



## APL + Link

APL+Win-Code in Textfiles

Alexander Krämer, Dittrich & Partner Consulting GmbH

08.11.2024, Solingen



## Inhaltsübersicht

1. Vorbild & Idee
2. Userkommandos in APL+Win
3. UCMDLINK
  1. Kommando LINKIN
  2. Kommando LINKOUT
4. Fazit: Was geht? Was geht nicht?



## Vorbild & Idee

# Die Problemstellung

- APL-Code traditionell in Workspaces (=Binärdatei)
- aber: Textformat wünschenswert für verschiedene Zwecke
  - Versionskontrolle (Git, SVN, ...)
  - Codeanalyse mit APL-externen Tools
  - Vergleich von ähnlichen Funktionen aus verschiedenen Workspaces
  - ...

# Link – Versionierungstool aus Dyalog APL

- In Dyalog ermöglichen die Tools SALT und Link den Export (und Import) von Code als Textdateien
- mit Link insbesondere nicht nur Laden & Speichern, sondern auch fortlaufende Synchronisation
- beide Tools sind als Userkommandos verfügbar

```
]Link -??
```

```
LINK          User commands for namespace-directory synchronisation (see https://dyalog.github.io/link):
Add           Associate items in linked namespaces with new files/directories in corresponding directory,
              optionally with simultaneous definition
Break        Break link between namespace and corresponding directory
Configure    Set directory or user configuration options
Create       Link a namespace with a directory (create one but not both if non-existent)
Export       Export a namespace to a directory (create the directory if absent); does not create a link
Expunge      Erase item and associated file
GetFileName  Return name of file associated with item
GetItemName  Return name of item associated with file
Import       Import a namespace from a directory (create the namespace if absent); does not create a link
Refresh      Manually synchronise namespace or directory contents
Resync       Automatically synchronise namespace-directory differences
Status       List active namespace-directory links
Stop         Set, clear or report on breakpoints
Trace       Set, clear or report on lines traced
```

# Der Lösungsansatz in APL+Win

- APL kann bereits:
  - Funktionscode in Textarrays umwandeln,
  - Textarrays auf diverse Art formatieren und kodieren,
  - Textdateien schreiben und lesen
- Idee: vereine all diese Schritte in Userkommandos



# Userkommandos in APL+Win

# Allgemeines

- alle folgenden Aussagen gelten für APL+Win Version 19
- Speicherung in Komponenten-/SF-Dateien (sog. Kommandodateien)
- Verwaltung durch Userkommandos: ]umake, ]usave, ]uset, ]ufile u. a.
- vollständige Dokumentation im Benutzerhandbuch im Installationsordner

 [UserManual.doc](#)

# Der Kommandoprozessor

- grundsätzlich entsprechen Userkommandos regulären Funktionen, aber die Aufrufumgebung hat Besonderheiten:
  - semiglobale Objekte aus der Kommandodatei (teils nur für das ausgeführte Kommando, teils für alle)
  - bestimmte Systemvariablen (z. B. `□ELX` oder `□IO`) automatisch lokalisiert
  - automatisches Schließen aller mit `□NTIE` (o. ä.) geöffneten Dateien am Ende
- bestimmtes Verhalten kann beeinflusst werden, z. B. mit `]uset`

## „Bestandteile“ eines Userkommandos

- „Hauptfunktion“ mit Präfix CMD
- Gruppenvariable: Liste aller zu ladenden Objekte; mit Präfix GRP
- Dokumentation mit Suffix  $\Delta$ DOC
- z. B. ist für ]Bsp:
  - CMD Bsp die aufgerufene Funktion,
  - GRPCMD Bsp die Auflistung aller dazugehörigen Objekte
  - und Bsp $\Delta$ DOC der von der APL-internen Hilfe angezeigte String

## Standard-Utilityfunktionen und -variablen

- stehen in Userkommandos zur Verfügung, ohne in die GRP-Variable aufgenommen werden zu müssen
- ΔOUT: schreibe in die Session (kann mit /QUIET unterdrückt werden)
- ΔGETOPT: liest Parameter aus dem Aufruf
- ΔGLOBALIO (etc.): Wert von IO (etc.) außerhalb des Kommandoprocessors (nur Lesen)
- ΔRESULT: Rückgabewert von UCMD
- uvm.



