



Praktikum Workgroup Computing 1

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>



- ➔ **Action = zu einer View oder Form zugehöriges Programm**
- ➔ **Automatisieren von Aufgaben**
 - ➔ Beschleunigen sich wiederholender Aufgaben
 - ➔ Informationen erneuern
 - ➔ Die Benutzung der Datenbank vereinfachen
 - ➔ Komplexe Berechnungen durchführen
 - ➔ Auf Fehler überprüfen
- ➔ **Beispiele**
 - ➔ Notes Menü Befehle als Action-Button darstellen
 - ➔ Dokument erstellen, editieren und/oder speichern
 - ➔ Das aktuelle Dokument als Mail verschicken
 - ➔ Vom Benutzer ausgewählte Dokumente in einen Folder schieben
 - ➔ Komplexe Formeln berechnen, die nicht im Dokument gespeichert werden müssen



- **Welches Ziel soll die Action haben?**
- **Welche Programmiersprache eignet sich am besten?**
- **Wie und wo soll die Action angezeigt werden?**
- **Wie soll die Action formatiert werden?**



- **Programmieroptionen für Actions**
 - Simple Action
 - Können nicht angepasst werden und werden im Web nicht unterstützt
 - @Formulas
 - LotusScript
 - JavaScript
- **Wie bzw. Wann werden Actions angezeigt**
 - Wenn die View, zu der die Actions gehören, geöffnet wird
 - Wenn ein Dokument mit der Form, zu der die Actions gehören, geöffnet wird
 - Anzeige in der Action Bar (als Button)
 - Anzeige im Menü „Actions“
- **Verstecken von Actions mit Hide-When-Properties möglich und oft sinnvoll**
 - Bsp.: Action, das ein Dokument im Editiermodus öffnet, verstecken, wenn sich das Dokument im Editiermodus befindet



- ➔ **Neuerstellte Forms und Views enthalten erstmal 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 haben



- ➔ **Häufig benutzte @Commands (siehe Domino 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([FileCloseWindow])
EditDocument	Öffnet das aktuell ausgewählte Dokument im Editiermodus	@Command([EditDocument])



- ➔ **Um das Benutzen aller Buttons im Web zu ermöglichen ist es notwendig in den Database Properties auszuwählen „Web access: Use JavaScript when generating pages“**
 - ➔ Ohne diese Einstellung erkennt Domino nur den ersten Action Button und behandelt ihn als Submit Button (Speichern und Schließen)
- ➔ **Reihenfolge der Ausführung von @Commands**
 - ➔ Von rechts nach links
 - ➔ Von oben nach unten
 - ➔ Jeder Befehl wird beendet, bevor der nächste ausgeführt wird
 - ➔ Es gibt @Commands, die erst nach allen anderen @Functions ausgeführt werden, z.B. FileCloseWindow



- ➔ **Werden als Ressource für die gesamte Datenbank erstellt**
- ➔ **Einmal erstellt, können sie in alle Views und Forms eingefügt werden**
- ➔ **Müssen Änderungen an der Action durchgeführt werden, müssen diese nur einmal vorgenommen werden**
- ➔ **Shared Actions haben die gleichen Eigenschaften wie Standard Actions**
- ➔ **Die Eigenschaften werden einmal festgelegt und können nicht für jede Form/View angepasst werden**
- ➔ **Nur die Position kann jeweils geändert werden**



F1 50:50



STOP

15	DM 1 MILLION
14	DM 500.000
13	DM 250.000
12	DM 125.000
11	DM 64.000
10	DM 32.000
9	DM 16.000
8	DM 8.000
7	DM 4.000
6	DM 2.000
5	DM 1.000
4	DM 500
3	DM 300
2	DM 200
1	DM 100

Bei welchem @Command macht es in aller Regel keinen Sinn, eine Shared Action zu erstellen?

A: @Command([FileClose Window])

B: @Command([FileSave])

C: @Command([Compose];"Antwort")

D: @Command([EditCocument])



F1 50:50



STOP

15	DM 1 MILLION
14	DM 500.000
13	DM 250.000
12	DM 125.000
11	DM 64.000
10	DM 32.000
9	DM 16.000
8	DM 8.000
7	DM 4.000
6	DM 2.000
5	DM 1.000
4	DM 500
3	DM 300
2	DM 200
1	DM 100

Bei welchem @Command macht es in aller Regel keinen Sinn, eine Shared Action zu erstellen?

A: @Command([FileClose Window])

B: @Command([FileSave])

C: @Command([Compose];"Antwort")

D: @Command([EditCocument])



Sub Actions

- ➔ **Gruppieren von Actions mit ähnlichen Funktionen unter einer Haupt-Action**
- ➔ **Hilfreich, bei**
 - ➔ Einer großen Anzahl von Actions um den Benutzer nicht zu überfordern
 - ➔ Wenn Gruppieren ähnlicher Actions gewünscht ist



- ➔ **Die Action Bar anpassen**
 - ➔ Action Bar Properties
 - ➔ Darstellung anpassen
 - ➔ Ausrichtung der Buttons
 - ➔ Hintergrundbilder
 - ➔ Aussehen und Reihenfolge der Buttons bestimmen
 - ➔ Usw.
 - ➔ Darstellung fürs Web als Applet aktivieren
- ➔ **Action Bar als Java Applet darstellen**
 - ➔ Bietet eine Scrollbar, falls nötig
 - ➔ Die Darstellung erfolgt genauso wie in Notes
 - ➔ Unterstützt geschachtelte Actions (Sub Actions)



→ Action Hotspots

- Programmierbare Bereiche einer Form, Subform oder Page, die automatisch Aufgaben ausführen können
- Hotspots können sein:
 - Text
 - Grafiken/Bilder



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

- Eine Policy oder eine Question aus dem Policies View zu erstellen
- Eine Policy zu editieren, wenn das Dokument geöffnet ist
- Eine Question oder Answer aus dem Q&A View erstellen
- Question oder Answer zu editieren wenn das Dokument geöffnet ist oder im Q&A View ausgewählt ist
- Alle Dokumente durch klicken eines Buttons zu speichern, ohne das Menü benutzen zu müssen

→ Aufgaben

- Erstelle eine Liste der Actions, die Du brauchst und wo sie benutzt werden sollen
- Lege fest welche davon shared Actions sein können
- Erstelle die Actions und füge sie den genannten Views und Forms hinzu
- Verstecke die Actions, wenn sie nicht gebraucht werden. Z.B. ist im Editiermodus die Anzeige des Editieren Buttons unnötig und sollte daher nicht angezeigt werden