

APL - Journal

A Programming Language

2/2001

APL-Germany e.V.

In dieser Ausgabe:

Dr. Tilman Otto
APL als Client & Server für SAP R2/R3

Herbert Voß
Die mathematischen Funktionen
von Postscript

Reiner Nussbaum
Linux und APL

Klaus-Peter Friedrich
A.ID.A - Das Analysesystem der RSGV-
Informationsdienste in APL2

Frank Diehl
APL und MQSeries

Martin Hintz
Verarbeitung von Großrechnerdaten
im PC-Umfeld

Maren Röttgers, Nils Dennstedt
Die Hamburg-Mannheimer präsentiert
ihre SRBB

Sergy Alpin
Java-APL+Win Interface



Im Blickpunkt

David Liebttag/Ursula Holzer
APL2 Mainframe Features

Nr. 2 2001

Jahrgang 20

ISSN - 1438-4531

RHOMBOS-VERLAG

Inhaltsverzeichnis

Dr. Tilman Otto	
APL als Client & Server für SAP R2/R3	6
Herbert Voß	
Die mathematischen Funktionen von Postscript	13
Reiner Nussbaum	
Linux und APL APL kann für systemnahe Aufgaben unter LINUX verwendet werden	16
Klaus-Peter Friedrich	
A.ID.A Das Analysesystem der RSGV- Informationsdienste in APL2	19
David Liebttag/Ursula Holzer	
APL2 Mainframe Features	20
Frank Diehl	
APL und MQSeries Die Performance stimmt	24
Martin Hintz	
Verarbeitung von Großrechner- daten im PC-Umfeld	28
Maren Röttgers, Nils Dennstedt	
Die Hamburg-Mannheimer präsentiert ihre SRBB	29
Sergy Alpin	
Java-APL+Win Interface	30
Dr. Wolf Dehus	
Opa und das Internet	33
Blickpunkte	34
Aufnahmeformular	36
Impressum	2
Editorial	3

6 APL als Client & Server für SAP R2/R3

Über die Remote Function Call Schnittstelle (RFC) bietet SAP externen Systemen die Möglichkeit, Funktionsbausteine aus SAP R2/R3 Systemen zu nutzen, und darüber hinaus können auch externe Funktionsbausteine für SAP Systeme zur Verfügung gestellt werden. Mit Hilfe des neuen Funktionspakets S-APL kann man SAP Client- und Server-Applikationen mit APL entwickeln. Damit steht SAP-Anwendern ein mächtiges Werkzeug zur Verfügung, mit dem sie multidimensionale Daten verarbeiten und falls nötig auch interaktiv analysieren können.

13 Die mathematischen Funktionen von Postscript

PostScript ist faktisch genauso alt wie TeX, im Vergleich dazu aber vielen Computeranwendern nicht so geläufig, insbesondere wenn es darum geht, die Eigenschaften und Funktionen von Postscript zu beurteilen. Außerdem wird häufig vergessen, dass sich mit den PostScript-Funktionen viele Dinge erledigen lassen, für die man sonst auf externe Programme zurückgreifen muß. In diesem Beitrag wird dies anhand der mathematischen Funktionen und des LATEX-Pakets „pst-plot“ gezeigt.

16 Linux und APL: Systemnahe Aufgaben

Unter LINUX kann APL auch für systemnahe Arbeiten herangezogen werden. So kann man beispielsweise Postscript-Daten in PDF-Daten (Acrobat-Reader-Formate) konvertieren. In diesem Beitrag wird vorgeführt, wie man LINUX in ein Computernetzwerk einbindet und anschliessend mit sehr einfachen Mitteln Informationen von nahezu beliebigen Quellen verarbeiten und an beliebige Ziele verteilen kann. Als technische Schnittstellen werden Datei-/Verzeichnis-Freigaben sowie Standard-Druck-Schnittstellen benutzt. Zu den vorgestellten Möglichkeiten zählen das Auslesen von Verzeichnissen, die Unterscheidung zwischen Verzeichnissen und Dateien, das Erkennen, ob Dateien in Benutzung sind, das Verschieben von Dateien sowie das Konvertieren von Dateien in das PDF-Format.

20 Documentation: APL2 Mainframe Features

Why Mainframe Features? *Customers often simply don't know about all the features of APL2.* Many features are documented in books that most users do not own. This article gives an overview of all non-interpreter facilities and enhancements made since last release.

Herausgeber:	Dipl.-Math. Dieter Lattermann, Walldorf	Verlag:	RHOMBOS-VERLAG, Berlin Kurfürstenstr. 17 D-1785 Berlin Tel. (030) 261 9461 Fax (030) 261 6300 Email: verlag@rhombos.de, Web: www.rhombos.de
Redaktion:	Dipl.-Volksw. Martin Barghoorn (verantwortlich) Technische Universität Berlin Franklinstr. 28 D-10587 Berlin Tel. (030) 314 24392 Fax (030) 314 25901 Email: barg@cs.tu-berlin.de	Gestaltung:	RHOMBOS-VERLAG, Berlin
Erscheinungsweise	halbjährlich	Druck:	Kopierfabrik, Eberswalde
Erscheinungsort:	Berlin	Copyright:	APL Germany e.V. (für alle Beiträge, die als Erstver- öffentlichung erscheinen)

Homepage des APL-Journals: <http://www.rhombos.de/apljourn.htm>

FTP-Server zum Datei-Upload: <ftp.cs.tu-berlin.de/pub/lang/apl/stat/incoming>.

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung von Redaktion und Herausgeber nicht übernommen werden. Sämtliche Veröffentlichungen im APL-Journal erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Mit Namen gekennzeichnete Artikel geben nicht unbedingt die Meinung des Herausgebers oder der Redaktion wieder. Für unverlangte Einsendungen wird keine Haftung übernommen. Nachdruck ist nur mit Zustimmung des Herausgebers sowie mit Quellenangabe und Einsendung eines Beleges gestattet. Überarbeitungen eingesandter Manuskripte liegen im Ermessen der Redaktion.

Wie immer freuen wir uns auf Ihr Feedback zu diesem Heft. Schreiben Sie uns an barg@cs.tu-berlin.de, oder faxen Sie an 030 - 314 25901.

Technische Hinweise für Autorinnen und Autoren

Manuskripte werden von der Redaktion entgegengenommen. Sie müssen im Besitz der Nutzungsrechte für die von Ihnen eingereichten Texte, Fotos und Zeichnungen sein. Die Zustimmung zum Abdruck wird vorausgesetzt.

Manuskripte und Beiträge aller Art werden in jeder Form entgegengenommen. Es werden jedoch Beiträge in gespeicherter Form bevorzugt. Bitte speichern Sie Ihren Text im Format Ihrer Textverarbeitung und zusätzlich als RTF-Datei (Rich Text Format). Sie brauchen sich nicht die Mühe machen, Ihren Text aufwendig zu formatieren. Auszeichnungen im Text sollten sich auf „Kursiv“ beschränken. Ihre Datei schicken Sie der Redaktion bitte auf einem Datenträger (Diskette oder CD-ROM). Alternativ hierzu können Sie der Redaktion Ihre Daten auch per Email (barg@cs.tu-berlin.de) oder auch zum FTP-Server der Technischen Universität Berlin schicken. Die Adresse lautet: <ftp.cs.tu-berlin.de/pub/lang/apl/stat/incoming>.

Auch ein komfortables FTP-Transferprogramm für Windows gibt es beim ftp-Server des Fachbereichs Informatik der TUB unter: <ftp.cs.tu-berlin.de/pub/lang/apl/bin/>

Wenn Sie Ihr Manuskript einreichen, sollten Sie zusätzlich eine Gliederung und Zusammenfassung Ihrer Arbeit vorlegen. In der Zusammenfassung sollte in wenigen Sätzen die wichtigste Aussage Ihres Textes wiedergegeben sein.

Grafiken

Bitte senden Sie uns einen sauberen Ausdruck der Graphiken auf rein-weißem Papier. Linien (keine Haarlinien!) in den Graphiken sollten Schriften so groß gewählt sein, dass sie auch bei (stark) verkleinerter Wiedergabe im Heft noch zu erkennen sind; auf der Rückseite vermerken Sie bitte Ihren Namen und die Nummer der Abbildung; Zusätzlich wäre es schön, wenn Sie uns die mit einem gängigen Vektor-Grafikprogramm erstellten Zeichnungen und Grafiken auch als Datei liefern. Grafikdateien (Strichzeichnungen) sollten möglichst im Austauschformat EPS (skalierbar) mit 300 bis 600 dpi Auflösung in schwarz-weiß bzw. Graustufe abgespeichert werden. Nach Rücksprache können sie zusätzlich Ihre Grafiken in Ihrem Original-Dateiformat liefern oder in ihr Textdokument einbetten.

Digitale Fotos benötigen wir im TIF-Format mit einer maximalen Auflösung von 250 dpi. Nicht geeignet sind Abbildungen und Fotos, wenn Sie aus dem Internet stammen beziehungsweise im Original als Bildschirmdarstellung (Auflösung 75 - 150 dpi) vorliegen. Ebenfalls nicht geeignet sind Grafiken, die mit einem pixelorientierten Zeichenprogramm erstellt wurden.

EDITORIAL



Walldorf, 11. Dezember 2001

Liebe APL-Freunde,

nun wächst zusammen, was zusammen gehört. Was in der Politik nur mühsam voran geht, macht den Array-Verarbeitern weniger Schwierigkeiten. APL Germany e.V. ist jetzt Mitglied des APL-Arbeitskreises der GSE (G.U.I.D.E. SHARE Europe – Benutzerorganisation von IBM-Kunden). Das bedeutet, dass Treffen unseres Vereins in Zukunft häufiger gemeinsam mit Tagungen des GSE-Arbeitskreises durchgeführt werden können, was Reisekosten minimiert und das Gewinnen von Referenten erleichtert. Spräche

ich „Neusprech“, würde ich von Synergieeffekten reden. Das bedeutet nicht, das wir IBM-lastig werden, wir werden weiterhin allen Anbietern und Nutzern von Arrayverarbeitungssystemen ein Forum bieten und dies auch im APL Journal zeigen. Sie können sich in der neuen Ausgabe, die wir Ihnen hiermit vorlegen, davon überzeugen.

Nach der erfolgreichen Hamburger gemeinsamen Tagung wollen wir im April 2002 auch wieder mit der GSE zusammen ein Treffen durchführen. Dem steht noch ein Problem entgegen, das hoffentlich noch überwunden wird: Der ins Auge gefasste Gastgeber (IBM in München) hat keinen Raum verfügbar, der die vereinten APL-Enthusiasten fassen kann.

Hier noch fünf gute Nachrichten, die auf früheren Tagungen zwar verlautbart wurden, aber vielleicht nicht zu Ihnen allen vorgedrungen sind:

- o Für Lehr- und Forschungszwecke kann man eine kostenlose Lizenz von IBM APL2 bekommen (Kontakt: Klaus Drebenstedt, Kwd@de.ibm.com)
- o Tilman Otto bietet seinen APL-Compiler (noch) kostenlos an: TilmanOtto@web.de
- o APL Germany hat eine Lizenz für Dyalog APL (Dieter_Lattermann@compuserve.com)
- o Soliton bietet eine kostenlose Version von Sharp APL für Linux an (sales@soliton.com)
- o IBM arbeitet an einer Linux-Version von APL2 (das ist keine Ankündigung)

Zum Schluss noch zwei Tipps für Engagierte: Schreiben Sie doch mal einen Artikel für das APL Journal und/oder reichen Sie einen Vortrag für die nächste internationale APL-Konferenz ein: Array Processing Languages - Lore, Problems and Applications in Madrid vom 22. – 25. Juli 2002 (dazu haben Sie Zeit bis zum 31. Januar 2002)

Ich wünsche Ihnen erfolgreiches Jahr 2002,

Ihr

Dieter Lattermann

APL2002

(APL-LPA 2002)

ARRAY PROCESSING LANGUAGES

Lore, Problems and Applications

July 22-25, 2002

Madrid, Spain

Call for Participation: Papers, Tutorials and Workshops

The APL 2002 Conference will be held July 22-25, 2002 in Madrid, Spain. This is the annual international conference on APL and other array programming languages (A+, J, K, Matlab, Mathematica, Nial, S-Plus, etc.). The conference will be held in cooperation with ACM (SIGAPL) and the British APL Association.

The subject matter of the APL 2002 Conference is three-fold:

- **Lore:** *The history of Array Programming Languages, including historical considerations on any of the following or other questions: design, implementations, applications, difficulties, standards wars, divergence, vendors, etc.*
- **Problems:** *Current use of Array Programming Languages in research and innovation, stressing the advantages provided by the use of the language in building a prototype or the final system. Straightforward descriptions of advanced research questions, or proposals for extensions for current or new Array Programming Languages are also encouraged.*
- **Applications:** *Actual use of Array Programming Languages for real-life applications, with successful case studies and comparisons with other languages. The application in question need not be at the front end of research in its field.*

The APL 2002 Conference will also hold a number of Tutorials, hands-on Workshops and Panel discussions. The conference committee invites you to submit your proposals and intention to participate, indicating the following information:

- *Papers (40 minute presentations): an extended abstract with a maximum of two pages.*
- *Short Communications (20 minute presentations): an abstract with a maximum of one page.*
- *Tutorials (90 minute session): a one-page program of the proposed tutorial.*
- *Workshops (two-hour hands-on session): a one-page description of the proposed workshop.*

In all cases:

- *Your name and affiliation.*
- *Your e-mail address.*
- *Type of proposal (paper, tutorial, workshop, panel discussion).*
- *Title of your proposal*

Submissions via e-mail

Abstracts are welcome now and should be submitted as plain ASCII files. The preliminary deadline for paper submission is January 31st 2002. More information you can find on our web page at <http://apl2002.com>.

- *Submit proposals and short communications to Xavier.Alaman@ii.uam.es*

- Submit proposals for tutorials and workshops to enrique@cs.york.ac.uk

Submissions via fax:

Submit all proposals to Xavier Alaman, +34 91 348 2235

Submissions via standard mail:

Submit all proposals to:

Xavier Alaman
E.T.S. Informatica
Universidad Autonoma de Madrid
Campus de Cantoblanco
28049 Madrid SPAIN

■ Organization

Conference Chair:

Manuel Alfonso (Universidad Autónoma de Madrid, Spain)
eMail: Manuel.Alfonseca@ii.uam.es

Local Organizing Chair:

Pilar Rodríguez (Universidad Autónoma de Madrid, Spain)
eMail: Pilar.Rodriguez@ii.uam.es

Local Logistics:

Pablo Castells (Universidad Autónoma de Madrid, Spain)
eMail: Pablo.Castells@ii.uam.es

Francisco Saiz (Universidad Autónoma de Madrid, Spain)
eMail: Francisco.Saiz@ii.uam.es

Program Committee:

Xavier Alamán (Universidad Autónoma de Madrid, Spain)
eMail: Xavier.Alaman@ii.uam.es

Robert Bernecky

eMail: bernecky@acm.org

Robert Brown (ACM SIGAPL)

eMail: bob@acm.org

James Brown (SmartArrays Corporation):

eMail: Jim.Brown@SmartArrays.com

Dieter Lattermann:

eMail: Dieter_Lattermann@compuserve.com

Adrian Smith (British APL Association):

eMail: adrian@causeway.co.uk

Tutorials/Workshop Chair:

Enrique Alfonso

eMail: enrique@cs.york.ac.uk

Webmaster:

Estrella Pulido (Universidad Autónoma de Madrid, Spain)

eMail: Estrella.Pulido@ii.uam.es

ACM Contact:

Robert Brown (ACM SIGAPL)

eMail: bob@acm.org

See you in Madrid!

July 22-25, 2002



Contact Information:

<http://apl2002.com>

Conference site

The Conference will be held at the Technical School of Computer Science in the Universidad Autónoma of Madrid. This School is located in a new building, completed in July 1999 and providing all the opportunities for a successful conference. It has lecture theaters and a variety of seminar rooms with computers and projection equipment, plus several workshop rooms with 24 computers each. The University is located 16 km from the center of Madrid, accessible by excellent train and bus lines. Special buses will be provided for participants in the morning and the afternoon. Several hotels with a scale of prices will be offered to visitors.

<http://apl2002.com>

Dr. Tilman Otto

APL als Client & Server für SAP R2/R3

Über die Remote Function Call Schnittstelle (RFC) bietet SAP externen Systemen die Möglichkeit, Funktionsbausteine aus SAP R2/R3 Systemen zu nutzen, und darüber hinaus können auch externe Funktionsbausteine für SAP Systeme zur Verfügung gestellt werden. Mit Hilfe des neuen Funktionspakets S-APL kann man SAP Client- und Server-Applikationen mit APL entwickeln. Damit steht SAP-Anwendern ein mächtiges Werkzeug zur Verfügung, mit dem sie multidimensionale Daten verarbeiten und falls nötig auch interaktiv analysieren können.

Remote Function Call API (RFC)

SAP Systeme bestehen im allgemeinen aus einem Verbund mehrerer großer Rechner (Datenbank- und Applikations-Servern). Die auf den verschiedenen Systemen laufenden SAP Applikationen kommunizieren über Netzwerk mithilfe der sogenannte *Remote Function Call* Schnittstelle. SAP stellt diese intern verwendete Schnittstelle auch für externe Systeme in Form von Bibliotheken zur Verfügung. Es werden alle Unix-Plattformen unterstützt, auf denen auch R2/R3 verfügbar ist. Zusätzlich stehen diese Bibliotheken auch für Windows, OS/2 und Macintosh zur Verfügung.

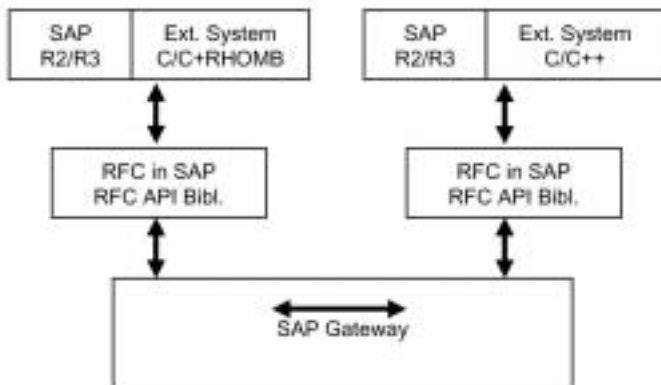


Abbildung 1: Kommunikation über RFC

Bindeglied zwischen den über RFC verbundenen Systemen bildet die sogenannte SAP Gateway. Außer der Verbindung von externen Systemen mit einem SAP System könnten daher über RFC im Prinzip auch zwei externe Systeme miteinander verbunden werden.

Die von SAP zur Verfügung gestellte RFC Bibliothek erfüllt mehrere Aufgaben:

- Verbindungsaufbau (Auswahl des Zielsystems, Verifikation der Benutzerdaten, etc.)
- Netzwerkkommunikation (über TCP/IP)
- Datenkonversion; die miteinander verbundenen Systeme können unterschiedliche interne Datenformate haben (z.B. andere „byte order“)
- Datentransfer über Netzwerk

- Aufruf von remote-fähigen Programm-Modulen in SAP Systemen
- Empfang von remote-Aufrufen aus SAP Systemen heraus, d.h. das externe System hat Serverfunktion

APL als SAP R2/R3 Client

Um aus APL heraus Programm-Module eines SAP Systems aufrufen zu können, wurde ein Funktionspaket in der Sprache C entwickelt (S-APL genannt) und als zusätzliche APL Funktionen zur Verfügung gestellt. Zur Implementierung wurde das APL System APL2C¹ des Autors verwendet. Die Funktionen können aber auch an andere APL Systeme angepasst werden (z.B. als externe Funktionen für IBM's APL2). Abbildung 2 zeigt den schematischen Aufbau der Verbindung.

