren

- **Die Action Bar kann mittels der Action Bar Properties den eigenen Bedürfnissen angepasst werden**
  - Ausrichtung der Buttons
  - Icons
  - Hintergrundbilder
  - Aussehen und Reihenfolge der Buttons bestimmen
- **Um die Action Bar einer Anwendung auch im Web nutzen zu können, ist es nötig, die Action Bar als JavaApplet darzustellen (Action Bar Properties)**
  - Bietet eine Scrollbar, falls nötig
  - Die Darstellung erfolgt genauso wie in Notes
  - Unterstützt geschachtelte Actions (Sub Actions)

Action Hotspots

- **Action Hotspots sind programmierbare Bereiche einer Form, Subform oder Page, welche automatisch Aufgaben ausführen können**
- **Action Hotspots können auf Texten oder Grafiken und Bildern hinterlegt werden**
  - Text oder Grafik/Bild selektieren und im Menü Create → Hotspot → Action Hotspot... auswählen
  - Action Hotspots können wie „gewöhnliche“ Actions mittels Simple Actions, @Formulas, LotusScript oder (Common) JavaScript programmiert werden

Übung

Das Erstellen und Editieren von Dokumenten soll automatisiert werden. Die Benutzer sollen die Möglichkeit haben, ...

- **... einen Auftrag oder eine Frage aus der Order View zu erstellen**
- **... eine Order zu editieren, wenn das Dokument geöffnet ist**
- **... eine Frage oder Antwort aus der Questions and Annotions View erstellen**
- **... eine Frage oder Antwort zu editieren, wenn das Dokument geöffnet oder in der Questions and Annotions View ausgewählt ist**
- **... die einzelnen Dokumente durch Klicken eines Buttons zu speichern, ohne das Menü benutzen zu müssen**

Fallbeispiele:

- Außendienstmitarbeiter einer Firma, welche mit einem Webbrowser auf die Notes-Datenbank zugreifen müssen, beschweren sich, dass die in Forms und Views genutzten Buttons nicht zu funktionieren scheinen: Viele Buttons funktionieren nicht, andere scheinen immer nur Daten an den Server zu senden, obwohl das gar nicht ihre designierte Funktionalität ist. Welche Ursache hat dieses Verhalten?

- Nachdem das beschriebene Problem behoben werden konnte, wünschen sich die Außendienstmitarbeiter eine übersichtlichere Action Bar. Womit kann das erzielt werden? Was hat man hierbei zu beachten?