



Workgroup Computing Praktikum

Introducing Formula Language

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>

Mit Formula Language arbeiten

- ➔ **Formula Language (Formelsprache) ermöglicht Anwendern mit geringen Programmierkenntnissen das Programmieren in Lotus Notes**
 - ➔ Hat eine sehr vereinfachte Syntax
 - ➔ Für Standardprobleme gibt es eine Vielzahl von einfachen Befehlen (@Functions, @Commands)

- ➔ **Die Formelsprache wird insbesondere verwendet zum**
 - ...
 - ➔ ... Berechnen und Vergleichen von Werten
 - ➔ ... Automatisieren von Aufgaben (z. B. Agenten)
 - ➔ ... Validieren von Eingabewerten (Input Validation Formula)

→ Formeln in Views

- Auswahl von Dokumenten für eine View, bei der keine der einfachen Auswahlbedingungen erfüllt wird
- Daten aus mehreren Feldern in einer Spalte anzeigen

→ Formeln in Forms

- Titel für neue Dokumente anzeigen und beim Bearbeiten der Dokumente einen anderen Titel anzeigen
- Enddatum aus Anfangsdatum und Dauer berechnen
- Erstelldatum berechnen
- Liste von Optionen erstellen
- Benutzereinträge konsistent formatieren (Input Translation)
- Input Validation, Default Values

→ Der Code wird direkt in den Designelementen (Forms, Views, Agenten, Actions, ...) und Feldern gespeichert

→ Ermöglicht Codeausführung beim Eintreten bestimmter Ereignisse (Events), z.B. ...

- ... beim Öffnen einer Maske
- ... beim Öffnen (z. B. Default Value), beim Bearbeiten (z.B. Input Translation) und beim Speichern eines Dokuments (z.B. Input Validation)
- ... beim Betätigen eines Buttons oder einer Action

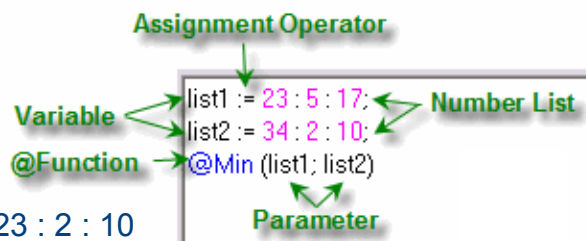
Element	Verwendungszweck
Konstanten	Konstanten sind statische Werte → Text , z. B. "Hallo Welt" oder "\"Pacific\"" → Zahlen , z. B. 10 oder -6.37 oder 37E6 → Datum / Uhrzeit , z. B. [17:30] oder [11.7] oder [18.10.71 6:30]
Variablen	„Platzhalter“ für Felder oder Werte
Operatoren	Verknüpfen von Werten (z. B.: !, , &, :, >, =)
Funktionen	→ „@functions“ für viele Standardaufgaben vorhanden: → Informationen über die Anwendung zurückgeben → Werte verändern → Neue Werte berechnen
Schlüsselwörter (Keywords)	Reservierte Wörter, die spezielle Funktionen unterstützen (z. B.: SELECT, FIELD, REM, DEFAULT)

→ **@Functions und deren Parameter werden mit einem Semikolon voneinander getrennt**

→ **Achtung:**

Einzelne @Functions nutzen Listen als Parameter, das Trennzeichen der einzelnen Listenelemente ist ein Doppelpunkt

→ **Beispiel:**



Rückgabe: 23 : 2 : 10

- ➔ **Groß- / Kleinschreibung ist bei den Programmier-
elementen der Formelsprache nicht relevant**
 - ➔ Konventionen: Keywords werden immer in Großbuchstaben geschrieben, @Functions und @Commands in gemischter Schreibweise

- ➔ **Formelcode wird nicht automatisch eingerückt**
➔ **Handarbeit**

- ➔ **Kommentare erleichtern das Verständnis des Codes**
 - ➔ Kommentare müssen mit einem Semikolon abgeschlossen werden
 - ➔ Syntax: REM {Kommentar ohne Anführungszeichen};

```
Run Client Formula
REM{Euer Code sollte immer gut dokumentiert sein};
@If(Status="Dokumentiert";
    "Gut gemacht, der Code ist dokumentiert";
    @Failure("Bitte kommentiere deinen Code")
);
REM{Zum kommentieren nutzt das REM};
REM{Euer Code sollte außerdem sinnvoll eingerückt sein, damit Euer Tutor diesen besser verstehen kann};
@If(@IsMember("Karl";namelist);
    @Prompt([OK];"Ereignis";"Karl ist steht in der Namensliste");
    @Prompt([OK];"Ereignis";"Leider steht Karl nicht in der Liste")
);
REM{Kommentare helfen, Code zu verstehen. Das Kommentieren ist Pflicht};
```

⇒ **Operatoren ermöglichen Berechnungen und Verknüpfungen**

- ⇒ Einfache arithmetische Operationen bei Zahlenwerten: +, -, /, *
- ⇒ Verknüpfen von Textstrings (Konkatenation):
"abc" + "def" → "abcdef"
- ⇒ Erzeugen von Listen mittels Doppelpunkt: "abc"."def"."ghi"
- ⇒ Vergleichsoperatoren: <, <=, <> und >< („ungleich“), >=, >, =
- ⇒ Logische Operatoren: & („und“), | („oder“)
- ⇒ Negation: ! („nicht“)

