

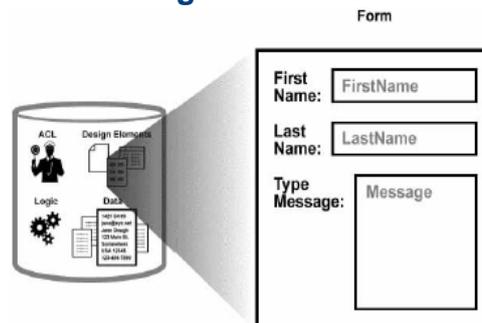


Workgroup Computing Praktikum

Working with Forms

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>

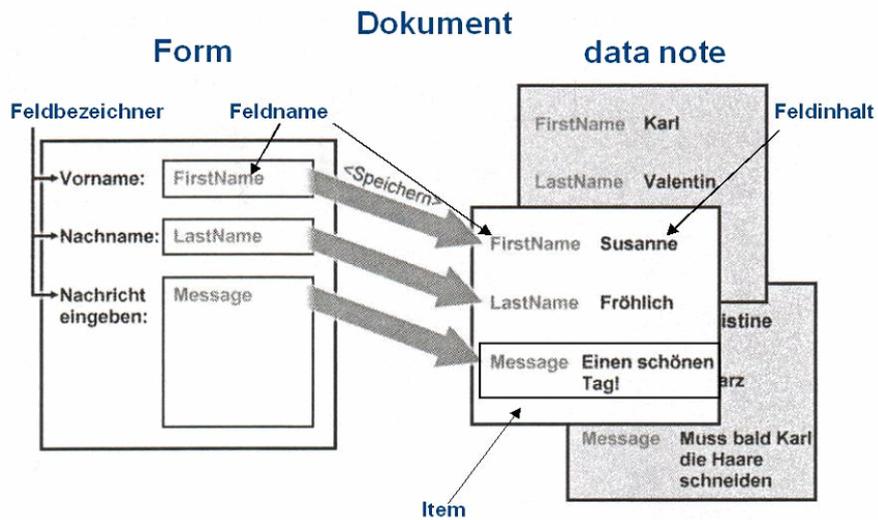
- **Forms („Masken“) sind Designelemente zur Eingabe und zur Betrachtung von Daten in einem Dokument**



- **Forms können statische Elemente wie Texte oder Bilder enthalten sowie eine eigene Anwendungslogik („Code“) zur Datenverarbeitung besitzen**

- ➔ **Eine Form ermöglicht die strukturierte Aus- und Eingabe von Daten, die innerhalb eines Dokumentes (genauer: data note) abgelegt werden.**
 - ➔ Dokumente (genauer: data notes) beinhalten gespeicherte Daten, die mittels einer Form eingegeben wurden
 - ➔ Standardmäßig werden Form und Dokumente unabhängig voneinander gespeichert
 - ➔ Beim Anzeigen eines Dokuments dient eine Form als Template, das die Darstellungsweise der Daten vorgibt

- ➔ **Forms enthalten Fields (Felder), in welche die Daten eingegeben werden**
- ➔ **Die Feldinhalte werden innerhalb einer data note in items gespeichert, welche dieselben Namen wie die Felder der Form besitzen**
- ➔ **Die Zuordnung einzelner items zu einem Feld erfolgt über den Namen des items zu einem Feld mit identischem Namen auf der in der data note spezifizierten Form**
 - ➔ Feldnamen müssen auf einer Form eindeutig sein
 - ➔ Es können Felder mit gleichem Namen auf unterschiedlichen Forms vorkommen



Wenn dann ...
... statische Inhalte wie Unternehmensdaten und Hintergrundinformationen, angezeigt werden sollen, benutze Pages zum Anzeigen und Strukturieren dieser statischen Informationen.
... Daten von einem Anwender erfasst und gespeichert werden sollen, verwende Forms , um die Informationen strukturiert eingeben und wieder auslesen zu können.



