

# Effektives Trigger Management

- Trigger kontrollieren
- Trigger protokollieren
- Einfacher, einheitlicher Ablauf für alle Trigger
- Flexible Lösung
  - Konfigurierbar
  - Erweiterbar
  - Vielseitig

# Trigger kontrollieren

- Trigger aktivieren bzw. deaktivieren (einzelnd oder in Gruppen)
  - Globale Variable \$\$trigger.disabled
  - CF trigger.disable( triggerlist )
  - CF trigger.enable( triggerlist )
  - CF trigger.setState( triggerlist )

# Trigger kontrollieren

- Trigger-Status überprüfen  
(einzelnd oder in Gruppen)
  - CF trigger.isDisabled( triggerlist )
  - CF trigger.isEnabled( triggerlist )
  - CF trigger.getState( triggerlist )

# Trigger Script Template

If [[trigger.enter](#)]

**# Disable trigger**

Set Variable [\$trigger.state; Value: trigger.disable("All")]

If [[param.script.check](#)]

    If [condition 1]

        ...

    Else If [condition 2]

        ...

    End If

End If

**# Reset trigger state**

Set Variable [\$\_; Value: trigger.setState( \$trigger.state )]

End If

Exit Script [Result: [trigger.exit](#)]

# CF trigger.enter

- Prüfen, ob Trigger aktiviert ist
  - Trigger Name ergibt sich aus Script Name
- Trigger-abhängige Variablen setzen
  - \$trigger.proceed
  - \$\$trigger.level
- Umgebungsvariablen setzen
  - \$tr.layout, \$tr.mode, \$tr.content, \$tr.field, \$tr.key, ...
- Logbucheintrag

# CF trigger.exit

- Trigger Level zurücksetzen
- Logbucheintrag abschließen
- Script Rückgabewert bereitstellen

# Trigger Logbuch

- **Unaufdringlich integriert**
  - Keine Tabellenfelder
  - Gesteuert mit globalen Variablen (`$$triggerlog`, `$$triggerlog.on`)
  - Keine Datensammlung, wenn nicht aktiviert
- **Steuer- und konfigurierbar**
  - Deaktivierte Trigger nicht anzeigen (`$$triggerLog.enabledOnly`)
  - Einzelne Trigger nicht protokollieren (`$trigger.silent`)
  - Auswahl der angezeigten Information (`CF trigger.option`)

# Trigger Logbuch

- Trigger-Aktivitäten protokollieren
  - Wann feuert welcher Trigger?
  - Ist der Trigger zur Zeit aktiv?
  - In welcher Umgebung feuert der Trigger? (Feld, Objekt, Layout, ...)
  - Wird das auslösende Ereignis fortgesetzt?
- Trigger-Aktivitäten debuggen
  - Eigene Meldungen im Logbuch anzeigen
  - Bevor oder während der Trigger ausgeführt wird



# Trigger Logbuch Beispiel

**OnObjectEnter**

Script: OnObjectEnter  
Layout: Trigger Beispiel  
Object: txtTelefon  
Field: Daten::Telefon

**OnObjectExit**

Script: OnObjectExit  
Layout: Trigger Beispiel  
Object: txtTelefon  
Field: Daten::Telefon  
Event proceeded

**OnObjectEnter**

Script: OnObjectEnter  
Layout: Trigger Beispiel  
Object: txtNachname  
Field: Daten::Nachname  
Das Feld 'Nachname' wurde zum 5. Mal betreten.

**OnObjectExit**

Script: OnObjectExit  
Layout: Trigger Beispiel  
Object: txtNachname  
Field: Daten::Nachname  
Event canceled

**OnObjectModify**

Script: OnObjectModify  
Layout: Trigger Beispiel  
Object: txtNachname  
Field: Daten::Nachname  
Content.short: "Kegebein"  
FrontTabs: tabKontakt

- Triggername
  - OnObjectEnter
  - OnObjectKeystroke disabled
- Umgebungsvariablen
  - Layout: Trigger Beispiel
  - Field: Daten::Telefon
- Ereignisfortführung
  - Event proceeded
  - Event canceled
- Debug-Meldung
  - Das Feld 'Nachname' wurde ...

# **Arnold Kegebein**

FileMaker Blog "Gallimaufry"  
<http://blog.kegebein.net>