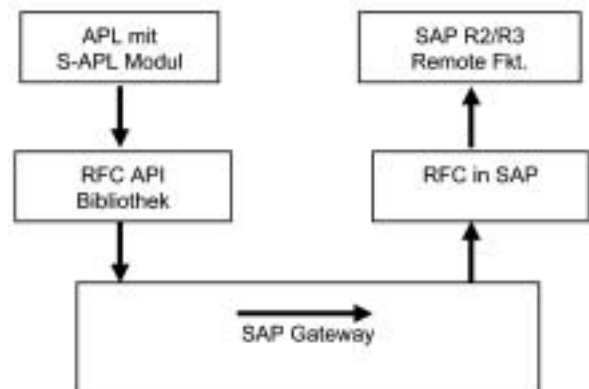


Abbildung 2: APL als SAP Client

Verbindungsaufbau

Zum Aufbau einer neuen Verbindung zum SAP System dient die Funktion R3OPEN:

```
H←R3OPEN 'CLIENT=... USER=...'
```

Über weitere Parameter können u.a. Passwort und gewünschtes des SAP Systems konfiguriert werden:

1 APL2C - The APL Compiler, <http://www.apl2c.com>

PASSWD, TYPE, LANG, LCHECK, TRACE, DEST, GWHOST, GWSERV, MSHOST, R3NAME, GROUP, ASHOST, SYSNR, TPHOST, TPNAME, USE_SAPGUI, ABAP_DEBUG, SNC_MODE, SNC_QOP, SNC_MYNAME, SNC_PARTNERNAME, SNC_1

Das Ergebnis der Funktion R3OPEN ist eine Zahl H (Handle), die für alle weiteren Aufrufe zum SAP System benötigt wird.

Aufruf in das SAP System

Bereits in jedem SAP R3 Basis System sind über 5.000 remote-fähige Programm-Module enthalten. Das bedeutet, dass diese Bausteine von anderen Systemen über RFC nutzbar sind. Zusätzlich kann der SAP Entwickler eigene Funktionsbausteine erstellen und als „remote-fähig“ markieren.

Schnittstelle

Zum Aufruf eine solchen Remote-Funktion von APL aus dient die Funktion R3CALL:

```
H R3CALL <FKT> <EXP> <IMP> <TAB>
```

Der linke Parameter ist das Handle, das die Verbindung zum SAP System beschreibt (siehe R3OPEN). Die Parameter auf der rechten Seite haben folgende Bedeutung:

<FKT> Bezeichnet den Namen des Funktionsbausteins im SAP System. Funktionsbausteine im SAP System sind in der SAP-spezifischen Sprache ABAP geschrieben.

<EXP> -Sogenannte „Export“ Parameter. Das bedeutet, hier werden alle Parameter angegeben, deren Werte an den Remote-Funktionsbaustein übergeben werden sollen.

<IMP> Sogenannte „Import“ Parameter. Das bedeutet, hier werden alle Parameter angegeben, deren Werte vom Remote-Funktionsbaustein zurückgegeben werden.

<TAB> In diesem Parameter werden sogenannte „interne“ Tabellen angegeben. Interne Tabellen werden in beiden Richtungen übergeben, das heißt, es können sowohl Tabellendaten an die Funktion übergeben werden als auch Tabellen innerhalb der Funktion gefüllt und zurückgegeben werden. Die Funktionen zur Bearbeitung interner Tabellen werden im Abschnitt „Interne Tabellen“ genauer beschrieben.

Die SAP Programmiersprache ABAP hat eine Besonderheit bei der Parameterübergabe an Funktionsbausteine: Die Parameter werden nicht durch deren Reihenfolge spezifiziert sondern die Zuordnung erfolgt allein über den Parameter-Namen. Die Reihenfolge der Parameter spielt also beim Aufruf keine Rolle. Das Format für die Parameter <EXP> und <IMP> ist wie folgt definiert:

```
'Typ1:Param1 Typ2:Param2 ...'
```

Hierbei steht Typ1 und Typ2 für den Datentyp der entsprechenden Parameter. Param1 bzw. Param2 bezeichnen hierbei sowohl die Namen von APL Variablen im APL-Work-

space als auch den Namen des Parameters für das SAP System. Bei den <EXP> Parametern müssen die APL Variablen bereits vor dem Aufruf existieren (da deren Werte ja als Parameter an das SAP System übergeben werden sollen). Bei <IMP> Parametern werden die entsprechenden APL-Variablen gegebenenfalls angelegt bzw. überschrieben.

Die Datentypen Typ1, Typ2 usw. spezifizieren entweder einen primitiven Datentyp (siehe Tabelle unten) oder einen strukturierten Datentyp, der im SAP System im sogenannten Data Dictionary definiert ist. Bereits im SAP R3 Basis-System sind mehrere tausend verschiedener Daten- bzw. Tabellentypen definiert. Primitive Datentypen werden in eckigen Klammern definiert (z.B. '[IC20]' bedeutet eine Struktur bestehend aus einer Ganzzahl und einer Zeichenkette der Länge 20), während Datentypen aus dem SAP Data-Dictionary direkt über deren Struktur-Namen spezifiziert werden. Die folgende Tabelle zeigt alle Primitive Datentypen der Sprache ABAP und deren Repräsentation in APL:

ABAP Typ	Format	Beispiel	APL Rep.
C	Buchstabe	'A'	Char
X	Byte	0x45	Char
N	Nummer	"1234"	Char
I	Zahl	334455	Integer
P	BCD	0x1234	Double
D	Datum	"YYYYMMDD"	Char-Vektor
T	Zeit	"HHMMSS"	Char-Vektor
F	Fließkomma	3.14159	Double

Abbildung 3: Interne ABAP Datentypen

Interne Tabellen

Der Parameter <TAB> zur Funktion R3CALL ist wie folgt definiert:

```
'Typ1:Tab1 Typ2:Tab2 ...'
```

Hierbei bezeichnen Typ1 und Typ2 wieder primitive Datentypen oder im Data-Dictionary definierte Datenstrukturen. Tab1 und Tab2 sind hierbei Namen von Variablen innerhalb des APL Workspaces. Sollen Daten in den internen Tabellen an den aufzurufenden SAP Funktionsbaustein übergeben werden, so müssen diese Variablen gültige interne Tabellen bezeichnen. Sind diese Variablen nicht definiert, so werden interne Tabellen vom bezeichneten Typ erzeugt, damit diese vom SAP Funktionsbaustein gefüllt werden können. Nach dem Aufruf bezeichnen die APL Variablen dann die internen Tabellen.

Interne Tabellen werden innerhalb der RFC Bibliothek erstellt und verwaltet. Interne Tabellen werden daher über eindeutige Bezeichner (Ganzzahl bzw. Handle) im APL verwaltet. Innerhalb des S-APL Moduls stehen mehrere

Funktionen zur Verwaltung interner Tabellen zur Verfügung:

```
H ITNEW 'TYP:TABNAME'
```

erzeugt eine interne Tabelle des Typs TYP. Die APL Variable TABNAME enthält anschließend das Handle für die interne Tabelle.

```
<Struktur-Daten> ITAPPEND TABNAME
```

hängt eine oder mehrere Zeilen an die interne Tabelle an. Das im linken Argument übergebene Array muss entsprechend dem Typ der Tabelle strukturiert sein (siehe Beispiel unten).

```
T<Zeilen-Nummern> ITREAD TABNAME
```

Liest den Inhalt einer, mehrerer oder aller (ohne linkes Argument) Zeilen einer internen Tabelle

```
T<Spalten-Nummern> ITREADCOL TABNAME
```

Liest den Inhalt einer oder mehrerer Spalten einer internen Tabelle. Da je eine Spalte in ein APL-Array gelesen wird, ist diese Form schneller und vom Speicherbedarf günstiger als über die Funktion ITREAD.

```
T ← ITSIZE TABNAME
```

Ermittelt die Größe der internen Tabelle. Das Ergebnis ist ein Vektor der Länge zwei und enthält die Anzahl der Zeilen und Spalten der Tabelle.

```
T ← <Zeilen-Nummern> ITDELETE TABNAME
```

Löscht eine, mehrere oder alle (ohne linkes Argument) Zeilen einer internen Tabelle.

```
T ← ITFREE 'TABNAME'
```

Löscht alle Daten einer Tabelle und zerstört die Tabelle selbst.

Beispiel: Tabellen-Erzeugung, -Erweiterung, -Lesen

```
ITNEW '[IC10]:T
3 'TEST' ITAPPEND T
(7 'DATEN') (55 'XYZ)' ITAPPEND T
ITREAD T
3 TEST
7 DATEN
55 XYZ
ITSIZE T
3 2
```

► Beispiel

Es soll die SAP Funktion „Z_APL_READ_FLIGHTS“ aufgerufen werden. Die in ABAP implementierte Funktion liest über eine einfache Datenbankabfrage die Daten der Tabelle SFLIGHT (Demonstrations-Daten über Fluglinien) in die interne Tabelle mit dem Namen FLIGHTS. Abbildung 4 zeigt die im SAP System definierte Funktion „Z_APL_READ_FLIGHTS“.

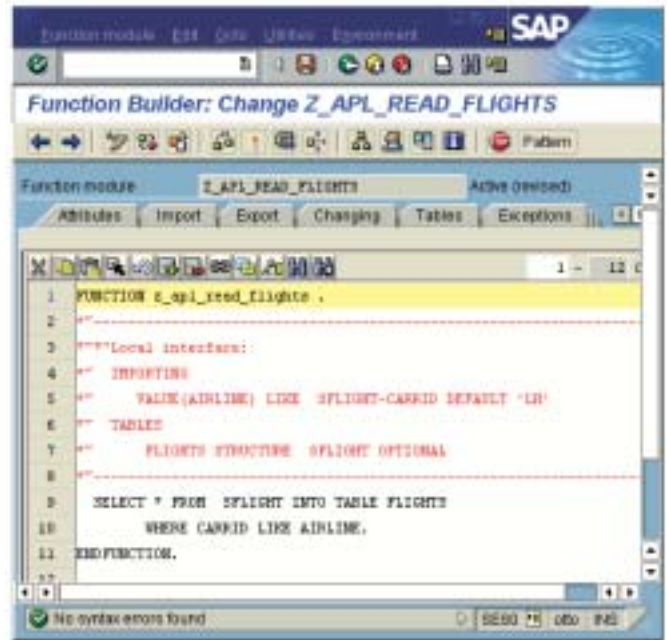


Abbildung 4: Remote ABAP Funktion

Um diese Funktion aus APL heraus aufrufen zu können, muss zuerst eine Verbindung zu einem SAP Server hergestellt werden:

```
H←R3OPEN 'SYSNR=00 USER=...'
H
1
```

Der Import-Parameter „AIRLINE“ der ABAP Funktion muss über eine APL Variable gleichen Namens übergeben werden (hier mit dem Wert 'LH' für „Lufthansa“)ð. Die Funktion R3CALL erhält als Parameter den ABAP Funktionsnamen, die Liste der Import Parameter (hier ein Parameter mit dem Namen „AIRLINE“ und vom Typ „SFLIGHT-CARRID“, d.h. vom gleichen Typ wie das Feld „CARRID“ in der Tabelle SFLIGHT), die Liste der Export-Parameter (hier leer) und die Liste der internen Tabellen (hier eine Tabelle mit dem Namen FLIGHTS vom Data-Dictionary-Typ SFLIGHT).

```
AIRLINE←'LH'
FN←'Z_APL_READ_FLIGHTS'
IMP←'SFLIGHT-CARRID:AIRLINE'
EXP←''
TAB←'SFLIGHT:FLIGHTS'
H R3CALL FN IMP EXP TAB
```

Die Liste der Flüge ist nun in der internen Tabelle FLIGHTS gespeichert. Die Anzahl der Lufthansaflüge lässt sich leicht ermitteln:

```
ITSIZE FLIGHTS
110 10
```

Die ersten 10 Zeilen der Tabelle FLIGHTS lassen sich leicht über die Funktion ITREAD lesen:

```
(i 10) ITREAD FLIGHTS
000 LH 0400 20010608 1332 DEM A310-300 280 258 321518.16
000 LH 0400 20010914 1332 DEM A310-300 280 261 326486.52
```

```
000 LH 0400 20011028 1332 DEM A310-300 280 259 323955.72
000 LH 0400 20011128 1332 DEM A310-300 280 266 331028.64
000 LH 0400 20011207 1332 DEM A310-300 280 266 332267.4
000 LH 0400 20020215 1332 DEM A310-300 280 61 76936.32
000 LH 0400 20020412 1332 DEM A310-300 280 35 43969.32
000 LH 0400 20020524 1332 DEM A310-300 280 45 55211.4
000 LH 0400 20020621 1332 DEM A310-300 280 29 35870.76
000 LH 0400 20020705 1332 DEM A310-300 280 2 2664
```

Die Bedeutung der einzelnen Felder kann über die (hier nicht weiter erläuterte) Funktion ITFIELDS bestimmt werden:

```
H ITFIELDS 'SFLIGHT'MANDT CARRID CONNID
FLDATE PRICE
CURRENCY PLANETYPE SEATSMAX SEATSOCC
PAYMENTSUM
```

Die Anzahl der maximal verfügbaren Sitze (8.Feld: „SEATSMAX“), die Anzahl der gebuchten Sitze (9.Feld: „SEATSOCC“) und die Summe aller Flugbuchungen (10.Feld „PAYMENTSUM“) lässt sich leicht berechnen:

```
F←ITREAD FLIGHTS
+/[1]F[;8 9 10]
28990 15552 13424827.67
```

Das heißt, von 28990 sind bereits 15552 Sitze gebucht. Die Gesamtbuchungssumme beträgt ca. 13.4 Mio DM.

APL als SAP R2/R3 Server

Um in APL implementierte Programme von SAP Systemen heraus aufrufen zu können, wurden in S-APL zusätzliche Funktionen zur Verfügung gestellt, mit denen ein vollwertiger SAP Remote Server in APL realisiert werden kann. Abbildung 5 zeigt den schematischen Aufbau der Verbindung:

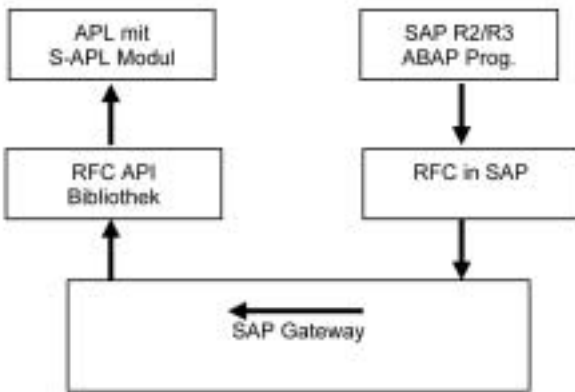


Abbildung 5: APL als SAP Server

Damit ein APL Programm als Funktionsbaustein innerhalb des SAP Systems zur Verfügung stehen kann, müssen mehrere Voraussetzungen erfüllt sein. Das APL Programm muss im SAP System als RFC-Ziel eingetragen werden (in der Tabelle RFCDES über Transaktion „SM59“) und das APL-Programm muss sich an der Gateway anmelden (d.h. registrieren) um auf einen Aufruf aus dem SAP System zu warten.

Zum Aufbau einer Verbindung zur SAP Gateway und um auf einen Aufruf aus dem SAP System zu warten, dient die Funktion R3SERVE:

```
R3SERVE '-a<PROG> -g<HOST> -x<GW> ...'
```

Im rechten Argument können diverse Parameter angegeben werden, wie etwa die SAP Gateway an der sich das APL System als Server registrieren soll.

Erfolgt ein Aufruf aus dem SAP System, dann wird in der Funktion R3SERVE der Name der aufzurufenden APL Funktion ermittelt. Falls eine entsprechende Funktion im geladenen APL Workspace nicht existiert, dann wird ein Fehler an das SAP System gemeldet. Existiert die APL Funktion, dann wird diese aufgerufen. Als einzigen Parameter erhält die aufgerufenen APL Funktion ein Handle (Ganzzahl) für die RFC Verbindung. Über dieses Handle können dann die Importing-Parameter und internen Tabellen vom SAP System gelesen werden. Dies erfolgt mithilfe der Funktion R3GET:

```
H R3GET <IMP> <TAB>
```

Das linke Argument ist das Handle der RFC Verbindung zum aufrufenden SAP System. Das Format des rechten Argumentes zur Ermittlung der übergebenen Parameter und Tabellen ist analog zur den Argumenten <IMP> und <TAB> der Funktion R3CALL, die bereits beschrieben wurde.

Über die Funktion R3SEND können die Exporting-Parameter und Tabellen an das aufrufende SAP System zurückgegeben werden:

```
H R3SEND <EXP> <TAB>
```

Nach der Rückgabe der Parameter an das aufrufende SAP System und Beendigung des aufgerufenen APL Programms kehrt die Kontrolle zur Funktion R3SERVE zurück um auf den nächsten Aufruf warten zu können.

Beispiel

Die folgende APL Funktion zur Berechnung von Primzahlen (nach der Methode des „Siebs von Erathostenes“) soll als SAP Remote Server zur Verfügung gestellt werden:

```
[0] P←PRIMESTO N;ØIO;J;I;S
[1] ØIO←1
[2] P←iN
[3] S←N+0.5
[4] J←2
[5] LOOP:
[6] I←J×1+i|N÷J
[7] P[I]←0
[8] J←J+1+2|J
[9] →(J≤S)/LOOP
[10] P←1+i(P≠0)/P
```

Um diese Funktion als Remote-Funktionsbaustein von ABAP heraus aufrufen zu können wird die folgende Funktion definiert.

```
[0] PRIMES H;LIMIT;PNUMBERS
[1] H R3GET '[I]:LIMIT' '[I]:PNUMBERS'
[2] (PRIMESTO LIMIT) ITAPPEND PNUMBERS
[3] H R3SEND '' '[I]:PNUMBERS'
[4] LIMIT
```

Diese Funktion hat als einzigen Parameter ein Handle H für die RFC Verbindung zum aufrufenden System, wie es von der Funktion R3SERVE geliefert wird.

In Zeile 1 wird der Import-Parameter LIMIT und die interne Tabelle PNUMBERS vom SAP System bezogen. LIMIT ist hier eine Ganzzahl und PNUMBERS eine interne Tabelle von Ganzzahlen. In Zeile 2 werden die Primzahlen bis zur Zahl LIMIT mithilfe der Funktion PRIMESTO berechnet und die Zahlen an die Tabelle PNUMBERS angehängt. In Zeile 3 wird die Tabelle PNUMBERS wieder an das SAP-System zurückgegeben (hier ohne Export-Parameter). Zeile 4 gibt den Wert LIMIT an der APL Konsole aus.