- ➔ **Fields (Felder) werden genutzt, um Daten einzugeben oder anzuzeigen**
 - ➔ Daten können in ein Feld von Hand eingegeben oder automatisch berechnet werden
 - ➔ In einem Feld eingegebene Daten werden im Allgemeinen gespeichert

- ➔ **Statische Elemente wie Texte, Grafiken und Tabellen können genutzt werden, um das Benutzerinterface zu gestalten und den Nutzer bei der Dateneingabe zu unterstützen**

- ➔ **Weitere Formelemente zur Unterstützung des Entwicklers und des Anwenders**
 - ➔ Subforms
 - ➔ Shared fields
 - ➔ Buttons und Actions



- ➔ **Elemente, welche nicht auf Pages, jedoch auf Forms genutzt werden können:**

Element	Page	Form
Layout Region	Nein	Ja
Subforms	Nein	Ja
Fields (Felder)	Nein	Ja
Shared Fields	Nein	Ja
Embedded Elements:		
➔ File upload		
➔ Scheduler	Nein	Ja
➔ Editor		

- ➔ **Merksatz: Alle Elemente, welche auf einer Page benutzt werden können, können auch auf einer Form benutzt werden. Der umgekehrte Schluss ist jedoch NICHT ZULÄSSIG!**



→ **Drei typische Schritte beim Erstellen einer neuen Form**

- 1. Schritt: Entwurf der Form
 - Wofür benötige ich die neue Form?
 - Welche Informationen soll der Benutzer eingeben?
 - Welche Informationen sollen automatisch berechnet werden?
 - Welche(r) Prozess(e) sollen laufen, wenn die Form abgeschickt/gespeichert wird?
 - Soll die Form auch im Web darstellbar sein?
- 2. Schritt: Erstellen der Form, Hinzufügen der Felder und Einstellen der relevanten Parameter
- 3. Schritt: Testen der Form in der angedachten Einsatzumgebung



- **Der Domino Designer ist eine Entwicklungsumgebung, welche Rapid Prototyping ermöglicht**
 - Die integrierte Entwicklungsumgebung des Domino Designers ermöglicht es Programmierern, in verhältnismäßig kurzer Zeit komplexe und aufwendig gestaltete Programme zu realisieren
- **Problem: Der unter Zeitdruck stehende Entwickler ist geneigt, Kompromisse beim Design der Anwendung einzugehen, um die festgesetzten Termine einhalten zu können**
- **Eine saubere Konzeption der zu erstellenden Anwendung und ein strukturiertes Vorgehen bei der Entwicklung erspart bei der späteren Wartung und Weiterentwicklung viel Arbeit und Kopfzerbrechen!**





- ➔ **Festlegen des Formnamens in den Form Properties**
 - ➔ Formname darf bis zu 64 Zeichen lang sein
 - ➔ Erlaubt sind Buchstaben, Leerzeichen, Zahlen und Interpunktion
 - ➔ Bei Formnamen wird zwischen Groß- und Kleinschreibung unterschieden

- ➔ **Festlegen eines oder mehrerer Aliase**
 - ➔ Hinzufügen der Aliase durch Anhängen an den Formnamen
 - ➔ Format: Formname | Alias1 | Alias2 | ...
 - ➔ Ein Alias ist eine Kurzbezeichnung, mit welcher die Form im Programmcode referenziert wird
 - ➔ Es sollten ausschließlich Aliase im Programmcode genutzt werden
 - ➔ Aliase sollten niemals gelöscht oder geändert werden
 - ➔ Vorteile: Einfaches Umbenennen einer Form, ohne dass Programmcode an den neuen Formnamen angepasst werden müsste



- ➔ **Überlegungen beim Anlegen neuer Felder**
 - ➔ Was für Daten soll das Feld aufnehmen (Text, Zahlen, Datum, ...)?
 - ➔ Eindeutige und selbsterklärende Feldnamen wählen, um als Programmierer effizient arbeiten zu können (s. S.14)
 - ➔ Felder unterstützen keine Aliase
 - ➔ Wo soll das Feld auf der Form positioniert sein?
 - ➔ Unterstützung des Benutzers bei der Bedienung der Anwendung (Aufgabenangemessenheit)
 - ➔ Lotus Notes berechnet die Felder auf einer Form von „links oben“ nach „rechts unten“
 - ➔ Wie soll dem Nutzer Sinn und Zweck des Feldes verdeutlicht werden?
 - ➔ Erläuterung der verlangten Eingabe (Help Description)
 - ➔ Beispieldaten (Field Hint)
 - ➔ Wie sollen die Daten dem Nutzer präsentiert werden?
 - ➔ Schriftart, -größe und -stile