# UCMDLINK

# Allgemeines

- ]LINKIN und ]LINKOUT, gepackt als eigene Kommandodatei UCMDLINK.sf (Einbinden mit ]ufile mit Administratorrechten)
- zusätzliche Dokumentation in  [APLW\\_LINK\\_Doku.docx](#)
- wesentliche Arbeitsschritte/Bestandteile:
  - Objekte identifizieren (0◦□NL, 0◦□NC)
  - Funktionscode (0◦□CR, □FX) und Arrays (0◦□VGET,  $\Phi$ ) verarbeiten
  - Speichern: Funktionen in afn-Dateien, Variablen in Vars.sf
  - Errorhandling (inkl. benutzerdefinierte Fehler)
  - Logging

## Kommando LINKIN

- Import von Funktionen und Variablen
- nur in leerem Workspace
- funktioniert nicht bei Namenskonflikten zwischen Importelementen und lokalen Objekten
- Bestandteile:

```
lunames GRPCMDLINKIN /G /F=D:\APL+Win\APLWIN_Link\UCMDLINK

AGroup:  GRPCMDLINKIN
CMDLINKIN▽      ucmdFileExist▽  ucmdLink_Fns▽   ucmdLink_Vars▽  ucmdΔTEXTREPL▽  ucmdΔRPL▽

AObjects not in any group:
LINKINΔDOC←     LINKOUTΔDOC←
```

## Kommando LINKIN - Syntax

`]LINKIN Dir /F=Fns /V=Vars /E=Ext`

- Dir: zu ladendes Verzeichnis
- Fns: Liste zu importierender Funktionen – mit Kommata getrennt
- Vars: Liste zu ladender Variablen (inkl. Systemvariablen) – mit Kommata getrennt
- Ext: Dateiendung für Funktionscode-Textdateien (Default: afn)

## Kommando LINKOUT

- Export von Funktionen und Variablen
- funktioniert auch bei Namenskonflikten zwischen Exportelementen und lokalen Objekten
- mit LOCK gesperrte Funktionen werden ignoriert
- Bestandteile:

```
Junames GRPCMDLINKOUT /G /F=D:\APL+Win\APLWIN_Link\UCMDLINK
```

```
  #Group:  GRPCMDLINKOUT  
CMDLINKOUT▽   ucmdLink_Fns▽   ucmdLink_Vars▽   ucmdFileExist▽   ucmdΔTEXTREPL▽   ucmdΔRPL▽
```

```
  #Objects not in any group:  
LINKINΔDOC←   LINKOUTΔDOC←
```

## Kommando LINKOUT - Syntax

`]LINKOUT Dir /O=Obj /E=Ext`

- Dir: Ablageverzeichnis
- Obj: Liste zu exportierender Objekte (Variablen und Funktionen, inkl. Systemvariablen) – mit Kommata getrennt
- Ext: Dateiendung für Funktionscode-Textdateien (Default: afn)

## Fazit: Was geht? Was geht nicht?

- Was geht?
  - Im- und Export von (Funktions-)Code wie mit JLINK.Export/Import
  - Logging (Fehlermeldungen & Fortschritts-/Ergebnismeldungen)
  - Variablen (inkl. einiger Systemvariablen) in Komponentendatei
- Was geht nicht?
  - textbasierte Versionierung von Variablen
  - Synchronisation zwischen Workspace und Speicherverzeichnis
  - MOM-Objekte ex-/importieren
  - Lesen von Objekten u. U. mit Problemen verbunden



**Vielen für Ihre Aufmerksamkeit!**