Zum Test wird nun das APL System in den Servermodus versetzt:

```
R3SERVE '-aapl.123 -glh -xsapgw00'
```

Die in Abbildung 6 gezeigte ABAP Funktion Z_APL_GET_PRIMES ruft die APL Funktion PRIMES auf:

In Zeile 8 wird die Funktion „PRIMES“ im Zielsystem „APL2C“ aufgerufen. Das RFC-Ziel „APL2C“ ist in der SAP Systemtabelle RFCDES definiert. Nach Aktivierung der Funktion Z_APL_GET_PRIMES kann diese im SAP System von anderen ABAP Programmen aufgerufen werden oder di-

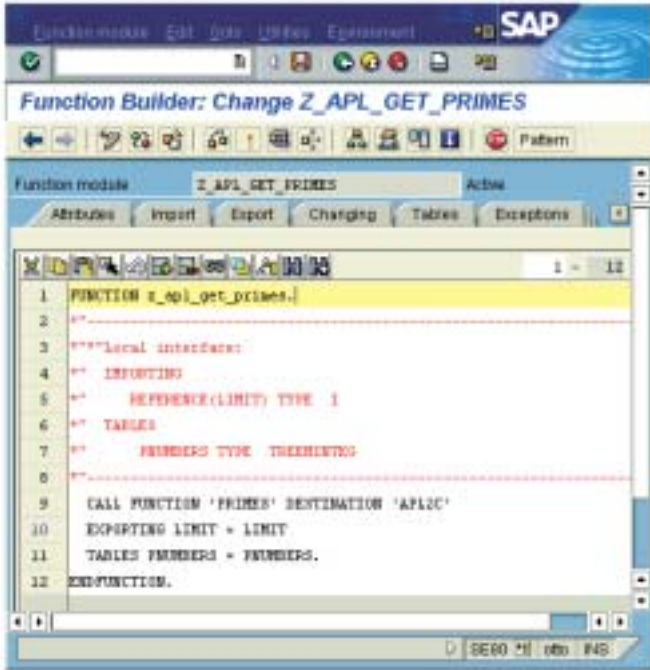


Abbildung 6: ABAP Programm mit APL-Aufruf



Abbildung 8: Ergebnis des Funktionsaufrufs



Abbildung 7: Test der Funktion Z_APL_GET_PRIMES



Abbildung 9: Die berechnete Tabelle PNUMBERS

rekt getestet werden. In der in Abbildung 7 gezeigten Test-Umgebung kann der Wert für LIMIT direkt eingegeben werden.

Nach nur ca. 37 Millisekunden (Pentium III, 1 GHz) steht das Ergebnis bereit. Die 1229 Primzahlen bis zur Zahl 10.000 wurden in der APL Funktion PRIMES berechnet:

Ein Blick in die Tabelle PNUMBERS zeigt die berechneten Primzahlen:

Kompilierung

Mithilfe des im APL2C System integrierten Compilers lassen sich APL Funktionen (und alle referenzierten Funktionen) in C Code umsetzen. Dies erlaubt die einfache Generierung von SAP Client oder Server Applikationen in Form von ausführbaren Dateien (EXE Dateien). Darüber hinaus ermöglicht die Verfügbarkeit des C Codes die homogene Integration der generierten C-Dateien in andere Applikationen auf Source-Ebene.

Zur Demonstration dieses Mechanismus wird hier der Primzahl-Server kompiliert. Als Quelle dient das folgende APL-Programm:

```
[0] PRIMESERVER
[1] R3SERVE '-aapl.123 -glh -xsapgw00'
[2] →0
[3] PRIMES 1
```

In Zeile 1 registriert sich das Programm an der SAP-Gateway unter der Programm-ID „apl.123“ die in der SAP Systemtabelle RFCDES als logisches Ziel „APL2C“ eingetragen ist. Da die Funktion R3SERVE nie beendet wird, erscheinen die Zeilen 2 und 3 eigentlich überflüssig. Die Funktion PRIMES in Zeile 3 wird aufgerufen, da der APL2C Compiler nicht wissen kann, welche Funktionen zur Laufzeit vom SAP Server aufgerufen werden. Aus diesem Grund müssen alle vom SAP Server aufrufbaren Funktionen hier explizit aufgeführt werden.

Die Kompilierung eines APL Programms erfolgt in APL2C über den Befehl òAPL2C (die zu kompilierende Funktion wird als linkes Argument angegeben):

```
'PRIMESERVER' òAPL2C 'PRIMESERVER'
APL2C: Compiling APL
=====
TARGET OS: Microsoft Windows 95/NT/2000
TARGET CC: Microsoft Visual C/C++ Compiler
GENERATING C CODE:
-----
PRIMESERVER → apl_PRIMESERVER.c
PRIMES → apl_PRIMES.c
PRIMESTO → apl_PRIMESTO.c
```

Im Verzeichnis PRIMESERVER (rechtes Argument zu òAPL2C) befinden sich dann die folgenden Dateien:

```
C:\APL2C\PRIMESERVER>DIR /B
apl_PRIMESERVER.c
apl_PRIMES.c
```

```
apl_PRIMESTO.c
PRIMESERVER.h
PRIMESERVER.c
makefile
apl2c.c
apl2c.h
```

Die Funktion òAPL2C erzeugt bereits ein Makefile für die Übersetzung in eine ausführbare Datei. Die Übersetzung kann dadurch unmittelbar durchgeführt werden:

```
C:\APL2C\PRIMESERVER>NMAKE

Microsoft (R) Program Maintenance Utility
Version 1.62.7022
Copyright (C) Microsoft Corp 1988-1997. All rights reserved.

cl -I..\src -MD -nologo -c apl_PRIMESERVER.c
cl -I..\src -MD -nologo -c apl_PRIMES.c
cl -I..\src -MD -nologo -c apl_PRIMESTO.c
cl -I..\src -MD -nologo -c PRIMESERVER.c
cl -I..\src -MD -nologo -c apl2c.c
cl -MD -nologo -o PRIMESERVER.exe
    apl_PRIMESERVER.obj apl_PRIMES.obj apl
    _PRIMESTO.obj PRIMESERVER.obj apl2c.obj
    ..\src\apl2cdm.lib

C:\APL2C\PRIMESERVER>DIR *.EXE

Volume in drive C has no label
Directory of C:\APL2C\PRIMESERVER

PRIMES~1 EXE                20.752  03.12.01  11:33
PRIMESERVER.exe
                            1 file(s)          10.752 bytes
```

Das fertige PRIMESERVER.EXE Programm kann nun gestartet werden und erfüllt die gleiche Funktion wie das im APL Interpreter ausgeführte Programm.

Es ist sogar möglich, das PRIMESERVER Programm mehrfach zu starten. Da die SAP Gateway über ein sogenanntes „Load-Balancing“ verfügt, werden dann die Aufrufe aus dem SAP System an das nächste freie (d.h. wartende) PRIMESERVER Programm weitergegeben. Damit können auch bei länger dauernden Aufgaben (z.B. die Primzahlberechnung bis 10.000.000) die Antwortzeiten des APL-Servers kurz gehalten werden.

Zusammenfassung

Die hier aufgezeigte Schnittstelle zwischen APL und SAP Systemen erweist sich als sehr flexibel und leistungsfähig. Mit minimalem Aufwand lassen sich sowohl SAP Client- als auch Serverapplikationen in APL entwickeln. Der Zugriff auf das SAP Data-Dictionary und die darauf basierende automatische Konversion von Datenstrukturen sowie die verschiedenen Funktionen zur Manipulation von internen Tabellen ermöglichen einen reibungsfreien Datenaustausch zwischen beiden Systemen. Die Generierung von C Code aus den APL Programmen durch den APL2C Compiler erlaubt die einfache Erstellung von ausführbaren Programmen.

men und die homogene Einbindung in andere Applikationen auf Source-Ebene.

Der APL2C Interpreter und Compiler kann kostenlos im Internet heruntergeladen werden, interessierte SAP Anwender können das S-APL Modul testweise direkt vom Autor beziehen.

Kontakt

Dr. Tilman Otto, Paul-Martin-Ufer 13,
68163 Mannheim
eMail: tilman.otto@apl2c.com
Internet: www.apl2c.com

Anzeige

APL2C ... kompiliert APL!

APL2C ist ein neuer APL Interpreter mit der Möglichkeit, APL Funktionen in C Quellcode zu übersetzen (inklusive des dafür benötigten „Makefiles“). Der erzeugte Quellcode kann mit einem C Compiler unmittelbar in ein ausführbares Programm übersetzt werden oder auf Quellcode-Ebene mit anderen C/C++ Applikationen homogen verbunden werden.

APL2C Eigenschaften:

- Extended APL Interpreter (u.a. verschachtelte Arrays, Vektor- und selektive Zuweisung)
- läuft auf allen 32-bit Windows Plattformen (Linux in 2002)
- Konvertiert APL Funktionen, Operatoren & Variablen nach C
- erlaubt selbst-definierte Operatoren
- nur durch den Systemspeicher limitierte Workspace-Größe
- APL Symbole *und* intrinsische ASCII Übersetzung: z.B. {iota}
- Neue Kontrollstrukturen übere mehrere Zeilen im APL-Stil
- Plattform-unabhängige GUI Programmierung (über Tcl/Tk)
- System-Befehle „• IN“ und „• OUT“ zum Daten Im/Export

Weitere Informationen und Download einer 90-Tage Testversion unter www.apl2c.com!

Die Testversion kann kostenlos per Email an register@apl2c.com in eine unbefristete Version freigeschaltet werden (bitte unbedingt Seriennummer aus „Help / About“-Dialog angeben).

Dr. Tilman Otto • Paul-Martin-Ufer 13 • 68163 Mannheim • Germany

info@apl2c.com • Phone 069 -791206183 • FAX 069-791206182

Herbert Voß

Die mathematischen Funktionen von Postscript

PostScript ist faktisch genauso alt wie TeX, im Vergleich dazu aber vielen Computeranwendern nicht so geläufig. Insbesondere wenn es darum geht, die Eigenschaften und Funktionen von Postscript zu beurteilen. Außerdem wird häufig vergessen, dass sich mit den PostScript-Funktionen viele Dinge erledigen lassen, für die man sonst auf externe Programme zurückgreifen muß. In diesem Beitrag wird dies anhand der mathematischen Funktionen und des LATEX-Pakets „pst-plot“ gezeigt.

möglichen Parameter können der Dokumentation zu *pstricks* entnommen werden, die zwar schon älteren Datums ist, jedoch völlig ausreichend.[2] Hier ist im Zusammenhang mit den Funktionen lediglich der Parameter *plotpoints* interessant, welcher die Anzahl der Stützstellen angibt und standardmäßig auf 50 gesetzt ist. Sämtliche berechneten Punkte werden intern mit */psline* verbunden, so dass sich eine zu geringe Anzahl an Stützstellen durch einen Polygonzug bemerkbar macht. Mit Werten um 200 liegt man in den meisten Fällen auf der richtigen Seite. In diesem Zusammenhang ist unbedingt zu beachten, dass geänderte Parameter solange ihren Wert behalten bis innerhalb eines L^AT_EX-Laufs eine neue Wertzuweisung erfolgt.

Der Variablenname für *psplot* ist per Definition *x* und für *parametricplot* *t*. Beide können nicht verändert werden, was jedoch bezüglich der Anwendung keinerlei Einschränkung darstellt. Die Variablen können beliebig oft innerhalb eines Ausdrucks verwendet werden, denn erst mit der schließenden Klammer für den Funktionsausdruck wird davon ausgegangen, dass die zweite Koordinate für einen Punkt des Graphen oben auf dem Stack liegt. Der einzige Unterschied zwischen diesen beiden Befehlen ist, dass bei *psplot* nur der oberste Stackwert (*y*) und bei *parametricplot* die obersten beiden Stackwerte (*x*; *y*) als Argumente verwendet werden.

Zu beachten ist unbedingt, dass mit den beiden Befehlen keine Fehlermeldungen ausgegeben werden, was insbesondere für diejenigen mathematischen Funktionen von Interesse ist, deren Definitionsbereich nicht ganz \mathcal{R} entspricht. Denn bei *einem* fehlerhaften Argument, z.B. $\sqrt{-1}$, wird der komplette Graph nicht gezeichnet!

Der Vollständigkeit halber wird noch erwähnt, dass *psplot* noch weitere drei Befehle zum Zeichnen von Funktionsgraphen zur Verfügung stellt, die jedoch alle externe Datensätze voraussetzen:


```
\fileplot[Parameter]{Dateiname}
\dataplot[Parameter]{Befehle}
\listplot[Parameter]{Liste}
```

(durch `<enter>` symbolisiert), bevor eine der mathematischen Funktionen aufgerufen wird. Die hier beschriebenen Befehle beziehen sich immer auf das oberste Stackelement oder die obersten beiden.

Die direkte Anwendung der PostScript-Befehle für die Darstellung mathematischer Zusammenhänge bringt gegenüber Programmen wie z.B. *gnu-plot* nicht unbedingt Vorteile in der endgültigen Druckausgabe. Auch lässt sich nicht unbedingt jedes mathematische Problem mit den PostScript-Befehlen einfach lösen. Vorteile ergeben sich durch die Übersichtlichkeit und vor allen Dingen in der Länge der Textdateien.

■ Anwendung mit pst-plot

Das umfangreiche Paket *pstricks* [*CTAN:/graphics/pstricks/*] ist nichts weiter als ein Frontend für (L^A)T_EX zu den PostScript-Funktionen. Das Paket *ps-plot* [*CTAN:/graphics/pstricks/latex/pst-plot.sty*], welches Teil des Pakets *pstricks* ist, ermöglicht mit zwei Befehlen die Anwendung der PostScript-Funktionen für das Zeichnen von mathematischen Funktionen. Mit der Anweisung `\usepackage{pst-plot}` wird automatisch das übergeordnete Paket *pstricks* geladen.

Die grundsätzliche Struktur der hier behandelten Befehle ist: 

```
\psplot[Parameter]{xmin}{xmax}{Funktion f(x)}
\parametricplot[Parameter]{tmin}{tmax}{Funktionen x(t), y(t)}
```

Hierin bedeutet $[x_{\min}, x_{\max}]$ bzw. $[t_{\min}, t_{\max}]$ das jeweilige Definitionsintervall (Start- bzw. Endwert). Die

Weitere Informationen kann man der Dokumentation zu *pstricks* entnehmen.²

PostScript, sehr häufig nur als Druckertreiber bekannt oder im Zusammenhang damit als Seitenbeschreibungssprache, ist ähnlich wie T_EX eine Programmiersprache, wobei PostScript eindeutig als sogenannte Hochsprache bezeichnet werden muss.¹ Während die Ursprünge der Entwicklung bis in die Anfänge der siebziger Jahre zurückgehen, datiert die erste veröffentlichte PostScript-Version von 1982, was nicht zufällig das Jahr der Gründung von Adobe System Incorporated ist. Der erste Laserdrucker mit vollständiger PostScript-Implementierung war der *Apple Laserwriter*. Der einzige Computer der PostScript auch für die Bildschirmausgabe (Display PostScript) verwendete, war *Next*.

In der letzten Ausgabe des APL-Journals wurde das Frontend L^AT_EX ausführlich behandelt, sodass die hier angegebene Beispiele, die sich auf das Textsatzsystem L^AT_EX beziehen, ebenfalls mit L^AT_EX bearbeitet werden können. Andererseits besteht für viele Textverarbeitungen die Möglichkeit des PostScript-Exports, was eine Anwendung der Befehle möglich macht.

■ PostScript-Befehle

PostScript arbeitet mit dem sogenannten Stacksystem, welches den Benutzern von HP-Taschenrechnern geläufig und auch unter dem Namen „UPN“, *Umgekehrte Polnische Notation* (Reverse Polish Notation) bekannt ist, und letztlich den internen Standard für alle Computer darstellt. Die normale Notation für die Multiplikation „*axb*“ wird zu „*a<enter>b<enter>x<enter>*“. Es sind immer zuerst die Parameter (Variablen) auf dem Stack abzulegen

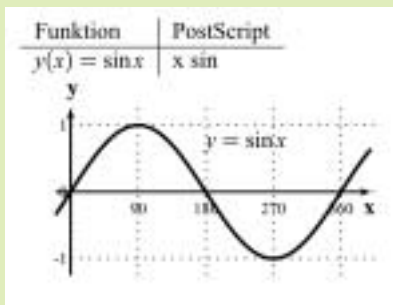
Beispiele für PostScriptplot

Für alle Beispiele wird jeweils die komplette pspicture-Umgebung angegeben, so dass eine direkte Übernahme

der Beispiele möglich ist. Eine Dokumentation des multido-Befehls findet man unter [CTAN: /macros/

latex209/contrib/multido/multido.doc], alle anderen in der Dokumentation zu pstricks.[2]

• Sinusfunktion



```
\psset{xunit=0.0111cm,yunit=1cm}
\begin{pspicture}(-20,-1.25)(400,1.25)
\psline[linewidth=1pt]{->}(-20,0)(400,0)
\psline[linewidth=1pt]{->}(0,-1.25)(0,1.25)
\multido{\n=90}{5}{\psline[linestyle=dotted] (\n,-1.25)(\n,1.25)\rput{\n,0.25}{\scriptsize \n}}
\multido{\n=-1+1}{3}{\psline[linestyle=dotted](0,\n) (405,\n)\rput[r]{-4,\n}{\scriptsize \n}}
\psplot[plotstyle=curve,linewidth=1.5pt]{-20}{400}{x sin}% postscript function
\rput[|]{-5,1.75}{\mathbf{f}(y)}
\rput[|]{390,-.25}{\mathbf{f}(x)}
\rput[|]{180,0.75}{\mathbf{y}=\sin x}
\end{pspicture}
```

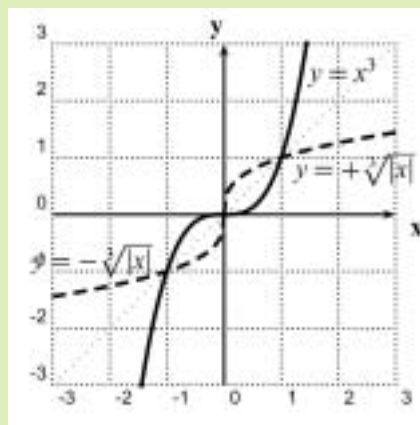
• Potenzfunktion

Die Parabel dritten Grades sowie ihre Umkehrfunktion, wobei die In-

tervallunterscheidung nicht zwingend ist, wenn man die Exponentialschreib-

weise mit $y = x^{\frac{1}{3}}$ wählt.