⇒ **Variablen müssen in der Formelsprache nicht explizit deklariert werden**

- ⇒ Zuweisungsoperator „:=“
- ⇒ Beispiel: `x := @Power(a;2) + @Power(b;2);`

**Vorname: Susanne
Nachname: Fröhlich**

⇒ **Vorname:=“Susanne“;**

⇒ **Nachname:=“Fröhlich“;**

⇒ **Vorname + Nachname**

⇒ SusanneFröhlich

⇒ **Nachname + “, “ + Vorname**

⇒ Fröhlich, Susanne

⇒ **(Vorname != “Harald“) & (Nachname = “Fröhlich“)**

⇒ wahr

➔ @Functions zum Abfragen von hilfreichen Informationen

- ➔ @UserName, @Created, @IsNewDoc, @ViewTitle, @DbName, @ClientType, @BrowserInfo

➔ @Functions zum Ändern von (Feld)Daten

- ➔ @Adjust([5.12.2001];0;-2;5;0;0;0)
- ➔ @Text(10), @ProperCase ("HALLO WeLt"), @Trim (" hallo ")

➔ @Functions zum Extrahieren von Bestandteilen eines Strings

- ➔ @Left; @LeftBack, @Middle, @MiddleBack, @Right; @RightBack, @Begins, @Contains, @Ends

➔ Bedingte Ausführung von Anweisungen mittels der @If-Funktion

- ➔ @If(Bedingung1; WahrAktion1; Bedingung2; WahrAktion2; ... ; FalschAktion)

➔ Wiederholtes Ausführen von Anweisungen (Schleifen)

- ➔ @While(Ausführungsbedingung ; Anweisung1; Anweisung2 ; ...)
- ➔ @DoWhile(Anweisung1; Anweisung2 ; ... ; AnweisungN ; Ausführungsbedingung)
- ➔ @For(Initialisierung; Ausführungsbedingung; Inkrementierung; Anweisung1 ; ... ; AnweisungN)

➔ Einfaches Testen und Debuggen von @Functions:

<p>To _____</p> <p>cc _____</p> <p>bcc _____</p> <p>Subject vorname="Karl"; nachname="Karg"; vollname=vorname+" "+nachname+" "; @trim(" Name: "+vollname);</p>	-... Shift-F9 ...->	<p>cc _____</p> <p>bcc _____</p> <p>Subject Name: Karl Karg</p>
--	---------------------	---

Groupware
Competence Center
GCC

Domino Designer Hilfe

Lotus Domino Designer 8 Help @LeftBack - IBM Lotus Notes

File Edit View Actions

Contents Index Search Tips

17 topics matched. Indexed ? x

@leftback Search Clear

Examples: @LeftBack

- @LeftBack
- Locating and extracting substrings
- @Ends
- Examples: @DbName
- @Begins
- @Contains
- Where does this @function work? (Part 2 L - R)
- @RightBack
- @Left
- Where does this @function work? (Part 1 L - R)
- Examples: Accessing the user environment
- StrLeftBack function
- @Right
- @Functions
- @MiddleBack
- @Middle

FORMULA LANGUAGE

@LeftBack

Example
Searches a string from right to left and returns a substring.

Syntax
@LeftBack(*stringToSearch* ; *numToSkip*) or
@LeftBack(*stringToSearch* ; *startString*)

Parameters

stringToSearch
Text. The string where you want to find the leftmost characters.

numToSkip
Number. Counting from right to left, the number of characters to skip. All the characters to the left of that number of characters are returned. If the number is negative, the entire string is returned.

startString
Text. A substring of *stringToSearch*. All the characters to the left of *startString* are returned.

Return value
resultString
Text. The leftmost characters in *stringToSearch*. The number of characters returned is determined by either *numToSkip* or *startString*.

Language cross-reference

University of Paderborn
Dept. Business Information Systems
Prof. Dr. Ludwig Nastansky

13

Groupware
Competence Center
GCC

Übung: Formel interpretieren

Vorname: Susanne
Nachname: Fröhlich
Erstellt am 5.5.2006 um 23:59:37

- ➔ **VollstName := Vorname + " " + Nachname;**
"Willkommen " + VollstName
➔ Willkommen Susanne Fröhlich
- ➔ **Vorname = "Susanne"**
➔ wahr
- ➔ **@Adjust(@Created; 0; 6; 0; -19; 0; 0)**
➔ 5.11.2006 04:59:37
- ➔ **@If (@Created < @Today; "Alt"; "Neu")**
➔ Alt
- ➔ **@Text (10,22)**
➔ 10,22

University of Paderborn
Dept. Business Information Systems
Prof. Dr. Ludwig Nastansky

14

Klaus will eine Validation Formula schreiben, die er in jedes Feld übernehmen kann. Welches der folgenden Statements sollte er nutzen?

- A. `@If(@GetField = ""; @Failure("Please enter a value for this field");@Success)`
- B. `@If(@ThisName = "";@Failure("Please enter a value for this field");@Success)`
- C. `@If(@Command([EditGotoField]) = ""; @Failure ("Please enter a value for this field"); @Success)`
- D. `@If(@ThisValue = "";@Failure("Please enter a value for this field");@Success)`