Feldnamen müssen folgende Bedingungen erfüllen:

- Sie dürfen **nicht** mehr als 32 Zeichen umfassen
- Sie dürfen **nicht** mit einer Zahl beginnen
- Sie dürfen **nicht** das Symbol „@“ enthalten
- Sie dürfen **kein** Leerzeichen enthalten
- **Vorsicht bei der Benutzung von Dollarzeichen („\$“)**
 - Felder mit vorangestellten „\$“-Zeichen sind oft von Lotus Notes intern genutzte Felder



→ Zusätzlich sollten bei der Vergabe von Feldnamen die folgenden Konventionen eingehalten werden:

- Feldnamen sollten **englisch** sein
- Sie sollten **keine Unterstriche** enthalten
- Bei Zusammensetzungen aus mehreren Wörtern sollten diese **mit einem Großbuchstaben beginnen**
(z.B. DateHired, HomeAddress)
- Feldnamen müssen nur **innerhalb einer Form eindeutig** sein
- Sie sollten **keine Präfixe** enthalten
 - Falsch: „cusName“ oder „customerName“
 - Richtig: „Name“
- Betreffzeilen oder ähnliches heißen immer **“Subject“**
- Ein Richtextfeld, das in der Regel den Inhalt eines Dokumentes darstellt, heißt immer **“Body“**
- Ein Feld zur Kategorisierung heißt immer **“Categories“**



Feldtyp	Wert wird...
Editable (Bearbeitbar)	... durch den Benutzer eingegeben. Felder können beim Erstellen eines neuen Dokuments mit fest vorgegebenen oder dynamisch berechneten Standardwerten gefüllt werden, welche vom Benutzer geändert und gespeichert werden können.
Computed (Berechnet)	... durch eine Formel berechnet und im Dokument gespeichert. Die Berechnung des Feldes erfolgt automatisch beim Speichern und Aktualisieren (Refresh) des Dokuments.

Feldtyp	Wert wird...
Computed for display (Berechnet zum Anzeigen)	... durch eine Formel berechnet. Der Feldinhalt wird allerdings nicht mit dem Dokument gespeichert. Die Berechnung des Feldes erfolgt automatisch beim Anzeigen, Speichern und Aktualisieren (Refresh) des Dokuments
Computed when composed (Berechnet beim Anlegen)	... einmalig durch eine Formel beim Erstellen eines neuen Dokuments berechnet und im Dokument gespeichert. Der Wert ist nachträglich nicht veränderbar.



Datentyp	Felder, welche diesen Datentyp nutzen
Text	Text, Dialog list, Checkbox, Radio button, Listbox, Combobox, Authors, Names, Readers, Password, Time zone, Color
Number	Number
Time/Date	Date/Time
Formula	Formula
Rich Text	Rich Text, Rich Text Lite



- ➔ **Felder der Datentypen Text, Number, Time/Date und Formula weisen folgende Merkmale auf:**
 - ➔ Felder dieser Datentypen können in Views angezeigt werden
 - ➔ Felder können eine Liste von Werten des gleichen Datentyps enthalten
 - ➔ Speicherung der in einem Feld eingegebenen Daten erfolgt generell in Listen
 - ➔ Die Formatierung dieser Felder wird mittels des korrespondierenden Feldes auf der Form, welche zum Anzeigen der Daten genutzt wird, festgelegt

- ➔ **Felder des Datentyps „Rich Text“ weisen folgende Merkmale auf:**
 - ➔ Rich Text Felder speichern alle Arten von Daten, einschließlich Grafiken und Dateien
 - ➔ Die Formatierung der Textinhalte kann individuell vorgenommen werden
 - ➔ Rich Text Felder können in Views **nicht** angezeigt werden





→ Daten in „number fields“ können unter anderem zum Ausführen mathematischer Berechnungen verwendet werden.