Funktion	PostScript
$y(x) = x^3$	<code>x 3 exp</code>
$y^{-1}(x) = \begin{cases} +\sqrt[3]{x} & x > 0 \\ -\sqrt[3]{x} & x < 0 \end{cases}$	<code>x 0.333 exp</code> (Umkehrfunktion)



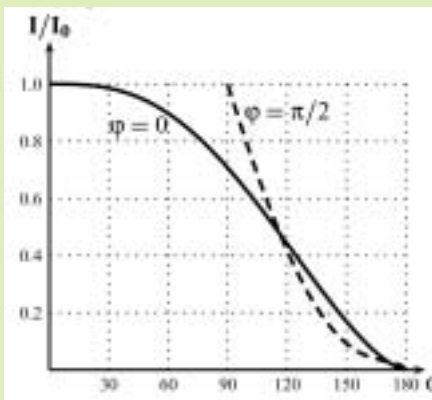
```
\begin{pspicture}(-3,-3)(3,3)
\psgrid[subgriddiv=1,griddots=10,gridlabels=7pt]{(-3,-3)(3,3)%
\psline[linewidth=1pt]{->}(-3,0)(+3,0)
\psline[linewidth=1pt]{->}(0,-3)(0,3)
\psline[linewidth=0.5pt,linestyle=dotted](-3,-3)(3,3)%
\psplot[plotstyle=curve,linewidth=1.5pt]{-1.4}{1.4}{x 3 exp}
\psplot[plotstyle=curve,linewidth=1.5pt,linestyle=dashed]{0}{3}{x 0.333 exp}
\psplot[plotstyle=curve,linewidth=1.5pt,linestyle=dashed]{-3}{0}{
x -1 mul 0.333 exp -1 mul}
\rput[|]{-.25,3.5}{\mathbf{f}(y)}
\rput[|]{3.5,-.25}{\mathbf{f}(x)}
\rput[|]{1.5,2.5}{\mathbf{y}=x^3}
\rput[|]{1.25,0.8}{\mathbf{y}=\sqrt[3]{|x|}}
\rput[r]{-1.25,-0.8}{\mathbf{y}=-\sqrt[3]{|x|}}
\end{pspicture}
```

• Beispiel aus der Leistungselektronik

Gezeigt wird die graphische Darstellung des relativen Strommittelwertes für eine Stromrichtersteuerung durch ein Thyristorpaar. Hierbei entspricht der Parameter phi der Phasenverschiebung zwischen Strom und Spannung. Die unabhängige Variable alpha bezeichnet den Steuerwinkel.

Zu beachten ist, dass PostScript die Argumente für die trigonometrischen Funktionen im Gradmaß erwartet, sodass für relative Winkel auf gleiche Einheiten zu achten ist. Der Ausdruck $\frac{\alpha}{\pi}$ ist daher durch $\frac{\alpha}{180}$ zu ersetzen.

Funktion	$\frac{I(\alpha)}{I_0} = \begin{cases} \sqrt{1 - \frac{\alpha}{\pi} + \frac{1}{2\pi} \sin 2\alpha} & \varphi = 0 \\ \sqrt{2 \left(1 - \frac{\alpha}{\pi}\right) (2 + \cos 2\alpha) + \frac{3}{\pi} \sin 2\alpha} & \varphi = \frac{\pi}{2} \end{cases}$
PostScript	<code>1 x 180 div sub 1 6.28 div x 2 mul sin mul add sqrt</code> $\varphi = 0$ <code>2 x 90 div sub x 2 mul cos 2 add mul x 2 mul sin 3</code> $\varphi = \frac{\pi}{2}$ <code>3.15 div mul add sqrt</code>



```
\psset{xunit=0.0333cm,yunit=3cm}
\begin{pspicture}(0,-0.25)(190,1.25)
\psline[linewidth=1pt]{->}(0,0)(190,0)
\psline[linewidth=1pt]{->}(0,0)(0,1.15)
\multido{\n=30+30}{6}{\psline[linestyle=dotted](\n,0)\rput{\n,0.05}{\scriptsize \n}}
\multido{\n=0.2+0.2}{5}{\psline[linestyle=dotted](0,\n)\rput[r]{-4,\n}{\scriptsize \n}}
\psplot[plotstyle=curve,linewidth=1.5pt]{0}{180}{
1 x 180 div sub 1 6.28 div x 2 mul sin mul add sqrt}
\psplot[plotstyle=curve,linewidth=1.5pt,linestyle=dashed]{0}{180}{
2 x 90 div sub x 2 mul cos 2 add mul x 2 mul sin 3
2 add mul x 2 mul sin 3 3.15 div mul add sqrt}
\rput(0,1.2){\mathbf{f}(I/I_0)}
\rput[|](190,-.05){\mathbf{f}(\alpha)}
\rput(45,0.85){\mathbf{\varphi}=0}
\rput(120,0.9){\mathbf{\varphi}=\pi/2}
\end{pspicture}
```

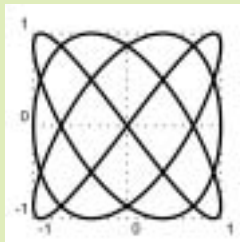
Beispiele für \parametricplot

• Lissajous Figur

Die aus der Physik oder Elektrotechnik bekannten Lissajouffiguren sind ein typischer Anwendungsfall für Gleichungen in Parameterform. Die hier angegebene Darstellung beruht auf den Funktionen:

Funktion	PostScript
$x = \sin 1.5t$	<code>t 1.5 mul sin</code>
$\sin(2t + \frac{\pi}{3})$	<code>t 2 mul 60 add sin</code>

Aufgrund der „Länge“ des Graphen wurde der Wert für `plotpoints` auf 200 gesetzt, so dass auch die „Ecken“ des Graphen mit starker Krümmung kontinuierlich erscheinen.



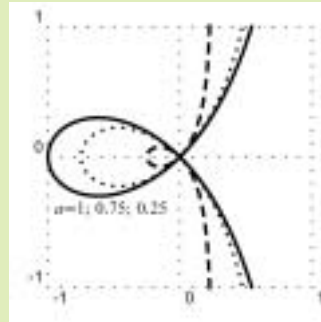
```
\psset{xunit=1.5cm,yunit=1.5cm}
\begin{pspicture}(-1,-1)(1,1)
\psgrid[subgriddiv=0.5,griddots=10,gridlabels=7pt](-1,-1)(1,1)
\parametricplot[plotstyle=curve,linewidth=1.5pt,%
plotpoints=200](-360){360}{t 1.5 mul sin t 2 mul 60 add sin}
\end{pspicture}
```

• Strophoide

Zum Schluss wird eine „Strophoide“ dargestellt, die durch folgende Bezie-

hungen gegeben ist, wobei α durch einen Zahlenwert zu ersetzen ist:

Funktion	PostScript
$x(t) = \frac{a(t^2 - 1)}{t^2 + 1}$	<code>t t mul 1 sub a mul t t mul 1 add div</code>
$y(t) = \frac{at(t^2 - 1)}{t^2 + 1}$	<code>t t mul 1 sub t mul a mul t t mul 1 add div</code>



```
\psset{xunit=2cm,yunit=2cm}
\begin{pspicture}(-1.25,-1.25)(1.25,1.25)
\psgrid[subgriddiv=1,griddots=10,gridlabels=7pt](-1,-1)(1,1)
\parametricplot[plotstyle=curve,plotpoints=200,linewidth=1.5pt](-1.85){1.85}%
{t t mul 1 sub t t mul 1 add div t t mul 1 sub t mul t t mul 1 add div}
\parametricplot[plotstyle=curve,plotpoints=200,linewidth=1.5pt,linestyle=dashed](-5){5}%
{t t mul 1 sub 0.25 mul t t mul 1 add div t t mul 1 sub t mul 0.25 mul t t mul 1 add div}
\parametricplot[plotstyle=curve,plotpoints=200,linewidth=1.5pt,linestyle=dashed](-1){1}%
{t t mul 1 sub 0.75 mul t t mul 1 add div t t mul 1 sub t mul 0.75 mul t t mul 1 add div}
\rput[!](-0.95,-0.4){\scriptstyle a=1:\ 0.75:\ 0.25}
\end{pspicture}
```

Zusammenstellung der mathematischen PostScript-Funktionen

Die folgende tabellarische Zusammenstellung enthält bis auf die Matrixbefehle alle mathematischen Funktionen mit ihren Eigenschaften. Hierin

sind `<int>` und `<real>` selbsterklärend, während für `<any>` jeder beliebige Typ und für `<num>` real oder integer gesetzt werden kann. Unabhängig von

der Null gelten für Zahlen unter PostScript die folgenden Grenzen, wobei die Angaben für real betragsmäßig zu verstehen sind:

Name	Bedeutung	Anwendung	Beispiel
abs	Absolutwert	<code><num> abs</code>	<code>-3 abs</code> → 3
add	Addition ¹	<code><real1> <real2> add</code>	<code>5 7 add</code> → 12
atan	Arcus Tangens ²	<code><real1> <real2> atan</code>	<code>2 45 atan</code> → 2.54
cos	Cosinus ³	<code><real> cos</code>	<code>60 cos</code> → 0.5
cvi	Real Integer	<code><real> cvi</code>	<code>14.13 cvi</code> → 14
cvr	Integer Real	<code><int> cvr</code>	<code>14 cvr</code> → 14.00
div	Division ⁴	<code><real1> <real2> div</code>	<code>100 8 div</code> → 12.5
dup	Dupliziere oberstes Stackelement	<code><any> dup</code>	<code>12 dup</code> → 12 12
exch	Exchange ⁵	<code><any1> <any2> exch</code>	<code>12 13 exch</code> → 13 12
exp	Potenz	<code><real1> <real2> exp</code>	<code>3 4 exp</code> → 81.0
idiv	ganzzahlige Division	<code><int1> <int2> div</code>	<code>100 8 div</code> → 12
ln	natürlicher Logarithmus	<code><real> ln</code>	<code>12 ln</code> → 2.48491
log	Zehner-Logarithmus	<code><real> log</code>	<code>1000 log</code> → 3.00
mod	Modulo	<code><int1> <int2> mod</code>	<code>5 3 mod</code> → 2
mul	Multiplikation ¹	<code><real1> <real2> mul</code>	<code>5 3 mul</code> → 15
neg	Negiere Vorzeichen	<code><num> neg</code>	<code>5 neg</code> → 5
round	Runden	<code><real> round</code>	<code>5.7 round</code> → 6
sin	Sinus ³	<code><real> sin</code>	<code>30 sin</code> → 0.5
sqrt	Quadratwurzel	<code><real> sqrt</code>	<code>16 sqrt</code> → 4.0
sub	Subtraktion ¹	<code><real1> <real2> sub</code>	<code>17 19 sub</code> → 2
truncate	Dezimalteil abtrennen ⁶	<code><real> truncate</code>	<code>-33.33</code> → 33.00

	integer	real
kleinster Wert	-2^{32}	$\pm 10^{-35}$
größter Wert	$2^{32} - 1$	$\pm 10^{38}$

Literatur

- Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. IWT, Vaterstetten, 1989.
- Timothy Van Zandt. *PSTricks - PostScript macros for Generic TeX*. <http://www.tug.org/application/PSTricks>, 1993.

Kontakt

Herbert Voß, Berlin
 eMail:voss@lyx.org

Anmerkungen:

- sind beide Argumente ganze Zahlen ist auch das Ergebnis vom Typ Integer
- entspricht der Darstellung $\alpha = \arctan \frac{\langle \text{real1} \rangle}{\langle \text{real2} \rangle}$
- das Argument wird im Gradmaß erwartet
- das Ergebnis ist grundsätzlich vom Typ real
- die Anordnung der letzten beiden Stackelemente wird vertauscht
- das Ergebnis bleibt vom Typ real

Tabelle 1: Übersicht der mathematischen Postscript-Funktionen

Reiner Nussbaum

Linux und APL

APL kann für systemnahe Aufgaben unter LINUX verwendet werden

Unter LINUX kann APL auch für systemnahe Arbeiten herangezogen werden. So kann man beispielsweise Postscript-Daten in PDF-Daten (Acrobat-Reader-Formate) konvertieren. In diesem Beitrag wird vorgeführt, wie man LINUX in ein Computernetzwerk einbindet und anschliessend mit sehr einfachen Mitteln Informationen von nahezu beliebigen Quellen verarbeiten und an beliebige Ziele verteilen kann. Als technische Schnittstellen werden Datei-/Verzeichnis- Freigaben sowie Standard-Druck-Schnittstellen benutzt. Zu den vorgestellten Möglichkeiten zählen das Auslesen von Verzeichnissen, die Unterscheidung zwischen Verzeichnissen und Dateien, das Erkennen, ob Dateien in Benutzung sind, das Verschieben von Dateien sowie das Konvertieren von Dateien in das PDF-Format.

keit, die zu verarbeitenden Daten nach Unterverzeichnissen weiter zu ordnen. Die zweite Möglichkeit (2) besteht darin, Unterverzeichnisse der Anwendung freizugeben. Damit können verschiedene Datenlieferanten voneinander getrennt werden.

2. Erzeugen von PDF Dateien

Für das Erzeugen von PDF-Dateien sind nur wenige Schritte erforderlich.

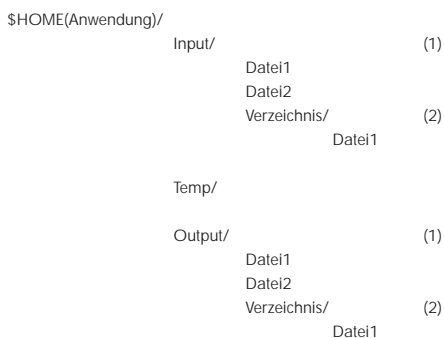
Der Datenlieferant muß die Dateien im Postscript-Format erstellen. Unter LINUX ist dies ohnehin das Standardformat. Windows-Benutzer können Postscript-Dateien einfach über die Druckerschnittstelle erzeugen. Es wird ein Postscript Drucker unter Windows definiert, welcher die Ausgabe in eine Datei erstellt.

Die Postscript-Datei wird in das entsprechende Eingabeverzeichnis der LINUX-APL-Anwendung gestellt. Für Windows-Computer bedeutet dies

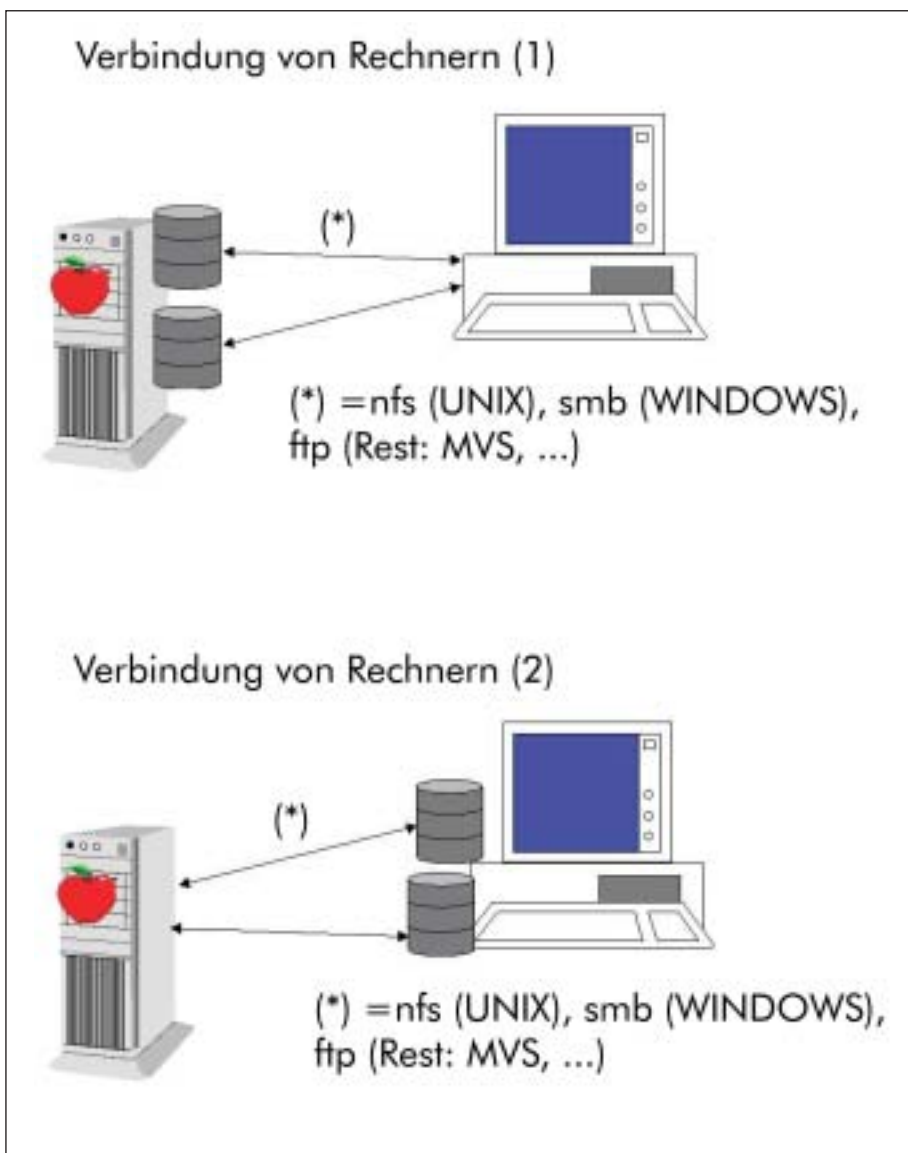
1. Verzeichnisstruktur

Aus der Sicht der LINUX-Anwendung wird von einem eigenen Stammverzeichnis ausgegangen (HOME Directory). Dieses kann Dateien oder Verzeichnisse beinhalten. Für die Beispielanwendung gehen wir davon aus, dass die Unterverzeichnisse nur Dateien enthalten sollen (Unterverzeichnisse darin werden ignoriert).

Nachfolgende Abbildung zeigt die Verzeichnisstruktur aus Sicht der Anwendung.



Wenn die Daten, welche verarbeitet werden sollen von 'außen' angeliefert werden, gibt es zwei Möglichkeiten der Vorgehensweise: zum Einen kann das Stammverzeichnis der Anwendung (1) freigegeben werden. In diesem Fall hat der Datenlieferant die gleiche Sicht auf die Verzeichnisstruktur wie die LINUX Anwendung. Dies bietet die Möglich-



beispielsweise, daß die Postscript-Datei auf ein Netzlauferk kopiert wird; unter LINUX ist es das Kopieren einer Datei in ein 'gemountetes remote' Verzeichnis.

Eine benutzerfreundlichere Alternative ist auch das Bereitstellen eines freigegebenen Postscript-Druckers, dessen Warteschlange dann von der APL-Anwendung bearbeitet wird. Dann heisst die generelle Benutzerschnittstelle nur: Nimm irgendeine Anwendung, lade die entsprechenden Daten und Drucke auf den LINUX-Drucker. Das Ergebnis wird als PDF-Datei von LINUX APL bereitgestellt.

Für LINUX sind im wesentlichen folgende Schritte erforderlich:

- warten auf Arbeit (Ist eine Datei vorhanden?)
- auswerten von Directory Informationen (Welche Verzeichnisse gibt es, sind Dateien vorhanden?)
- ist eine Datei, welche verarbeitet werden soll 'frei'? (nicht von anderen benutzt)
- verschieben der Postscript-Datei in ein temporäres Verzeichnis
- erzeugen der PDF-Datei (ps2pdf)
- verschieben der PDF-Datei in das Ausgangsverzeichnis (ggf. Unterverzeichnis davon)
- löschen der Datei im temporären Verzeichnis

3. Die APL-Anwendung

Bei den hier besprochenen Programmteilen beschränken wir uns auf das Verarbeiten von Druckerwarteschlangen (weil das der benutzerfreundlichste Ansatz ist). In Erweiterung der zuvor beschriebenen Verarbeitung kommt hinzu, dass innerhalb der Druckerwarteschlange nur Dateien, welche mit 'df' beginnen, zu verarbeiten sind. Weil die Dateinamen nicht sehr aussagekräftig sind, erfolgt zusätzlich die Prüfung innerhalb der Postscript-Datei, ob von der Anwendung ein Dateiname mitgeliefert wird; im positiven Fall kann die Ausgabedatei in einen benutzerfreundlichen Namen umbenannt werden. Falls -aus LINUX-Sicht- das Ausgabeverzeichnis in einem freigegebenen Verzeichnis des Senders liegt, so geschieht aus Sicht des Benutzers folgendes:

Es wird ein Druckauftrag aus irgendeiner Anwendung erteilt. Als Ergebnis des Druckauftrages erscheint eine PDF-Datei auf dem Computer des Benutzers.

