

# Session Track 3

## Analyse-Tool "Reveal"

---

# Entstehung des Tools

- eigenes Hobby-Projekt vor 12 Jahren: LotusScript-Scanner / Parser
- Kunden-Anforderung: Hinweise auf Daten-Sabotage im SourceCode suchen (Remove() etc.) -> zu viele False-Positives bei Suche im Domino Designer
- Aktuelle Projekt-Anforderung: Querverweise suchen für Notes-Migration (App-Clustering)
- Weiterentwicklung in den letzten 6 Jahren: "Nachtarbeit" nach Kundenprojekten, Familie und Haushalt

# Anwendungs-Szenarien

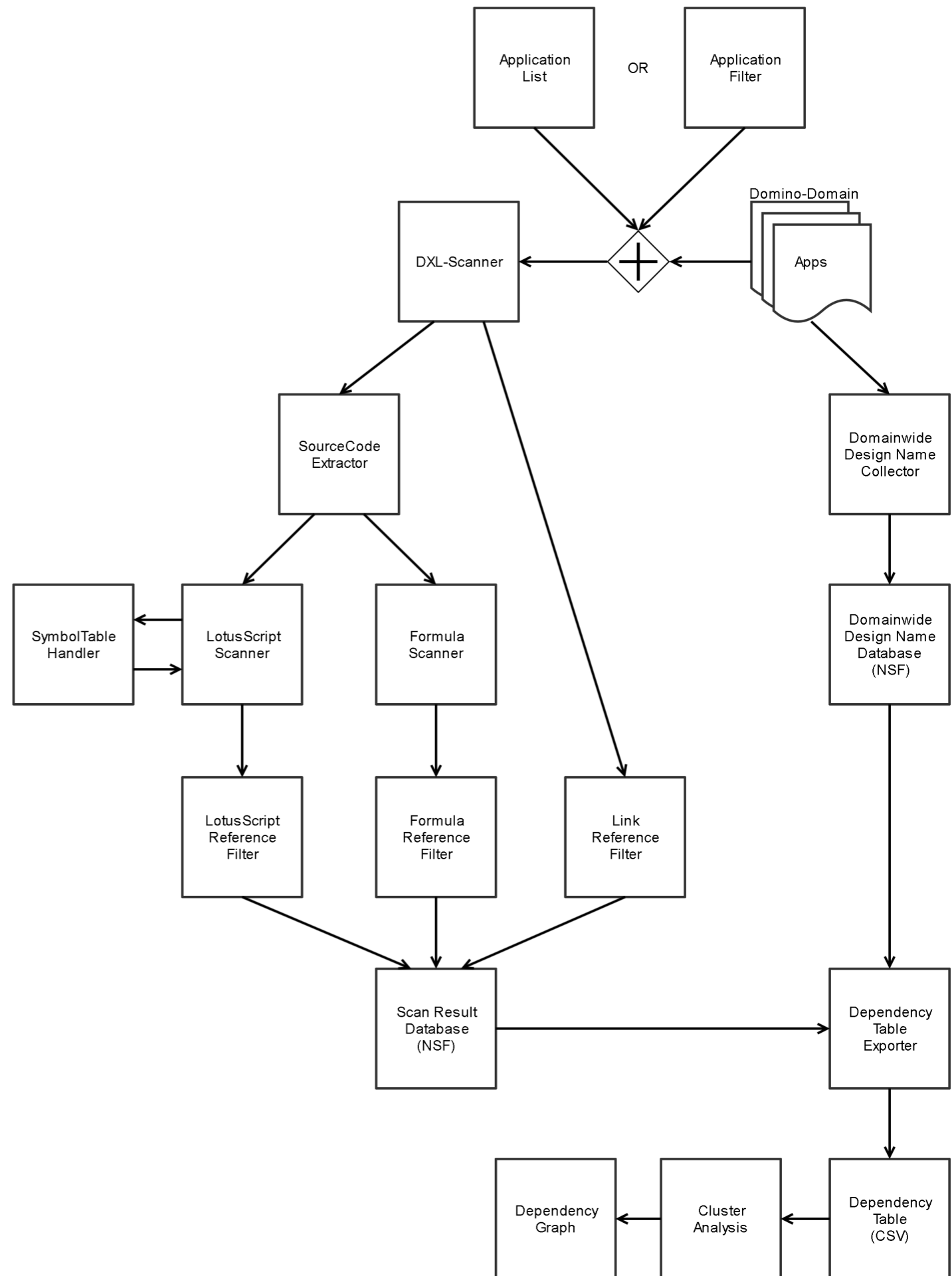
- Für Redesign oder Ausserbetriebnahme einer Anwendung muss bekannt sein, welche Designelemente darin von ausserhalb referenziert werden. In den meisten Domino-Umgebungen gibt es keine Dokumentation dieser externen Referenzen. Diese sind häufig im SourceCode enthalten und ohne manuelle Suche nicht auffindbar.
- Viele Notes-Anwendungen erzeugen durch unbenutzte Views erhebliche Speicher- und Serverlast und letztendlich lange Antwortzeiten. Für eine View-Bereinigung muss jedoch bekannt sein, welche Views intern und/oder extern referenziert werden.
- Viele Umgebungen enthalten Anbindungen an SQL-Datenbanken per LSX-LC, es gibt aber keine Dokumentation in welchen Anwendungen an welcher Stelle diese implementiert sind.

