

---

# Performanceanalyse für LotusScript

Jens-B. Augustiny  
LIGONET GmbH

## Wer sind wir?

---

- 
- A vertical stack of several dominoes, arranged in a slightly curved line, positioned on the left side of the slide.
- ◆ Business Partner mit Sitz in der Schweiz
  - ◆ International tätig
  - ◆ Seit mehr als 10 Jahren Notes/Domino
  - ◆ Referent, unter anderem Lotusphere 03/05
  
  - ◆ Verfügbar für Ihre Projekte und Events

# Agenda

---



- ◆ Performanceprobleme: Was ist das?
- ◆ Wann macht Performancetuning Sinn?
- ◆ Einfaches Beispiel
- ◆ Einfache Performanceanalyse mit LotusScript Möglichkeiten
- ◆ Präzises Timing
- ◆ Profiling mit Domino 7
- ◆ Granular mit Teamstudio Profiler
- ◆ Zusammenfassung
- ◆ Fragen

# Schlechte Performance?

---

- 
- ◆ Dieses Programm ist schlecht ....
  - ◆ Lieferfristen statt Antwortzeiten ...
  - ◆ Damit kann man unmöglich arbeiten.
  - ◆ Dieser Ablauf muss unbedingt verbessert werden .....

Wer kennt sie nicht – diese Aussagen ?



# Welche Performance?

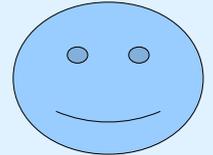
---

- 
- ◆ „Empfundene Performance“
    - ◆ Der User hat den Eindruck, das Programm erbringt nicht die gewünschte Leistung.
    - ◆ Schwer definierbar, daher auch schwer analysierbar
    - ◆ Mögliche Abhilfen:
      - ◆ Aufklärung über die Funktionsweise
      - ◆ Mehr Hinweise am Bildschirm
      - ◆ Aendern der Prozessreihenfolge

# Tatsächliche Performance

---

..... oder eher der Mangel derselben ...



- ◆ Durch Timing festgestellte Flaschenhäuse:
  - ◆ Praktisch nicht vorhersehbar
  - ◆ Differenz zwischen Testinstallation und Produktionsinstallation
  - ◆ Jede Aussage ohne Messung ist Spekulation
  - ◆ Die Messung selbst kann das Ergebnis verfälschen



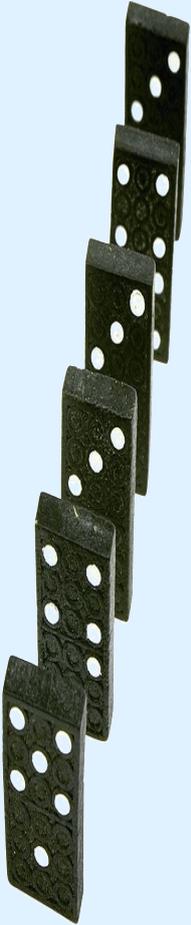
## Beispiel

---

- 
- A vertical stack of several dominoes, showing different faces with dots, positioned on the left side of the slide.
- ◆ 4 Varianten, die dasselbe tun: Die Views aus dem Names auflisten
  - ◆ 2 Parameter werden verändert: Die Wahl des Loops und die Art, wie auf die Eigenschaften der NotesDB zugegriffen wird.
  - ◆ Welcher Code ist der schnellste und warum?

## Folgerungen:

---

- 
- ◆ Resultate können überraschen
  - ◆ Messungen müssen in möglichst realistischer Umgebung erfolgen
  - ◆ Datenmengen und Zusammensetzung müssen realistisch sein, die üblichen Testumgebungen sind meist ungenügend
  - ◆ Vorgehen bei der Analyse muss streng systematisch sein (zB. Einkreisen der Problemstellen)

# Timing mit Now

---

BaseTime = Now

.... der zu testende Code hier

Print "Time used: " Now – BaseTime



◆ Vorteil:

◆ Einfach, günstig

◆ Nachteil:

◆ Verändert den Code

◆ Schlecht platziert wird das Resultat verfälscht (Loops)

◆ ungenügende zeitliche Auflösung

# Hochauflösender Timer I

---

- 
- A vertical stack of several dominoes, showing their faces with dots, positioned on the left side of the slide.
- ◆ Von Bruce Perry wurde auf SearchDomino eine nicht dokumentierte Technik veröffentlicht, welche hochauflösendes Timing erlaubt
  - ◆ [http://searchdomino.techtarget.com/tip/1,289483,sid4\\_gci895240,00.html?FromTaxonomy=%2Fpr%2F283841](http://searchdomino.techtarget.com/tip/1,289483,sid4_gci895240,00.html?FromTaxonomy=%2Fpr%2F283841)

# Hochauflösender Timer II



```
stc = Getthreadinfo(6) ' get the starting tick  
count
```

```
<the code you want to time>
```

```
ftc = Getthreadinfo(6) 'get the final tick count  
tps = Getthreadinfo(7) 'get the ticks per second
```

```
t = (ftc - stc) / tps 'final tick count minus  
starting tick count divided by ticks per second  
yields duration.
```

# Resultatausgabe

---

- 
- A vertical stack of several dominoes, arranged in a slightly curved line, positioned on the left side of the slide.
- ◆ Timing-Angaben mit Print (wie Beispiel) in die Statuszeile oder (serverseitig) ins Log
  - ◆ Günstiger die Ausgabe in ein (Profile-) Dokument oder in eine eigene Analysedatenbank

## Profiling in Domino 7

---



- ◆ Server basiert
- ◆ Nur für Agenten
- ◆ Java und LotusScript Agenten und WebServices
- ◆ Aktivieren in den Properties des Agenten
- ◆ Rechtsklick auf Agent gibt Zugriff auf Resultate
- ◆ Nur Notes-Objekte werden aufgeführt und mit Timing-Informationen versehen

## TS Profiler

---



- ◆ Timing von Funktionen
- ◆ Timing auf Zeilenebene möglich
- ◆ Zusätzliche Analysewerte verfügbar
- ◆ Server wie Clientseitig einsetzbar
- ◆ Nur LotusScript
- ◆ Starke Verlangsamung der Applikation
- ◆ Zusatzkosten

# Faustregeln I

---

## Häufig auftretende Performance-Probleme

- ◆ Variablen: Variant-Variablen sind deutlich langsamer als spezifisch deklarierte Variablen
- ◆ Gilt auch für String-Variablen gegenüber Variant
- ◆ Durchlaufen einer Document Collection: DCol.GetNthDocument(n) in einem For-Next Loop ist deutlich langsamer als DCol.GetFirstDocument und anschliessend DCol.GetNextDocument(Doc) in einem Do While Loop.

## Faustregeln II

---

- 
- ◆ Nested IFs können effizienter sein als kombinierte Bedingungen:

    If  $a > b$  then

        if  $c < d$  then ...

    besser als:

    If  $a > b$  AND  $c < d$  then

- ◆ Loops:
  - ◆ Forall Loops durch Arrays sind deutlich schneller als ein For – Next
  - ◆ For – Next ist schneller als Do ..... Loop/While...Wend

## Faustregeln III

---

- 
- ◆ Zugriff auf eine Dokumentenauswahl über View ist schneller als das Verwenden einer Document Collection
  - ◆ Zugriff auf Notes Datenobjekte über eine temporäre Variable ist in der Regel schneller als ein Direktzugriff auf das Objekt

# Zusammenfassung

---

- 
- ◆ Es gibt technische und nicht technische Gründe für Performance-Probleme
  - ◆ Bei technischen Problemen ist jede Aussage ohne genaue Messung spekulativ
  - ◆ Kein Fall ist gleich wie der andere
  - ◆ Systematische "Top-down" Analyse wichtig
  - ◆ Ein völliges Redesign einer Applikation kann angesagt sein
  - ◆ Es gibt ein paar Best Practices