3.1 Die Hauptsteuerung

Die Hauptsteuerung ist sehr kurz:

Hauptsteuerung

```
1⊞fd gtΔpslp2pdfstart
  ∇ gtΔpslp2pdfstart;z;dir;⊞io;cr;file;dir_in;dir_zwi;dir_out;subdir;dir2
[1] gtΔpslp2pdfinit a initialize stuff
[2] lp:gtΔpslp2pdfprocess a doall
[3] sleep:z+⊞dl 5⊞→lp a wait for work
[4] a 2001NOV-rmuss: postscript to pdf from a (LINUX) printer queue
  ∇
```

3.2 Weitere Funktionen

Initialisierung

```
∇ gtΔpslp2pdfinit
[1] z+⊞svn P⊞runs
[2] ⊞io+1⊞cr+⊞av[11]
[3] dir_in+ /usr/gt_ps2pdf/pslp_in a link to printer queue
[4] dir_zwi+ /usr/gt_ps2pdf/pslp2pdf_zwi a temp storage
[5] dir_out+ /usr/gt_ps2pdf/pslp2pdf_out a where to put .pdf files
[6] a 2001NOV-rmuss: applications home directory: /usr/gt_ps2pdf
  ∇
```

Verarbeiten der Dateien

```
∇ gtΔpslp2pdfprocess subdir;dir;x;z;file;ofile;fd;in_p;out_p
[1] in_p+dir_in,subdir ⊞ out_p+dir_out,subdir a path information
[2] dir+host ls -l ,in_p, /*
[3] lp:→(0=Pdir)/0 a process all entries
[4] file+(~1+z+dir\cr)dir ⊞ dir+z+dir a directory entry
[5] →(d =1↑file)/Lsubdir,Lfile
[6] →lp a nothing of interest
[7] Lfile:file+(1-(⊞file)1 / )↑file a dealing with a file
[8] →(~^/ df =2↑file)/lp a only df files are of interest
[9] pslp2pdf: processing file: ,file a tell what we re doing
[10] →(0 Phost sudo fuser ,dir_in, / ,file)/lp a file in use?
[11] x+host mv ,in_p, / ,file, ,dir_zwi, / ,file a move to temp
[12] ⊞(0<P,subdir)/ x +host mkdir ,out_p a to be sure
[13] x+host ps2pdf ,dir_zwi, / ,file, ,out_p, / ,file, .pdf
[14] ⊞(0=Pofile+gtΔpslp2pdfmkofile file)/ ofile+file a increase comfort
[15] →(file ofile)/dlt_tmp a backup old output if already present
[16] →(0=fd+gtΔfopen_ro out_p, / ,ofile, .pdf)/newname
[17] x+gtΔfclose fd
[18] x+host mv ,out_p, / ,ofile, .pdf ,out_p, / ,ofile, .pdf.bak
[19] newname: x+host mv ,out_p, / ,file, .pdf ,out_p, / ,ofile, .pdf
[20] dlt_tmp: x+host rm ,dir_zwi, / ,file a inputfile+nrwana
[21] →lp a next directory entry
[22] Lsubdir:→lp a subdirectories are not supported for printer queues
[23] a 2001NOV-rmuss: produce .pdf files from a postscript printer queue
  ∇
```

Ermitteln des Namens der Ausgabedatei

```
∇ ofile+gtΔpslp2pdfmkofile file;fd;n;i;rec
[1] a read postscript input to determine original filename
[2] a expect to find keyword %%Title: within the 1st 10 records
[3] ofile+file
[4] →(0=fd+gtΔfopen_ro dir_zwi, / ,file)/exit a should never happen
[5] n+10 ⊞ i + 0
[6] lp: →(0 rec+gtΔfgets fd)/elp
[7] →(~ %%Title: 9↑rec)/next
[8] ofile+~1+9↑rec a found document name (no header, no ending line feed)
[9] →exit
[10] next:→(0 gtΔfeof fd)/elp
[11] →(n>i+1)/lp
[12] elp: i+gtΔfclose fd
[13] exit:→0 a nothing to do any more
[14] a 2001-rmuss: find original doc-name from postscript file
  ∇
```

Zusammenfassung / Ausblick

APL kann auch für systemnahe Aufgaben unter LINUX verwendet werden. Hervorzuheben ist, dass dem Anwender nur die notwendigen Privilegien zu erteilen sind. Für das Verarbeiten der Druckerwarteschlange reicht es aus, die Anwendung in die Gruppe 'lp' (line print) aufzunehmen. Dadurch sind der Zugriff auf die Warteschlange sowie die für das Verarbeiten erforderlichen Rechte erteilt. Den privilegierten Zugriff auf das Prüfen von geöffneten Dateien kann man auf die LINUX-Funktion 'fuser' beschränken. Das Verteilen der PDF Dateien in die Ausgabeverzeichnisse wird mit den normalen Verzeichnis- und Datei- Berechtigungen geregelt.

Wenn man darüber hinaus noch berücksichtigt, dass die Funktionalität 'list directory' sowie 'move Datei' (=Ko-

pieren und Löschen nach Kopieren) auch mittels ftp realisierbar sind, so ist auch die Erweiterung auf andere Plattformen gegeben. Beispielsweise ist es möglich, die Daten von einem MVS-Host abzuholen und wie dargestellt zu verarbeiten.

Auch jede andere Form der Verarbeitung kann leicht realisiert werden: beispielsweise können auch HTML-Seiten (mit Seitenindex) erstellt werden, wenn man in der Anwendung die weitere Verarbeitung mittels 'pdftohtml' anwendet. Genauso einfach ist auch eine Verarbeitung wie das Komprimieren oder Dekomprimieren von Daten, was insbesondere für die Weitergabe von Informationen an Anwendungen auf unterschiedlichen Plattformen Vereinfachungen bietet.

Interessant ist auch ein Ansatz, bei dem nicht 'Daten' als Eingabe geliefert werden, sondern 'Auftragsinformatio-

nen'. Beispielsweise könnten Aufträge wie folgt verarbeitet werden: Hole von HOST A eine Datei, erzeuge eine PDF-Datei daraus und schicke das Ergebnis per e-mail an B.

Nicht zuletzt kann man aus sehr kurzen APL-Funktionen auch kurze C-Programme erstellen, so dass ein APL-Interpreter in einer LINUX-Produktionsumgebung nicht erforderlich ist.

Kontakt

Reiner Nussbaum,
eMail: reiner.nussbaum@web.de

AFAPL - Association Francophone pour la promotion du langage APL

Revue APL
Habits de [APL]
per S. Baron
Bureau 2000
G. Langlet
Quelques Photos
Liens
A visiter
Remerciements

! T-shirt AFAPL !
(à F. Jean-Loupot)
Contactez Françoise

AFAPL
Association Francophone pour la promotion du langage APL
Association régie par la loi de 1901

AFAPL
174 bd de Charonne
75020 PARIS
FRANCE
Tél: (33) 1 43 66 31 79
Fax: (33) 1 43 66 31 79

Nous contacter :
afapl@afapl.asso.fr
Lemagnen@ga1.com

Bulletin d'adhésion

(Dernière mise à jour : 12 Mars 2001)

<http://www.afapl.asso.fr/>

<http://www.afapl.asso.fr/> (11/01/01 00:54:52)

Deutschsprachige J-Schulungen

Das J-Support-Team der Gaertner Datensysteme wird ab 2002 jeden Monat eine kostenfreie, einstündige J-Online-Beratung durchführen.

Gaertner Datensysteme plant ferner, Anfang 2002 eine ca. dreitägige, deutschsprachige Intensiv-Schulung für J durchzuführen, sofern ein hinreichendes Interesse dafür besteht.

Kontakt: j-support@gaertner.de

Klaus-Peter Friedrich

A.ID.A - das Analysesystem der RSGV-Informationendienste in APL2

Der Fachbereich Informationsdienste beim Rheinischen Sparkassen- und Giroverband (RSGV), Düsseldorf, versorgt Sparkassen, RSGV-Fachabteilungen, Verbandsleitung, Rechenzentrum, Gremien und Verbundpartner mit geschäftspolitisch relevanten Informationen. Hierzu hat der RSGV auf der Basis von IBM-APL2 ein umfassendes Datenbanksystem mit einer Vielzahl von Werkzeugen geschaffen, die sowohl auf dem Großrechner als auch auf PC angeboten werden.

Das Sparkassen-Time-Sharing (STS/PSTS)-System umfaßt:

- Ca. 300.000 Zeitreihen für Betriebsvergleiche, Statistiken, Konkurrenzvergleich, Volkswirtschaft, Konditionen, Prognosen, Regionaldaten in mehrdimensionalen Datenbanken
- Ca. 1000 Auswertungsstandards
- Individuelle, mehrdimensionale Datenbanken
- Mächtige und flexible Abfrage- und Analysewerkzeuge zur individuellen Nutzung
- Entwicklungswerkzeuge für Sparkassen
- Schnittstellen zur operativen DV und Datentransfer/Konvertierung
- Spezielle betriebswirtschaftliche Analyseverfahren
- Backoffice-Unterstützung für betriebswirtschaftliche Beratungsprojekte

Die veraltete Benutzeroberfläche (TSO bzw. DOS) und das umständliche Verfahren zum Austausch von Daten und Programmen zwischen Großrechner und PC (Filetransfer) beeinträchtigen mittlerweile jedoch die STS/PSTS-Anwendungen. Es musste ein Verfahren gefunden werden, mit dem folgende Ziele erreicht werden:

- Moderne, benutzerfreundliche Oberfläche
- Investitionsschutz (APL2-Know-How der Mitarbeiter, Anwendungen, Verfahren)
- Keine Verzettelung durch verschiedene APL-Dialekte, keine anderen Sprachen
- Nutzung vorhandener Ressourcen
- Optimale Kommunikation zwischen Client und Host
- Einheitliche, umfassende Nutzungsmöglichkeit sowohl lokal als auch

bei Netz- und Server(Host) -Anbindung (Mitarbeiter im Außendienst, Telearbeiter)

- Geringe Netzbelastung
- Hohe Sicherheit

Der RSGV hat bereits vor einiger Zeit begonnen, seine Anwendungen zu modernisieren (Arbeitstitel A.ID.A), ist jedoch nicht so richtig vorangekommen. Zum Beispiel wurde viel Zeit in Browser-Technologie und JAVA investiert. Unsere technischen Anforderungen (Entwicklung von Werkzeugen für die individuelle Datenverarbeitung) konnten damit jedoch nicht zufriedenstellend realisiert werden. Letztendlich sind wir durch Gespräche und Hinweise von Kollegen aus dem APL-Guide zu der Überzeugung gekommen, dass wir mit dem von IBM gelieferten Konzept der ‚Cross-System Shared Variables‘ sehr gut auskommen. Dabei kommt uns zugute, dass wir ohnehin vorhatten, ausschließlich IBM-APL2 sowohl unter TSO als auch unter Windows zu nutzen und wir dadurch auf keine Konvertierungen wie z.B. über HTML angewiesen sind.

■ Die Lösung

Die Erstinstallation von A.ID.A erfolgt mittels eines Lotus Notes-Dokumentes, das die jeweils aktuellen Systemdateien enthält, die in ein beliebig platzierbares Verzeichnis ‚AIDA‘ zu kopieren sind. Von hier aus erfolgt die weitere Installation mit Hilfe eines kleinen paketierten APL2-Arbeitsbereiches.

Voraussetzung auf dem Client ist das IBM-APL2-System in der Runtime- oder Vollversion (für Entwickler bzw. zur Fehleranalyse).

Die Verarbeitung erfolgt grundsätzlich lokal. Es werden kaum Grossrechner-Ressourcen benötigt.

APL-Funktionen und Programm-Variablen werden bis auf einen ganz kleinen Start-APL-Arbeitsbereich in Filehandlings-Dateien auf dem Client vorgehalten, die bei einer Anmeldung am Server(Host) automatisch auf den neuesten Stand gebracht werden. Durch APL-Code ist nur eine geringe Übertragungskapazität erforderlich.

Beim Start von A.ID.A wird ein Arbeitsbereich AIDA.APL (bzw. AIDA.DLL bei einer Runtime-Installation) geladen. Damit werden die für die jeweilige Anwendung benötigten Funktionen und Variablen aus den Filehandlings-Dateien gelesen und ausgeführt.

Aufteilung der n-dimensionalen Daten-Dateien (DASY-Dateien) in einen variablen Strukturteil und einen Datenteil mit fester Satzlänge. Dabei werden die Strukturdateien, die teilweise relativ groß sind aber nur selten geändert werden (Ausnahme Zeitachse), soweit vom jeweiligen Benutzer benötigt, auf dem Client vorgehalten. Durch ein ausgefeiltes Versionsmanagement werden bei Bedarf (Zugriff auf eine Server-Datei) nur die unbedingt nötigen Strukturdaten (z. B. Zeitachse) übertragen.

Die DASY-Dateien des Verbandes werden grundsätzlich nur auf dem Server vorgehalten. Dies bringt große Vorteile beim Zugriffsschutz sowie der Aktualität der Daten und ist wegen des großen Datenvolumens ohnehin erforderlich. Die Auswahl der benötigten Daten erfolgt ausschließlich auf dem Client (nach eventuellem Update der Strukturdateien). Danach werden nur die benötigten Indizes zum Server übertragen. Hier erfolgt die eigentliche Selektion der rein numerischen Daten und soweit möglich eine Vorverdichtung der Daten (Verbandssumme, Mittelwerte etc.). Danach gelangen die Daten über die ‚APL2 across System‘-Variable zum Client, wo die weitere Verarbeitung stattfindet.

Nachdem IBM auch die GUI-Funktionen im APL2 für Windows laufend verbessert hat und wir uns entsprechendes Know-how angeeignet haben, kommen wir nun zügig voran und sind sicher, den für uns richtigen Weg gefunden zu haben.

Kontakt

Klaus-PeterFriedrich, RSGV, eMail:
klaus-peter.friedrich@rsgv.de

David Liebttag/Ursula Holzer

APL2 Mainframe Features

Why Mainframe Features? We often find customers simply don't know about all the features of APL2. Many features are documented in books that most users do not own. Some features have been added and the only documentation is in the PTF. This article gives you an overview of all non-interpreter facilities and enhancements made since last release.

APL2 Language and Development

Environment Enhancements

Unknown system names in function program definitions *
 Lower case characters in system names *
)EDITOR 2 PF key 12 Abandons editing *

- Supporting unknown system names make it easier to port from other dialects.
- Lower case characters aid usability.
-)Editor 2 PF12 was a GSE requirement.
- Enhancements are marked with asterisks (*).

i Unter Systemname werden Systemvariable und -funktionen verstanden. Es werden Systemvariablen und -funktionen akzeptiert, welche dem APL2 unbekannt sind. Damit ist es möglich, mit einem)IN Programme zu migrieren, die in APL2000 geschrieben sind. Mit einem)Copy werden die unbekannt Systemnamen nach wie vor abgewiesen.

APL2 System Features

Workspaces can be saved in VSAM or QSAM libraries
 Permanent session log can be saved in VSAM cluster
 National language messages can be tailored
 APL2 Application Environment can be used in production applications which don't require development tools.

- Miscellaneous features of the APL2 system

- Consult System Services Reference for info on defining VSM cluster
- People can change the system's messages or add their own AE is just mentioned for completeness.

i Die Fehlermeldungen stehen im PDS-Bestand, dessen last level qualifier SAP2LANG ist. Für jede Sprache gibt es ein Member. In dem entsprechenden Member können die Fehlermeldungen verändert, oder hinzu gefügt werden. Die Vorgehensweise ist in den einzelnen Members beschrieben.

Mit der externen Routine MSG können die einzelnen Nachrichten aufgerufen bzw angezeigt werden.

Public Workspaces

APLDATA	MEDIT
CHARTX	NETTOOLS *
CMS	PRINTWS
DISPLAY	SQL
EXAMPLES	SUPPLIED
FILESERV	TRANSFER
FSC126	TSO
FSM	UTILITY
GDMX	VAPLFILE
GRAPHPAK	VSAMDATA
MATHFNS	WSINFO

- Note: NETTOOLS is new and provides HTTP web server.

Shared Variable Processor

Supports communication between session and:

- Auxiliary Processors
- Users on same machine
- Users on TCP/IP network

- Overview of SVP to introduce list of auxiliary processors that follows.

- Point out cross system shared variables work across networks.

i Lokale Variable sind im Adressraum des Benutzers.

Globale Variable sind in einem eigenen Adressraum - Global Shared Memory. Global Shared Variable Prozessor (GSVP) ist ein OS/390 Subsystem, es ist eine eigene Autorisierung erforderlich]

Auxiliary Processors

- AP 100 Host
- AP 101 Stack
- AP 102 Storage
- AP 110 CMS Files
- AP 111 QSAM Files
- AP 119 TCP/IP
- AP 120 Session
- AP 121 APL2 Files
- AP 123 VSAM Files
- AP 124 Text Display
- AP 126 GDDM
- AP 127 SQL
- AP 210 BDAM Files
- AP 211 Object Files
- APL2 Shared Variable Interpreter Interface

- List of all the AP. Following slides will discuss APs which have been enhanced.
- Mention that Shared Variable Interpreter allows you to completely control interpreter including all system commands.

APL2 Shared Variable Interface:

- kontrolliert eine zweite APL-Session
- In/Output statt über Terminal oder files
- Shared Variables des rufenden Programmes
- Aufruf: SMAPL nn (nn=Prozessornummer)
- RAPL2 Remote

AP 100 Enhancements

Y2K Enhancements in)DDI,)DSI, and)PDSI
)DSI supports larger dataset lists

- Y2k enhancements support 4 digit years

- Dataset lists are generated by asking for all datasets under profile prefix.

i Das Creation Date z.B. hat früher eine dreistellige Tagesnummer angezeigt. Jetzt sind es TTund MM.

AP 119 Enhancements

Rewrite for OS/390 5.2
 Toleration of 1-element vectors in arguments
 Toleration of long userids in profiles
 Integer form of SELECT, compatibility with workstation APL2
 Accept SOCKET parameter STREAM, compatibility with Lingo Allegro
 SETSOCKOPT and GETSOCKOPT options
 KEEPALIVE, DEBUG, RCVBUF, others
 GIVESOCKET, TAKESOCKET, GETCLIENTID commands

- GIVESOCKET and TAKESOCKET allow master web server to distribute clients to slave servers.
- In the original AP 119, you had to provide boolean vectors to control which sockets would be waited on. For example, if you wanted to wait on sockets 1 3 and 4, you would give the vector: 1 0 1 1. It has 1's in the positions of the sockets to wait on. This was the Boolean form of select.
- The integer form of select lets you simply give a list of the socket numbers. Like this: 1 3 4.
- The original form of the socket command did not let you specify the stream type. It was like this:
- SV119 is 'TCPIP' 'SOCKET'
- Lingo Allegro built a version of AP 119 which supported a socket command like this:
- SV119 is 'TCPIP' 'SOCKET' type
- Where type could be STREAM or any of several other socket types.
- IBM's AP 119 always defaults to stream type. To aid compatibility with Lingo Allegro's version, we now tolerate the optional STREAM parameter.

AP 127 Enhancements

ISOL(UR) and ISOL(RS)

i Gibt das Cursorverhalten bei konkurrierend en Zugriffen auf Tabellen an

Associated Processors

Provide access outside the active workspace using APL2 syntax.