# Tool Reveal.Reference

Durchsucht eine Domino-Domäne nach Designelement-Referenzen in SourceCode und Design

- findet Referenzen sowohl im SourceCode als auch in Direkt-Verweisen (z.B. in Gliederungen)
- legt Analyse-Resultate in eine Notes-Datenbank ab
- vermeidet False-Positives (in Kommentaren, Masken-Beschriftung, View-Columns,..) durch syntaktisch Analyse
- identifiziert externe Referenzen (die nicht innerhalb derselben NSF liegen)
- kann externen Referenzen die richtige Ziel-Datenbank zuordnen  
identifiziert externe Referenzen die nicht aufgelöst werden können

# Programmstruktur



# Reveal.Reference - Entwicklungs-Stand

Bei 2 Kunden hatte ich die Gelegenheit Reveal.Reference testweise laufen zu lassen:

- es wurden insgesamt ca. eine Million unterschiedlicher Designelemente gescannt
- dabei wurden knapp 2 Gigabyte Sourcecode untersucht
- durchschnittliche Laufzeit: ca. 2 Minuten pro Datenbank
- Das Tool ist produktreif, es fehlt noch am UI und an der Options-Steuerung (Einstellungen zum Scan-Lauf verändern)

# Live-Demo von Reveal.Reference

Part Event	Sourcecode
▼ K:/Notesdat/bookmark.nsf	
▼ Agent	
▼ (ng-SetLocation)	
▼ Global	
▼ Initialize	Sub Initialize Dim s As New notessession Dim uiw As New notesuiworkspace Dim sServer As String curloc = s.getenvironmentstring("Loc View: Lcase("LocATionS") - (GetView) -> addrBook
▼ Form	
▼ LayoutEditorDlg	
▼ button.Select	
▼ Setgroupcalendar	Sub SetGroupCalendar(uiwork, thisframe, maildb) Set note = uiwork.currentdocument.document Set groupCalendarView = maildb.getvie View: "GroupCalendarFolder" - (GetView) -> maildb View: "GroupCalendarFolder" - (GetView) -> maildb View: "GroupCalendarFolder" - (GetView) -> maildb
▼ Global	
▼ Getmailver	Function getMailVer(mailDb As notesdatabase, tempprofile As notesdocument) Set FollowUpView = maildb.getview("MiniView - Followup) View: "MiniView - Followup2" - (GetView) -> maildb View: "MiniView - Notices2" - (GetView) -> maildb View: "ToDoByStatus" - (GetView) -> maildb
▼ Queryopen	Sub Queryopen(Source As Notesuidocument, Mode As Integer, Isnewdoc As Variant, Continue As Variant) Dim s As New notessession Se View: "People" - (GetView) ->
▼ ng-today_IconsC	
▼ NamedElementLink	
▼ Value	view: "Calendar" - (NamedElementLink) -> mailfile
▼ ng-today_IconsT	
▼ NamedElementLink	
▼ Value	view: "Tasks" - (NamedElementLink) -> mailfile view: "(\$Follow-Up)" - (NamedElementLink) -> mailfile
▼ ppage layout C	
▼ field.CalMailType	
▼ Defaultvalue	MailFilename := @Subset(@MailDbName;-1); MailServer := @Subset(@MailDbName; 1); view := "Calendar"; Lookup := @If(MailServer = "" View: "Calendar" - (DbColumn) -> @Subset(@MailDbName; 1):@Subset(@MailDbName;-1)
▼ field.ToDoMailType	
▼ Defaultvalue	MailFilename := @Subset(@MailDbName;-1); MailServer := @Subset(@MailDbName; 1); view := "Tasks"; Lookup := @If(MailServer = ""; @ View: "Tasks" - (DbColumn) -> @Subset(@MailDbName; 1):@Subset(@MailDbName;-1)
▼ NamedElementLink	
▼ Value	form: "Appointment" - (NamedElementLink) -> mailfile form: "Person" - (NamedElementLink) -> addressbook form: "Task" - (NamedElementLink) -> mailfile

# Reveal-Tool: Trouble

- DXL nach SourceCode scannen: überraschende Zeilenumbrüche
- Formelsprache scannen: z.B. "}" ist eine Formel !
- LotusScript scannen: Symboltabelle beim Scannen mitführen
- LotusScript scannen: viele Operatoren sind URLEncoded !



# weitere Reveal-Tools

weitere Tools der Reveal-Familie sind in Planung:

- Reveal.Test = automatische Generierung von UnitTest Libs zu bestehenden LotusScript-Libs
- Reveal.Quality = Codierungs-Schwachpunkte werden gefunden und angezeigt
- weitere Ideen ?

# Produkt-Verfügbarkeit

- Entscheidung in 2016: ich werde keinen Eigen-Vertrieb des Tools betreiben
- momentan biete ich Projekt-Mitarbeit an mit den Reveal.Tools "im Rucksack"
- ich suche weiterhin Lizenznehmer für Weiterentwicklung / Vertrieb / Verkauf der Reveal.Tools

# Abschluss: Fragen und Antworten

meine Fragen:

- gibt es Bedarf für dieses Tool?
- was sollte das Tool evtl. noch können?
- wer könnte Reveal vermarkten?

[gerd@gehil.de](mailto:gerd@gehil.de)

<http://www.reveal.tools>