→ Folgende Eingaben sind möglich

- Ziffern von 0 bis 9
- Plus- und Minuszeichen (+ u. -)
- Dezimaltrennzeichen (.)
- Wissenschaftliche Schreibweise (E)



Achtung!

→ Beim Speichern eines leeren „number fields“ wird ein Nullwert (Leerstring: „“) in diesem Feld gespeichert.

→ Das „number field“ wird von Notes zu diesem Zeitpunkt wie ein Feld des Datentyps Text behandelt!

→ Damit sichergestellt ist, dass der Datentyp Number im Feld gespeichert wird und es bei Berechnungen nicht zu Problemen kommt, sollte das Feld immer mit einem Vorgabewert (beispielsweise 0) belegt werden.



Feld	Wird verwendet, wenn ...
Dialog list (Dialogliste)	... der Benutzer aus einer Liste einen oder mehrere vorgegebene Einträge auswählen oder eigene Eingaben in das Feld tätigen soll. Die Auswahl der Einträge erfolgt über eine Dialogbox.
Checkbox (Kontrollkästchen)	... mehrere Objekte einer Liste ausgewählt werden sollen. Alle Auswahloptionen werden hierbei gleichzeitig auf der Form dargestellt.
Radio button (Optionsschaltfläche)	... ein Eintrag einer Liste ausgewählt werden soll. Alle Auswahloptionen werden hierbei gleichzeitig auf der Form dargestellt.

Feld	Wird verwendet, wenn ...
Listbox (Listenfeld)	... der Benutzer aus einer Liste einen oder mehrere vorgegebene Einträge auswählen soll. Die Auswahl der Einträge erfolgt direkt durch „scrollen“ und Auswählen der Einträge in der Liste auf der Form.
Combobox (Kombinationsfeld)	... der Benutzer aus einer Liste einen vorgegebenen Eintrag auswählen oder eigene Eingaben in das Feld tätigen soll. Die Auswahl der Einträge erfolgt über eine „Dropdown“-Menü auf der Form.

☞ Siehe auch Domino Designer Hilfe: Contents → Application Design → Designing fields → Creating fields to display lists of choices

➤ „Name fields“ sind spezielle Felder zum Eintragen von Benutzernamen

Feld	Verwendungszweck
Names	Den Namen eines Benutzers oder Servers einfügen oder anzeigen
Readers (Leser)	Benutzer identifizieren, die über die Berechtigung zum Lesen des Dokuments verfügen.
Authors (Autoren)	Benutzer identifizieren, die über die Berechtigung zum Bearbeiten des Dokuments verfügen. Im Authors Field aufgelistete Benutzer mit dem Zugriffslevel „Author“ in der ACL der Datenbank können diese Dokumente bearbeiten.

- Ermöglicht Kontrolle über das Anzeigen und Ausblenden eines Elements auf einer Form
- Hilfreich beim Erzeugen von Forms für den Gebrauch im Notes Client und im Web sowie zur Erleichterung der Bedienung von Anwendungen
- Kein Sicherheitsfeature, da der Zugriff auf Daten über Document Properties weiterhin möglich ist



- ➔ **Erstelle eine Form „Order“ mit den Alias „ord“**
- ➔ **Füge eine Überschrift hinzu und formatiere diese entsprechend**
- ➔ **In einer Tabelle sollen folgende Felder angeordnet werden:**
 - ➔ Number : „number“
 - ➔ Type : „type“ mit den Auswahlmöglichkeiten:
(Repair, Varnishing, Technical Inspection, oil change, tyre change) , weitere
Werte sollten nachträglich vom User hinzugefügt werden können
 - ➔ Comment : „comment“
 - ➔ Vehicle type: „vtype“ mit den Auswahlmöglichkeiten
 - ➔ (truck, van, passenger car)
 - ➔ Vehicle : „vehicle“
 - ➔ Estimated costs : „costs“
 - ➔ Owner : „owner“
 - ➔ Date: „date“ computed when composed ;@Now
 - ➔ Staffer : „staffer“ computed when composed; @Name([CN];@Username)
- ➔ **Erstelle fünf Dokumente mit dieser Form**



- ➔ **Begründe, warum Forms auch als „Templates für
Dokumente“ betrachtet werden können**