Processor 10 Rexx interface, deltaEXEC, and file IO tools deltaF, deltaFV, deltaFM

Processor 11 Non-APL external functions and namespaces

Processor 12 Access to APL2 and system files

Format '' defaults to 'GO 1 * C1 1 *' for system files *

Multiple volume support (Striping) for system files *

- This slider introduces the next make section of the presentation.
- Most of the reset of the presentation will be able Processor 11.
- DelF, DelFV, and DelFM are available on all platforms and provide easy portable file IO.
- Format '' supported for usability and compatibility with workstation APL2.
- Striping provides better performance; sequential records are written in different disks.

i Prozessor 11: Externe Routinen, Nicht-APL Routinen
 Prozessor 12: Externe Bestände werden wie APL-Variable behandelt

Supplied External Functions

System Facilities
 System Integration
 System Commands
 Communication Tools
 Packaged Public Workspace Functions
 Data Processing
 Namespace creation and management
 Converting Data in APL2 internal format
 Calling APL2 from other languages
 External module management
 Data Window Callable Services interfaces

- APL2 includes a wide variety and a large number of external functions that can be accessed through processor 11. This slider introduces the categories I used to make it easier to talk about them.

System Facilities

FED	Retrieve)MORE messages
OPTION	Set and query APL2 invocation options
	CASE
	DBCS
	DEBUG
	QUIET
	SYSDEBUG
	TRACE
PBS	Set or query)PBS setting
TIME	Performance analysis

- These functions provide access to APL2 system information.
- Tell them, when a message is followed by a plus sign, (+), they can type)MORE to get more information. FED can be used to get these „queued“ messages. (Die 10 letzten Fehlermeldungen)
- OPTION can be used to use authorized tools (QuadIB) in unauthorized session from locked functions.
- QuadIB is an undocumented system function which provides access to a lot of system internals. Many customers are familiar with some of its basic functions. Most importantly, you can use it to unlock locked functions. You can only use QuadIB if you are authorized. You become authorized by using the APL2 SYSDEBUG(128) invocation option. People usually do not run with SYSDEBUG(128). You can use the OPTION external function to temporarily change the SYSDEBUG setting. You might want to do this to use QuadIB to get the definition of a locked function in a workspace documentation tool.

System Integration

ATTN	Set or query attention bit
MSG	Display, queue, retrieve, and heck for APL2 messages
HELP	Retrieve keys text or list of keys

- Using ATTN you can use the QuadFX ignore attention property and still react to user attentions.
- Using MSG and HELP you can put your messages and online documentation in files so they can be changed without modifying the workspace.

- You might want to use ATTN if you have a program which needed to do some termination processing before halting. For example, if you had a program that was doing network communication, you would not want to just interrupt and)Reset the program. You would want the program to detect the user was trying to interrupt so it could send messages to the network partner telling it the connection was about to be broken. It could be fixed using the Ignore Attention property and use ATTN to poll whether the user was trying to interrupt.
- The message files can be found in the dataset allocated to ddname APL2LANG. We ship this dataset with a last qualifier of SAP2LANG. The exact dataset name is controlled by whoever installs APL2.
- The help files can be found in the dataset allocated to ddname APL2HELP. We ship this dataset with a last qualifier of SAP2HELP. The exact dataset name is controlled by whoever installs APL2.

System Commands

```
EDITORX )Editor xxx, non-APL editor access
EDITOR2 )Editor 2, full screen APL2 editor
IN      )In
PIN     )Pin
OUT     )Out
```

The following two commands are in the same namespace as IN and OUT. Simply add entries for them to the Processor 11 names file.

```
COPY )Copy *
PCOPY )Pcopy *
```

- These external functions enable you to invoke APL2 system commands.

i (Aber nicht unter dem Application Environment.)

- Note: We intend to document COPY and PCOPY so it is safe to use them. > Simply add entries: Does it mean, that these functions (P)COPY are still in the namespace or...
- You do not need a PTF. If you look in the dataset allocated to AP2TN011, you will find a names file member for each external function including IN and OUT. These names files indicate that the IN and

OUT functions can be found in namespace AP2IPDD2. This namespace also contains COPY and PCOPY functions. You can create new names file members for COPY and PCOPY and have them point to the same namespace. We will do this eventually; we just have not gotten to it yet.

Communication Tools

```
SVI      Shared variable user information
SERVER   Cross system shared system TCP/IP port server
RAPL2    Remote APL2 interpreter access
```

- Use SVI to find out the userid associated with a global SVP processor number.
- SERVER runs in a batch job providing remote processors with access local sessions.
- RAPL2 is run in a local session to control a remote session.

i (Nicht unter dem Application Environment)

Packaged Public Workspace Functions

```
DISPLAY   Display array structure
DISPLAYC
DISPLAYG
IDIOMS     APL2 idioms library
```

- We packaged these workspaces so you wouldn't have to)COPY them.

Data Processing

```
CAN  Compress and nest
DAN  Delete and nest
SAN  Slice and nest
CTN  Character to numeric
DFMT Format arrays containing DBCS data
MD5  Message Digest 5 *
```

- CAN, DAN, and SAN were precursors to partition. They are sometimes useful.
- CTN converts formatted numeric data produced by other tools to APL2 numeric data.
- MD5 is a checksum algorithm used in network communication.
- CAN, SAN, and DAN were all developed before the partition primitive.

ve. They were experiments to try to discover a good design for a primitive. The partition primitive can do everything CAN, SAN, and DAN can do. It might be better to use one of the external functions if it were easier to generate their form of left argument than partition's left argument. Note: CAN, SAN, and DAN are not supplied on the workstation so they are not portable.

- The execute primitive converts a character vector that is a number formatted by APL2. It handles high negative signs and APL2 style exponentiation. It does not handle formatting styles produced by other languages such as using the minus sign for negative numbers. Use CTN to convert data formatted by other languages.
- The MD5 algorithm is inherently iterative and does not perform well when coded in APL. Therefore, we provided a compiled version. The algorithm is too complicated to describe here. Complete details can be found in the Network Working Group's Request for Comments (RFC) 1321 at <http://www.ietf.org/rfc/rfc1321.txt>.

Namespace Creation and Management

```
PACKAGE Create namespace from saved workspace
EXP      Execute in previous namespace
QNS      Query namespace
```

- Workstation APL2 will provide the same facilities except PACKAGE is called CNS.

Converting Data in APL2 Internal format

```
ATR      Array to Record
RTA      Record to Array
PFA      Pattern from Array
SIZEOF   Size of pattern *
ATS      Array to Self-Contained Array *
STA      Self-Contained Array to Array *
CTUTF    Character to Unicode Transformation Format *
UTFTC    Unicode Transformation Format to Character *
CTK      Character to DBCS
KTC      DBCS to character
```

- ATR, RTA, PFA, and SIZEOF are used to pass data back and forth between APL2 and programs written in other languages.
- ATS and STA convert arrays between APL2 internal format and the SCAR format which can be used to send data to programs written in other APL dialects.
- CTUTF and UTFTC are used to convert data between APL2 and Unicode formats in network communications.
- CTK and KTC are used to convert DBCS data between APL2 and operating system formats.

Calling APL2 from other Languages

APL2PI	APL2 Programming Interface
ATP	Array to Pointer
PTA	Pointer to Array

- APL2PI enables APL2 to be called from other languages such as Fortran, PLI and Cobol.
- ATP and PTA enables APL2 programs to access parameters passed by reference.

External Module Management

BUILDRD	Build Routine Descriptor
BUILDRL	Build Routine List
AP2MP11L	Link edit exec found in sample libraries

- Objects in namespaces do not need names file entries because they are self-describing.
- Non-APL external functions can be also self-describing. BUILDRD is used to create routine descriptors which are linked with external routines to make them self-describing.
- A module can contain more than one function or namespace. BUILDRL is used to build the routine list which must begin such a module.
- AP2MP11L is a Rexx exec that eases the task of link-editing namespaces and external routines. It is supplied in the APL2 sample libraries.
- AP2MP11L is simply an exec for link-editing modules. The link-editor is difficult to learn how to use; AP2MP11L automatically invokes the link-editor with the correct parameters for building modules useable

from APL2. It is used to compile namespace objects (created by PACKAGE) and external functions written in other languages. It is documented in the System Services Reference.

Data Window Callable Services Interfaces

CSRIDAC	Request or terminate access to an object
CSRREFR	Refresh an object
CSRSAVE	Save changes made to a permanent object
CSRSCOT	Save object changes in a scroll area
CSRVIEW	Start or terminate a view of an object

- These functions provide access to data window objects through the callable services interfaces.

i *Im OS/390 können Adressräume für -zeitkritische- Daten definiert werden, sog. Hiperspaces. Der Zugriff auf diese Daten geschieht über Data Window Services.*

User Exits

Preview all system commands
On
Clear
Load, Copy, Pcopy, Mcopy
Save, Continue
Drop
Lib
Off
Host and AP 100 commands
SAM library dataset name generation
SAM library project library qualifier generation
GSVP
Cross system shared variable encryption exit*

- TSO APL2 provides the ability for installations to detect, monitor, and even change the behavior of system commands and shared variable interactions. This is done through „user exits“. A user exit is an assembler routine which is called by APL2 when a system command or shared variable interaction occurs.
- There are two shared variable exits:
 - The first is used by the GSVP for user-to-user shares on the same ma-

chine. It is used to validate that users are authorized to sign on to the GSVP.

- The second is used in cross-system shares which go across networks. This exit allows the developer to supply encryption and decryption functions which protect the data as it goes across the network. When a shared variable is specified, the encryption exit is called and it encrypts the data. The encrypted data is sent across the network. When the data is received, the encryption exit is again called to decrypt the data before it is assigned to the recipient shared variable.

APL2 Documentation

)Load 1 WSINFO
)Load 1 SUPPLIED
SH21-1054-01 System Services Reference
SH21-1056-01 Using the Supplied Routines
SH21-1058-00 Processor Interface Reference
SH21-1055-01 Installation and Customization under TSO
Search for APL2 at ibm.com/support *

- WSINFO describes the workspaces.
- The SUPPLIED workspace contains all the external function information from the Using the Supplied Routines book.
- IBM has a new online feature called Tech Notes. We're regularly adding Tech Notes to our web page.

Kontakt

David Liebttag, APL Products and Services, IBM Silicon Valley Laboratory, San Jose, California,
 eMail: liebttag@us.ibm.com.
 Ursula Holzer,
 eMail: uholzer@de.ibm.com

Frank Diehl

APL und MQSeries

Die Performance stimmt

Die zunehmende Anzahl von Intranet und Internet-Anwendungen bei der Allianz Lebensversicherung-AG, Stuttgart, führte dazu, daß mit dem I*net eine konzernweite Anwendungsarchitektur auf Basis von IBM-WebSphere, MQSeries und Java entwickelt wurde.¹ Mitte des Jahres 2000 kam der Auftrag, die bestehende APL-Host-Anwendung GGF (Gesellschafter-Geschäftsführer-Versorgung) ins Amis-Intranet anzubinden (Amis = Agentur Management Informationssystem, für Allianz-Agenturen). Die bisherige, mit APE-Schablonen arbeitende Anwendung wurde dafür in ein Java-Frontend und den eigentlichen Rechenkern aufgespalten, wobei beide Teile über MQSeries kommunizieren sollten. Der Rechenkern läuft weiter auf dem Großrechner. Der folgende Beitrag gibt einen Überblick über die Erfahrungen, die mit dieser Integration der APL-Anwendung gemacht wurden.

■ IBM MQSeries

▶ Message-Queueing

MQSeries (im folgenden auch MQS genann) ist ein IBM-Produkt zur Verarbeitung von Nachrichten. Es handelt sich um eine umfangreiche Produktfamilie, wovon hier nur einige Aspekte vorgestellt werden, die für den APL-Einsatz bei Allianz bislang interessant waren.

Message-Queueing ist eine **asynchrone** Verarbeitung. Wie beim Versenden einer eMail braucht der Empfänger dabei nicht aktiv sein. Eine synchrone Verarbeitung, z.B. wie beim APL-TCPIP-Prozessor AP119, würde dagegen dem Telefonieren entsprechen.

Die Arbeitsweise ist die, daß ein Rechner (z.B. ein Client) die Daten nicht direkt an den anderen Rechner (Ser-

ver) schickt, sondern sie in eine Messagequeue (eine Art Briefkasten oder Ablage) stellt, aus dem der zweite Rechner sie rausholt und verarbeitet.

MQSeries versendet und verwaltet die Nachrichten. Die Daten werden entsprechend einem Datenbanksystem gespeichert. Dies bedeutet, daß selbst bei einem Systemabsturz oder Plattencrash keine Daten verloren gehen.

Die Nachrichten in den Queues werden mit den von herkömmlichen Datenbanksystemen bekannten Mechanismen geloggt und wieder hergestellt.

Zudem sichert MQSeries, daß Nachrichten, die an MQSeries übergeben wurden, auf jeden Fall zum empfangenden System gesendet werden und nicht verloren gehen (assured delivery).MQSeries stellt ebenfalls sicher, daß Daten nur einmal versendet werden.

■ IBM MQSeries Eigenschaften

▶ Vorteile der asynchronen Verarbeitung

Im oberen Teil („Logical Connection“) von Abbildung 2 ist eine permanente synchrone Verbindung dargestellt. Hat System A oder B Schwierigkeiten, dann muß das andere System jeweils warten.

Im unteren Teil der Abbildung sind vier asynchrone Verarbeitungsvarianten abgebildet, das Element in der Mitte (grün) ist jeweils die Messagequeue, die z.B. auf einem Middleware-Server liegt.

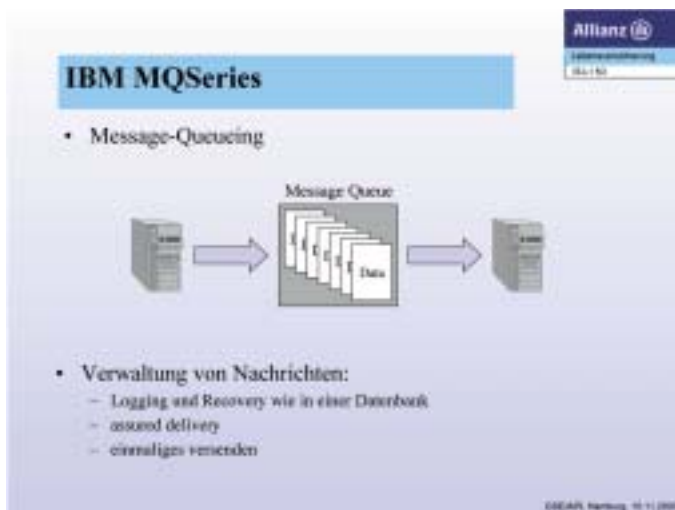


Abbildung 1: IBM MQSeries

¹ MQSeries wurde mittlerweile in WebSphere integriert als „WebSphere MQ“



Abbildung 2: Vorteile asynchroner Verarbeitung

1. Fall: Rechner B ist beschäftigt. Rechner A stellt die Daten für B in die Queue und kann weiterarbeiten. Rechner B holt sich die Daten und verarbeitet sie sobald er wieder Zeit hat.
2. Fall: Rechner B ist nicht aktiv. Ein Beispiel wäre Batchverarbeitung im Nachtbetrieb. Aus mehreren Anwendungen werden z.B. die Daten für Briefe auf die Queue gestellt, die dann im Nachtbetrieb gedruckt werden.
3. Fall: Rechner A ist nicht mehr aktiv oder ausgefallen. B verarbeitet die Daten trotzdem. Dies wäre z.B. der Nachtbetrieb zum 2. Fall.
4. Fall: Auch wenn beide Rechner A und B nicht verfügbar sind, bleiben die Daten in der Messagequeue erhalten. Bleibt der Einwand, daß die Maschine, auf der die Messagequeue liegt, auch ausfallen kann. Das stimmt. Trotzdem ist eine erhöhte Sicherheit gegeben: Gibt es beim Messagequeueserver Probleme, so merkt A dies gleich beim Senden und kann entsprechend reagieren, ist insbesondere nicht blockiert. Ist die Nachricht einmal beim Messagequeueserver angekommen, kann sie aufgrund der Logging- und Recovery-Mechanismen auch nicht mehr so leicht verloren gehen, selbst wenn der Server Probleme hat.



Abbildung 3: Lastverteilung/Skalierbarkeit

► Lastverteilung/Skalierbarkeit

Es ergeben sich aus der asynchronen Verarbeitung weitere Vorteile. Wie im oberen Bereich von Abbildung 3 dargestellt, können mehrere Agents parallel ihre Daten für den Server in die Messagequeue stellen, ohne den Server oder sich gegenseitig zu behindern.

Dem unteren Teil von Abbildung 3 kann entnommen werden, daß bei einem schnellen Rechner A oder vielen Anfragen, z.B. aus dem Internet, auch mehrere Server (Batchprogramme, Adressräume) parallel die Daten aus der Messagequeue holen und bearbeiten können.

Zu den weiteren Eigenschaften zählen:

- Triggering
- Commitpoint-Verarbeitung sowie
- das Versenden von Nachrichten mit Verfalldatum
- einfaches Codepage-Handling
- Logging und Tracing und
- Deadletterqueue

Triggering bedeutet, daß durch Einstellen einer Nachricht auf eine Messagequeue z.B. ein Serverjob automatisch gestartet werden kann, das spart Ressourcen.

Wie aus der DB-Welt bekannt ist eine *Commitpointverarbeitung* möglich: z.B. wäre es denkbar, verschiedene Zwischenergebnisse auf Queues zu stellen, Daten an Buchungssysteme zu schicken, Briefe/Dokumente auf eine Queue für einen Server zur Dokumentenverarbeitung zu stellen, sukzessive Protokoll Daten rauszuschreiben und diese Ergebnisse mit einem Rollback zurückzuziehen, falls letzten Endes eine wichtige Komponente fehlt oder sonst ein Fehler auftritt.

Im APL-Sektor wurden MQS-Commitpoints bislang noch nicht eingesetzt.

Verfalldatum: Wenn eine Nachricht nach einem bestimmten Zeitpunkt noch nicht abgeholt wurde, verschwindet sie aus dem System. Dann können z.B. alternative Aktionen gestartet werden (Meldung: "Server ist momentan nicht aktiv/überlastet, bitte tun sie dies und jenes") und bei der synchronen Emulation SENDRCV in DBCOOP wird der Client nicht weiter blockiert.

Es gibt eine *Codepage-Unterstützung*, die die eventuell unterschiedlichen Zeichensätze auf den beteiligten Systemen koordiniert.

Es gibt *Tracing- und Logging-Möglichkeiten*, die z.B. bei der Fehlersuche hilfreich sind.

Deadletterqueue: Problemnachrichten, die z.B. nicht dem vom Server erwarteten Format entsprechen können in einen besonderen Briefkasten gesendet werden, wo sie von der Administration aufgegriffen werden, sie gehen also nicht einfach verloren. Darüber hinaus gibt es noch eine Vielzahl von features, die für den APL-Einsatz noch nicht relevant waren, wie XML-, Java-Unterstützung, Zertifizierungen und Websphere-Interaktionen.

■ IBM MQSeries Plattformen

Derzeit gibt es 35 verschiedene IBM MQSeries-Plattformen². Hierzu zählen AIX, Compaq NSK, DOS, DYNIX/ptx, HP-UX, Linux, Mac OS, MVS/ESA, NUMA-Q, OpenVMS Alpha, OpenVMS VAX, OS/2, OS/390, OS/400, Solaris, UNIX, Unisys 2200 Series, Unisys A Series, UnixWare, VM/ESA, VSE/ESA, Windows 2000, Windows 3.x, Windows 95, Windows 98 and Windows NT und Java-Support.

Es werden diverse Protokolle unterstützt wie NetBIOS, SNA LU6.2, TCP/IP and SPX.

■ Erfahrungen mit MQSeries bei der Allianz

IBM MQSeries wurden bei der Allianz im Jahre 1996 im Bereich Lebensversicherungen und 1997 im Bereich Sachversicherungen eingeführt. MQS ist inzwischen eine der tragenden Säulen der Dialogsysteme. Hierbei hat sich MQS als stabil und unauffällig erwiesen. Die Betreuung von MQS erfolgt aus einer Hand über alle Plattformen.

■ MQS-Kapselung (DBCOOP)

Um MQSeries wurde eine Kapsel DBCOOP (Cooperative Processing) geschrieben. Mit dieser Maßnahme, die schon bei der allgemeinen MQSeries-Einführung bei Allianz erfolgte, und somit unabhängig von der APL-Nutzung, sollte das System vereinheitlicht und vereinfacht werden. Zu den Gründen für die Kapselung zählten zum Beispiel:

- die Abschottung der Anwendung von der Middleware
- die Reduzierung der Programmierschnittstelle (API)
- die "Emulation" einer synchronen Verbindung
- ein einheitliches Fehlerprotokoll
- eine einheitliche Bearbeitung fehlerhafter Nachrichten
- das Setzen von Defaultwerten, auch von außen

Die Realisierung erfolgt in ANSI C auf MVS, AIX, NT, OS/2 und Win 3.x. Unterstützt werden die Sprachen C, PL/1 und Assembler.

Zu den Funktionen der Kapselung zählen:

- Die Emulation einer synchronen Verbindung: SENDRCV,
- Unterstützung für den synchronen Standardfall (direktes MQGET auf ein MQPUT), ein Client erwartet eine Antwort vom Server. Die Angabe einer Verfallzeit (Expiry) gewährleistet, daß der Client nicht länger als z.B. 5 Sekunden blockiert wird.
- Der C-Code der Kapsel ist für die verschiedenen Plattformen zu 98% identisch.
- Eine Kommunikation von DBCOOP und reinem MQSeries ist gegeben, d.h. MQS kann die Daten lesen, die DBCOOP auf eine Messagequeue gestellt hat und umgekehrt.

Um die DBCOOP-Kapsel von APL aus aufrufen zu können wurde ein weiteres Schnittstellenprogramm (DP0040)



Abbildung 4: APL-Funktionalität

geschaffen, da die verpointerte Übergabestruktur an DBCOOP von (Host-) APL nicht direkt bedient werden konnte. Das Schnittstellenprogramm DP0040 wird über Prozessor 11 angebunden (name association).

■ APL-Funktionalität

Die aufgelisteten Funktionen wurden in APL realisiert. Bis auf die simulierte Synchronvariante DBCOOP_SENDRCV handelt es sich bei START, SEND, RECEIVE, etc. um die in MQSeries bekannten Funktionsnamen.

In Abbildung 5 ist der APL-Code eines Beispielservers zu sehen. Durch die erweiterten Möglichkeiten von MQSeries ist eine Vielzahl von Serverlogiken denkbar. Die gezeigte Variante arbeitet ohne Commitpointverarbeitung und ohne Bedienen der Deadletterqueue. Die abgefangenen Returncodes stellen auch nur eine mögliche Auswahl dar.

Die Schleife lp3 (Zeile 47, 65, 74) enthält die eigentliche Verarbeitung. Das Programm steht in der Zeile 49 bis über Receive eine Nachricht auf der in DBCOOP_QUEUENAME stehenden Messagequeue gefunden wird. Kommt innerhalb der in DBCOOP_WAIT stehenden Wartezeit keine Nachricht herein, beendet sich der Server.

Eventuell ist die Nachricht länger als die eingestellte Defaultlänge 1.000 bytes, dann wird in Zeile 54 nochmal mit der exakten Länge nachgelesen. Im folgenden wird unterschieden, ob der Sender eine Antwort gefordert hat, oder ob er den String 'SHUTDOWN' geschickt hat. Im letztgenannten Fall fährt der Server herunter. Im Antwortfall wird in Zeile 70 in CALL_FUNCTION eine APL-Rechenfunktion mit dem String TEXT aufgerufen und liefert die Rückgabe REPLYTEXT an den Rufer zurück. Außerhalb der Schleife wird die Kommunikation mit MQSeries mit DBCOOP_START bzw. DBCOOP_ENDE gestartet bzw. beendet.

In RCREASONEXTRACT (Zeile 35) werden temporär Variablen mit sprechenden Namen erzeugt, die die numerischen Returncodes von DBCOOP/MQSeries enthalten.

2 Vgl. <http://www-4.ibm.com/software/ts/mqseries/platforms/>

Bislang wurden beziehungsweise werden folgende APL-Projekte mit MQSeries durchgeführt: WPM-Rechner (in Arbeit), ESALES, WPM-Rechner (in Arbeit), GGF im Amis-Intranet (09/2001) und GGF für Makler (in Arbeit)³.

Fazit

APL ist in die aktuellste Allianz-Anwendungsarchitektur integrierbar. Das Triggering spart Ressourcen und geht schnell, wie MQS insgesamt eine sehr gute Performance aufweist.

Die zentrale MQS-Administration ermöglicht den Blick aufs Wesentliche, da sich der Anwendungsentwickler nicht um Netzwerkspezifika wie sich ändernde Maschinennamen/IP-Adressen oder das Firewall-Handling kümmern muß. Als Entwickler eines Front- oder Backends interessieren ohnehin nur die Daten, die in eine Komponente hinein und hinausgehen, und weniger die Frage, ob die Anwendung auf einem Großrechner, NT-Server, in einem Intra- oder Internet läuft und welcher Zeichensatz/Codepage das System am anderen Ende hat. Die Anwendungen und Komponenten werden portierbarer.

Abbildung 5: APL-Code eines Beispielservers

```
[ 1] LA SERVER01 RA,START1:H;I;N;TEXT;REPLYTEXT;LEN;namelist;USERID;LOGW;HANDLE
[ 2] A-----:-----Entspricht Guide-Standard---
[ 3] A0: Beispielserver der auf eine Messagequeue lauscht
[ 4] A0: N1[2] SERVER01 N1[]
[ 5] A0: Subroutinen (Unterprogramme)
[ 6] A0: ATS, CALL_FUNCTION, DBCOOP_RECEIVE, DBCOOP_REPLY, DBCOOP_ENDE
[ 7] Aa: DBCOOP_QUEUENAME, DBCOOP_WAIT
[ 8] Aa:
[ 9] Aa: Description (Benutzungs Beschreibung, besondere Formen)
[10] Aa: RA : USERID
[11] Aa: LA[1]: DBCOOP_QUEUENAME
[12] Aa: LA[2]: DBCOOP_WAIT
[13] A0: Comment (Funktions Beschreibung)
[14] A0: Dieses Programm ist ein Beispiel für die Nutzung der DBCOOP-
[15] A0: Funktionen. Je nach Anforderung (z.B. Commitpoint-Verarbeitung) |
[16] A0: sind auch andere Varianten denkbar.
[17] A0: Je nach Anwendung können ev. noch weitere Returncodes
[18] A0: abgefangen und verarbeitet werden
[19] A0: Die Serveraktionen werden im Session-Log mitprotokolliert
[20] Aa: Example (Beispiel f}r Benutzung)
[21] A-----:-----
[22] (USERID)+RA A ggf. noch mehr Parameter
[23] A die Parameter DBCOOP_QUEUENAME DBCOOP_WAIT müssen vorgegeben werden.
[24] A Entweder über das Linkargument LA oder als globale Variable. Letztere
[25] A Variante ermöglicht es die genannten Parameter über die JCL eines
[26] A Batchjobs mitzugeben.
[27] A →(-2=0NC 'LA')/ILA
[28] (DBCOOP_QUEUENAME DBCOOP_WAIT)+LA
[29] ILA:
[30] LOGW=120 A DEFAULT LOGBREITE
[31] A um nicht mit den Zahlen für die Return- und Reasoncodes arbeiten zu müssen,
[32] A werden die Zahlen sprechenderen Variablen zugewiesen, z.B. enthält die
[33] A Variable DBCP_OKAY den RC=0. Die Variable 'namelist' enthält die
[34] A Variablennamen:
[35] namelist=RCREASONEXTRACT
[36] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[37] A START A
[38] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[39] H-DBCOOP_START DBCOOP_QUEUENAME USERID ''
[40] REASON=2>H A|
[41] HANDLE=4>H A|
[42] →(-REASON=DBCPC_OKAY)/I/FEHLER A|
[43] LOGW!ATS,'Server gestartet' A| Ausgabe ins Sessionlog
[44] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[45] A RECEIVE-Schleife
[46] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[47] lp3: A|
[48] LEN=1000 A|Defaultlänge für RECEIVE-String
[49] H-DBCOOP_RECEIVE HANDLE DBCOOP_QUEUENAME USERID LEN '' DBCOOP_WAIT ''''''''''
[50] REASON=2>H A|
[51] →(REASON=DBCPC_MSG_TOO_BIG_FOR_RCV)/11 A|
[52] A Nachlesen mit genauer Länge, falls LEN zu klein war:
[53] LEN=5>H A|
[54] H-DBCOOP_RECEIVE HANDLE DBCOOP_QUEUENAME USERID LEN '' DBCOOP_WAIT ''''''''''
[55] REASON=2>H A|
[56] 11:→(-REASON=DBCPC_NO_MSG_AVAILABLE DBCPC_REPLY_WANTED DBCPC_OKAY)/I/FEHLER
[57] →(REASON=DBCPC_NO_MSG_AVAILABLE)/I/FERTIG
[58] LOGW!ATS,'Nachricht von ',4>H A| Ausgabe ins Sessionlog
[59] LOGW!ATS,TEXT,(5>H)16>H A| Ausgabe ins Sessionlog
[60] →(REASON=DBCPC_REPLY_WANTED)/14 A|
[61] A an dieser Stelle könnte eine Sonderverarbeitung für Messages erfolgen
[62] A die keine Antwort erfordert haben, z.B. um im Test den Server von einem
[63] A Client aus runterzufahren:
[64] →('SHUTDOWN'=TEXT)/I/FERTIG
[65] →lp3
[66] 14:
[67] A vom Sender wurde eine Antwort gefordert (SENDRCV), der Server muß nun
[68] A mit REPLY antworten
[69] LOGW!ATS,'Antwort gefordert:' A| Ausgabe ins Sessionlog
[70] LOGW!ATS,REPLYTEXT-CALL_FUNCTION TEXT A| Ausgabe ins Sessionlog
[71] H-DBCOOP_REPLY HANDLE DBCOOP_QUEUENAME USERID REPLYTEXT '' '' '' '' '' ''
[72] REASON=2>H A|
[73] →(-REASON=DBCPC_OKAY)/I/FEHLER A|
[74] →lp3 A|
[75] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[76] A Keine weiteren Nachrichten A
[77] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[78] I/FERTIG:LOGW!ATS,'Server beendet' A| Ausgabe ins Sessionlog
[79] →LENDE
[80] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[81] A unerwarteter Fehler A
[82] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[83] I/FEHLER:LOGW!ATS,'unerwarteter Fehler:' A| Ausgabe ins Sessionlog
[84] 'RC = ',1>H A| Ausgabe ins Sessionlog
[85] 'REASON = ',2>H A| Ausgabe ins Sessionlog
[86] 'FH-Meldung = ',3>H A| Ausgabe ins Sessionlog
[87] →LENDE A| Ausgabe ins Sessionlog
[88] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[89] A ENDE A
[90] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[91] LENDE: A|
[92] DBCOOP_ENDE HANDLE DBCOOP_QUEUENAME USERID '' 0
[93] 0NA=0EX namelist A| Löschen der RC-Variablen
```

Kontakt:

Frank Diehl, Allianz AG, Hamburg,
 eMail: Diehl@allianz.de

³ Amis = Agentur Management Informationssystem, für Allianz-Agenturen; GGF = Gesellschafter-Geschäftsführer-Versorgung; GGF-Aufrufe: momentan 5-600 Aufrufe pro Tag; WPM: Workplace-Marketing; Auskunftsseiten für Firmenkunden im Allianz-Intranet z.B. zur Altersversorgung der Mitarbeiter, oder direkt übers Internet. Die produktive GGF-Intranetvariante im Allianz-Intranet wird in 2002 erweitert auf Makler (nicht nur Allianz-Agenturen) im Internet

Dr. Martin Hintz

Verarbeitung von Großrechnerdaten im PC-Umfeld

Bei den Hamburg-Mannheimer Versicherungsgesellschaften kommt im Rahmen der Quartals- und Jahresabschlüsse ein unter *APL+Win* entwickelter Workspace zum Einsatz, der Daten aus PC- und Großrechnerdateien einliest, verarbeitet und sowohl auf dem PC als auch auf dem Großrechner Ergebnisdateien erzeugt. Die Kommunikation mit dem Großrechner findet dabei über TCP/IP statt.

Seit einigen Jahren erstellen die Unternehmen der Hamburg-Mannheimer-Gruppe (HM) wie viele andere deutsche Versicherungsunternehmen zusätzlich zum Jahresabschluß nach den Vorschriften des Handelsgesetzbuches (HGB) auch Jahres- und Quartalsabschlüsse nach internationalen Rechnungslegungsvorschriften (International Accounting Standards/IAS). Die Bilanzierung der IAS-Positionen erfolgt dabei i.d.R. durch eine Überleitungsrechnung, d.h. ausgehend von der HGB-Bilanz wird durch Addition von Korrekturgrößen die IAS-Bilanz erzeugt.

Speziell für den Posten Kapitalanlagen (Teil der Aktivseite der Bilanz) liegen bei der HM die Daten der HGB-Bilanz in einer Großrechner-Datenbank vor, während die für die Überleitung relevanten Zusatzinformationen auf mehrere Excel-Formblätter, also PC-Dateien, verteilt sind. Z.T. werden diese Excel-Blätter von der MEAG, der Kapitalanlagegesellschaft des ERGO-Konzerns, geliefert. Es entsteht also das Problem, die Großrechner- und PC-Daten so zusammenzuführen, daß eine IAS-Bilanz erstellt werden kann. Konkret ist ein umfangreiches Excel-Blatt zu erzeugen, das für die einzelnen Kapitalanlagen die Bilanzwerte und einige Positionen der Gewinn- und Verlustrechnung in übersichtlicher Form zusammenstellt (inkl. Zwischen- und Gesamtsummen) und anschließend auf vielfältige Weise innerhalb der HM weiterverwendet wird. Weiterhin ist auf dem Großrechner eine Datei von Buchungsinformationen zu erzeugen, damit die IAS-Bilanzwerte in die Buchhaltungsdatenbank eingebucht werden können. Verschärft wird dieses Problem der Datenzusammenführung durch einen von der Konzernführung vorge-

gebenen engen Zeitrahmen, der die Erstellung der IAS-Bilanz innerhalb weniger Tage nach Ende des Quartals fordert („Fast Close“).

Als Lösung wurde unter *APL+Win* (also auf dem PC) ein APL-Workspace entwickelt, der sowohl auf dem PC Excel-Dateien liest und schreibt als auch über eine TCP/IP-Schnittstelle mit dem Großrechner kommuniziert und dort Datenbanken liest und Dateien erzeugt.

Auf dem Großrechner (APL2 unter TSO) wurde hierzu ein Hilfs-Workspace programmiert, der die entsprechenden Hilfsfunktionen zum Lesen und Schreiben von Dateien und zum Starten eines TCP/IP-Servers (unter Verwendung des Hilfsprozessors AP119) enthält. Derartige Hilfsroutinen liegen in der HM zentral vor. Aus Gründen der Übersichtlichkeit wurde darauf geachtet, daß der Großrechner-Workspace möglichst wenig weitere Funktionalität enthält, so daß alle wesentlichen problemspezifischen Funktionen zentral im PC-Workspace vereint sind.

Vor Beginn der eigentlichen Anwendung muß auf dem Großrechner im Hilfs-Workspace eine APL-Funktion aufgerufen werden, die einen TCP/IP-Server startet, welcher dann anschließend vom PC aus angesprochen werden kann. Die Excel-Eingabedateien müssen auf dem PC (bzw. im PC-Netzwerk) unter einem gemeinsamen Verzeichnis abgelegt sein. Wenn diese Voraussetzungen erfüllt sind, läuft die Anwendung mit Aufruf einer Hauptsteuerungsfunktion im PC-Workspace dann ohne weitere Unterbrechung ab (Dauer ca. 2 Minuten). Der Name des Verzeichnisses mit den Excel-Dateien wird beim Funktionsaufruf als Parameter übergeben.

Das APL-Programm startet zunächst

eine Excel-Anwendung, liest die Excel-Dateien ein und legt die Daten in APL-Variablen ab (Vektoren von zweidimensionalen Matrizen – jede Matrix entspricht einem Tabellenblatt). Anschließend wird im PC-Workspace eine Matrix von Kennbegriffen für die benötigten Datenbankabfragen auf dem Großrechner erzeugt. Durch Aufruf von Routinen, die aus einem im HM-PC-Netz zentral vorgehaltenen Hilfs-Workspace kopiert werden, wird sodann eine Verbindung zu dem TCP/IP-Server auf dem Großrechner aufgebaut und die Matrix der Kennbegriffe zum Großrechner übertragen. Dann wird über die TCP/IP-Verbindung ein Befehl an den Großrechner abgesetzt, der bewirkt, daß dort die Routine zur Datenbankabfrage abläuft und die gelesenen Werte als zusätzliche Spalte an die vorher übertragene Matrix angehängt werden. Die Matrix wird dann an den PC-Workspace zurückübertragen. Damit liegen alle benötigten Daten im PC-Workspace vor.

Als nächstes wird das gewünschte Excel-Ausgabeformblatt (zunächst als APL-Variable) im Workspace erzeugt. In der Programmierung entstand an dieser Stelle der Hauptteil des Aufwandes; in der Ausführung ist die für diesen Abschnitt benötigte Rechenzeit hingegen nicht spürbar. Dann wird eine neue Excel-Datei angelegt, in die die Werte der Ausgabestruktur übertragen werden und die dann gespeichert wird.

Schließlich werden aus der Ausgabestruktur noch die benötigten Buchungsinformationen abgeleitet und in Form einer APL-Matrix an den Großrechner-Workspace übertragen. Es wird ein Befehl an den Großrechner abgesetzt, der den Start einer Routine bewirkt, die die APL-Matrix in eine sequentielle Datei übersetzt, die dann von den zentralen Buchungsprogrammen des HM-Großrechners verarbeitet werden kann. Die entsprechenden TSO-Befehle zum Schreiben der Datei werden dabei über den Hilfsprozessor AP100 abgesetzt, da der in der HM sonst üblicherweise für solche Zwecke verwendete AP101 nicht mit dem AP119 verträglich ist, welcher für den TCP/IP-Server benutzt wurde.

Kontakt:

Dr. Martin Hintz, Hamburg-Mannheimer Versicherungs-AG, Überseering 45, 22297 Hamburg

Maren Röttgers, Nils Dennstedt

Die Hamburg-Mannheimer präsentiert ihre Anwendung SRBB

Auf der Herbsttagung des GSE-AK „APL“ wurde das Projekt „Standardrechenprogramme Belgien“ (SRBB) präsentiert. Bei dieser Anwendung, die auf das Jahr 1989 zurückgeht, handelt es sich ein versicherungsmathematisches Rechenprogramm für die Tochtergesellschaft der Hamburg-Mannheimer in Belgien. In APL werden sämtliche versicherungsmathematischen Werte kalkuliert.

ser Hinsicht, als dass Funktionen, falls sie nicht universell genutzt werden können, auch nach Verarbeitungsart gruppiert sind.. Dies führt zu einer übersichtlichen Workspace-Architektur, insgesamt werden nur etwa 700 Funktionen (ohne das APE) benötigt. Eine Übersicht der Programmfunktionalität zeigt Abbildung 2.

Die Schnittstelle vom Subsystem zu anderen Anwendungen besteht aus einer APL2-Struktur, die in den ersten drei Elementen Verarbeitungsart, Returncode und Gesamtlänge des Verbindungsbereiches enthält, im vierten Element dann einen Datenteil, der wiederum aus mehreren Bestandteilen, je nach Verarbeitungsart verschieden, aufgebaut sein kann.

Für den Umgang mit den Rechenprogrammen, die von einer Entwicklerin betreut werden, stehen zum einen eine APE-Maskensteuerung zur Verfügung, zum anderen wird die Anwendung über TP-Programme, welche in PL/1 geschrieben sind, oder direkt über Aufrufe aus PL/1-Programmen gesteuert.

Zugriffe auf das HM-eigene Datenbanksystem erfolgen über ein PL/1-Interface durch den AP100. Die Einbettung in die HM-Anwendungslandschaft am Mainframe zeigt Abbildung 1.

Über die TP-Programme steht die Anwendung etwa 100 bis 200 Mitarbeitern in der HM-Belgien täglich zur Verfügung. Der APE-Dialog wird im wesentlichen von der Entwicklerin selbst genutzt, aber auch von Mitarbeitern des Aktuariats der HM-Belgien, die auch als Tester vor der Übernahme von Änderungen in das Produktionsprogramm die Funktionalität gegen eigene Programmierungen z.B. in MS-Excel testen. Dies sind etwa sechs weitere Nutzer dieses Mathematischen Subsystems.

Die eigentliche Steuerung der verschiedenen Berechnungen geschieht über Angabe einer Verarbeitungsart. Diese kann Antragsbearbeitung, Policierung, Vertragsänderung sein, aber auch für die HM als Muttergesellschaft wichtige Anwendungen wie Rückversicherung, Bilanzwerte und Deckungsbeitragsrechnung. Diese Steuerung strukturiert die Anwendung in gewis-

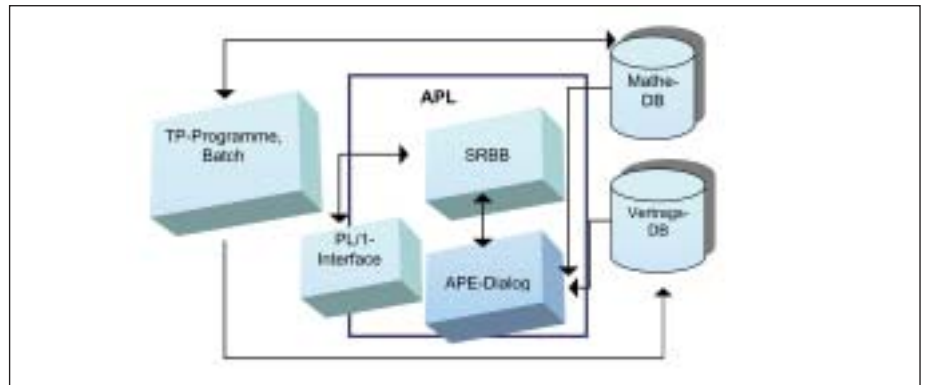


Abbildung 1: Einbettung von ARBB in die Mainframe-Anwendungsumgebung der HM

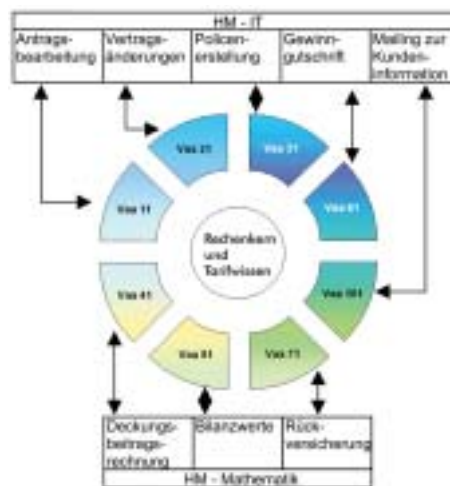


Abbildung 2: Darstellung der Anwendungsstruktur

Um die Verwaltbarkeit des Systems zu gewährleisten, wurden im wesentlichen nach dem Prinzip „jede Information nur an einer Stelle“ eine Produktsteuerung über APL-Matrizen geschaffen. So können anhand von Schlüsselkriterien alle zur Berechnung erforderlichen produktspezifischen Parameter aus Matrizen Produktindexgetrieben ausgelesen werden. Dies erleichtert nennenswert die Pflege der Anwendung.¹

1 Auf der Tagung wurde das System anhand des APE-Dialogs vorgeführt. In dieser kurzen Zusammenfassung wird auf eine Beschreibung verzichtet, da die APE-Masken nur für Kenner der Datenbank und der belgischen Tarifbesonderheiten leicht verständlich sind.

Kontakt:

Maren Röttgers, Nils Dennstedt, Hamburg-Mannheimer Versicherungs-AG, eMail: nils.dennstedt@hamburg-mannheimer.de

Sergy Alpin

Java-APL+Win Interface

Das **Java-APL+Win Interface** wurde entwickelt, um die Kommunikation zwischen Java-Anwendungen und APL+Win-Anwendungen zu ermöglichen. Unter <http://shop.dpc.de/flife/> finden Sie eine überzeugende Anwendung mit einer Implementation des Java-APL+Win Interface. Dort verbindet Java-APL+Win Interface ein Servlet mit einem APL+Win Rechenkern.

Um eine APL-Funktion von einer Java-Anwendung mit dem **Java-APL+Win Interface** aufzurufen, müssen Sie nur eine entsprechende Java-Klasse in Ihre Java-Anwendung importieren:

```
import de.dpc.japlpp.JAPLpax;
```

Anschließend erstellen Sie eine Instanz von JAPLpax:

```
JAPLpax ax = new JAPLpax();
```

Jetzt starten Sie den APL+Win ActiveX Server:

```
ax.getAPL(„0“)
```

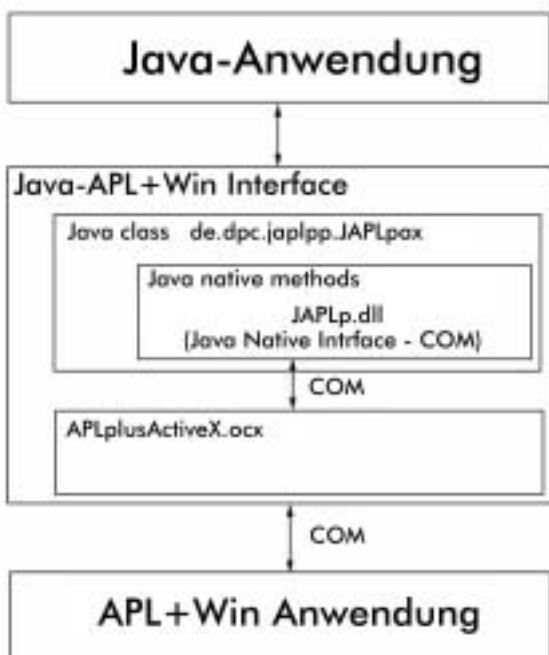
und laden ein Workspace herunter:

```
ax.loadWS(„C:/JAPLP/JAPL“)
```

Dann führen Sie eine **calls**-Methode aus:

```
String str=ax.calls(„JFUNCTION“,“Hello World!“ ).
```

In dem String `str` steht Ihnen Ergebnis zu Verfügung.



Java-APL+Win Interface besteht aus:

- einer Java-Klasse `de.dpc.japlpp.JAPLpax` (Die Klasse enthält native Methoden, die mit **JAPLp.dll** implementiert werden.)
- **APLplusActiveX.ocx** (einActiveX-Control, das die Verbindung zwischen `de.dpc.japlpp.JAPLpax` und dem APL+Win als ActiveX Server realisiert.)

Java-APL+Win Interface installieren

Softwarevoraussetzung

- APL+Win Software vorinstalliert
- APL+Win als ActiveX Server registriert

Wenn Sie bereits APL+Win Software auf Ihrem Rechner haben und APL+Win als ein ActiveX Server registriert ist, fangen Sie direkt mit der Installation der Java-APL+Win-Schnittstelle an.

Installation

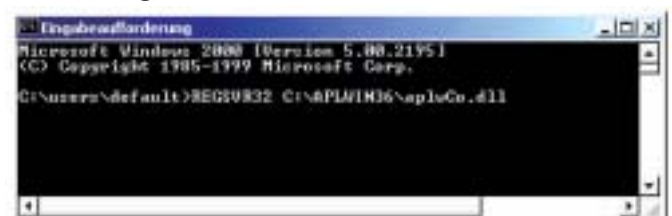
1. APL+Win installieren. (Wenn Sie keine APL+Win Software haben, wenden Sie an DPC GmbH. Eine Testversion können Sie auch herunterladen unter <http://www.apl-online.de/Downloads/downloads.html>.)

2. APL+Win als ActiveX Server registrieren:

- APL Component Object Modul

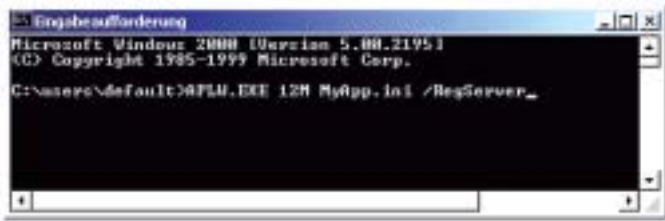
```
REGSVR32 <Pfad>\aplwCo.dll
```

Wenn Ihr APL+Win Installationsverzeichnis `C:\APLWIN36` ist, sieht Registrations-Zeile so aus:



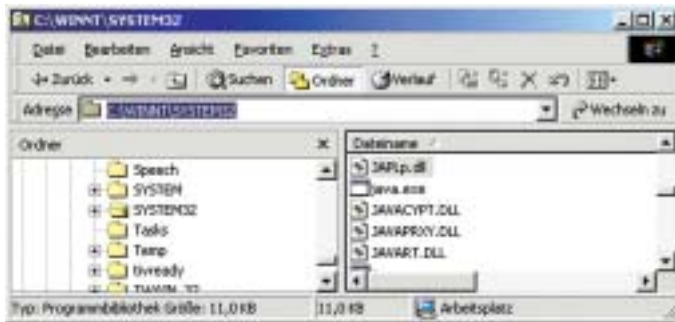
- APLW.EXE oder APLWR.EXE Modul

APLW.EXE 12M MyApp.ini /RegServer



3. Komponente des Java-APL+Win - Interface herunterladen (<http://www.dpc.de/Java/jap/japl.zip>), unzippen und installieren:

- JAPLp.dll in das Windows System Verzeichnis (Windows/System, WINNT/System32) kopieren:



- APLplusActiveX.ocx in ein beliebiges Verzeichnis stellen, zum Beispiel C:\japl
- JAPLax - Klasse in einem de.dpc.japlpp-Package in Ihr Arbeitsverzeichnis stellen, zum Beispiel C:\japl
- Beispiel-Dateien installieren
- eine Java Application JAPLptest.java
- ein APL+Win Workspace JAPL.w3

4. APLplusActiveX.ocx als ActiveX registrieren

REGSVR32 path\APLplusActiveX.ocx

Die Installation ist abgeschlossen. Jetzt können wir das Java-APL+Win-Interface testen.

Eine Beispiel-Anwendung

► Softwarevoraussetzungen

Sie brauchen ein Java SDK, um das JAPLptest.java zu kompilieren und auszuführen. Ein aktuelles Java SDK können Sie bei java.sun.com kostenlos beziehen. (Zum Beispiel das Java™ 2 SDK, Standard Edition, v 1.4 Beta).

► JAPLptest

Der Java-Code von JAPLptest.java lautet.

```
import java.lang.String;
import de.dpc.japlpp.JAPLpax; // JAPLpax importieren
class JAPLptest {
    public static void main(String[] args) {
        String str=null; //Einen Output String definieren
```

```
JAPLpax ax = new JAPLpax(); //neue Instanz von dem JAPL
pax erstellen

try {
    str = ax.getAPL(„0“); //APL+Win ActiveX Server starten
} catch (Exception e) {
    e.printStackTrace(System.out); //wenn dabei was
    schiefläuft

}

System.out.println(str); //return-Code zeigen
str=ax.loadWS(„C:/JAPLPP/JAPL“); //Workspace herunterladen:
Analog zum )LOAD

System.out.println(str); //return-Code zeigen
String arg=„Hello World!“; //rechtes Argument definieren
str=ax.calls(„JFUNCTION“,arg); //Funktion „JFUNCTION“ mit
dem arg aufrufen

System.out.println(str); //Ergebnis zeigen
str = ax.Killws(); //APL+Win ActiveX Server herunterfahren
System.out.println(str); //return-Code zeigen

}
}
```

► Workspace JAPL.w3

Starten Sie APL+Win und öffnen Sie Workspace C:\JAPLPP\JAPL (Datei JAPL.w3).



Öffnen Sie die Funktion JFUNCTION.



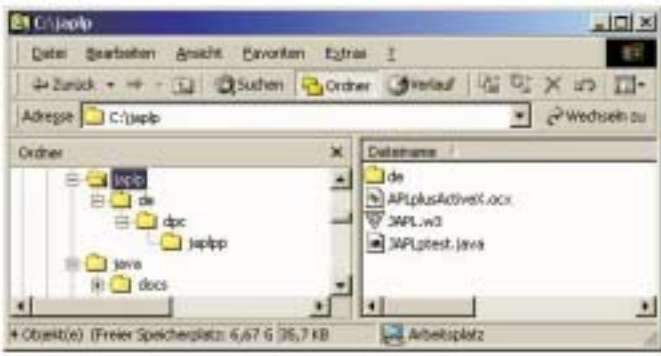
Diese sehr einfache Funktion liefert das Argument X zurück.

► JAPLptest kompilieren

Mit Eingabeaufforderungsfenster wechseln Sie zu Ihrem Arbeitsverzeichnis. In diesem Verzeichnis stehen JAPLptest.java und JAPL.w3.

Das Package **de** mit der de.dpc.japlpp.JAPLpax Klasse muss auch in diesem Verzeichnis sein.

C:> cd C:\japl



Als nächstes rufen Sie den Java-Kompiler auf:

```
C:\japl> javac -classpath .; JAPLptest.java
```

Jetzt lassen Sie JAPLptest laufen:

```
C:\japl> java JAPLptest
```

Das Ergebnis sieht so aus:

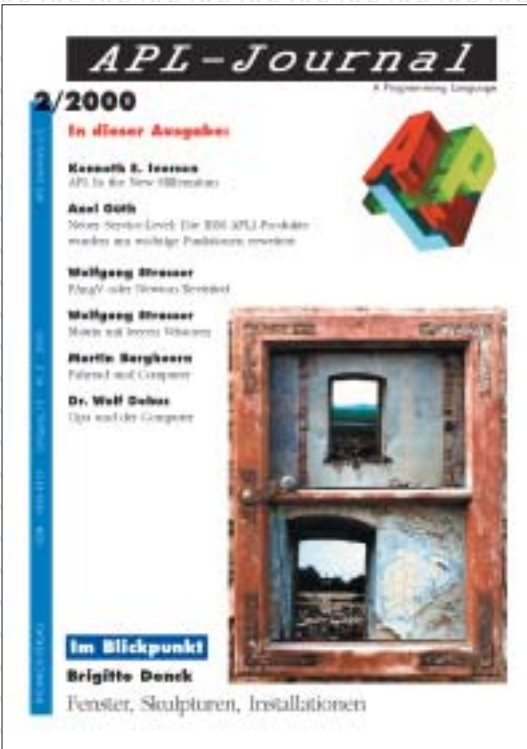
```
HelloWorld!
```

Hinweis:
 Unter http://www.dpc.de/Java/jap/JavaAPLWin_Interface.html steht Ihnen eine Online-Version dieses Artikels zu Verfügung. Dort finden Sie auch eine Beschreibung von weiteren Methoden und Eigenschaften von dem Java-APL+Win Interface JAPLpax (http://www.dpc.de/Java/jap/de_dpc_japlpp/index.html).
 Unter <http://shop.dpc.de/flife/> finden Sie eine überzeugende Anwendung mit einer Implementation von dem Java-APL+Win Interface. Dort verbindet Java-APL+Win Interface ein Servlet mit einem APL+Win Rechenkern.
 Das Java-APL+Win Interface JAPLpax ist eine von mehreren von mir entwickelten Schnittstellen zwischen APL und anderen Anwendungen, Programmiersprachen und Technologien. Zu dieser Familie gehören Schnittstellen zu Java, JSP, EJB, CGI, ISAPI, Webbrowser, SAP. Eine Beschreibung diesen Schnittstellen finden Sie an der Webseite der DPC GmbH unter www.dpc.de.

Kontakt:
 DPC GmbH, Sergy Alpin, eMail: alpin@snafu.de,
 Tel. 0331-622495



APL - Germany e.V.



Hier erhalten Sie das APL-Journal in Farbe per Download (PDF-Datei): www.rhombos.de/apljourn.htm



meinem Erstaunen feststellen, daß „Net“ mit nur einem „T“ geschrieben wird; Sicher stammt das wieder aus der komischen Sprachreform, aber eine Definition fand ich nicht. Nun durchwühlte ich das lateinische Wörterbuch. Dort fand ich „inter“ und in der zweiten Zeile zu meinem Schrecken die Buchstabenfolge „inter sicarios accusare,“ was ins Deutsche übersetzt „wegen Mordes heißt.“

Wohin war mein Enkel geraten? Sollte ich ihn über meine „Erkenntnis informieren. Ich hielt es aber dann für besser, in allmählich auf diese Gefahr hinzudeuten. Die Gelegenheit ließ auch nicht lange auf sie warten. Schon bald stand er wieder einmal von unserer Tür und wies so beiläufig auf eine gute Zensur bei der letzten Arbeit hin. Also kam sicher wieder eine Überraschung auf mich zu.

Seine Frage war jetzt an mich gerichtet, ich solle ihm erklären, wieso er eine solche Leistung zustande gebracht hätte; eine fadenscheinige Begründung. Ehe ich antworten konnte, sagte er schon: Das habe ich dem Internet zu verdanken. Nun war er also schon so weit gesunken, daß er sich mit inter sicarios accusare verband. Mit dem Lateinbuch in der Hand versuchte ich, ihn erst einmal auf die Gefahr hinzuweisen. Da mußte er lachen und tat mich mit dem Worten ab: Du verstehst überhaupt nichts von Internet.

Das muß man sich nun als Großvater sagen lassen. Anstatt mich aber aufzuklären verwendete er weitere mir völlig unbekannte Begriffe wie Fax, I mail und Tandem oder so ähnlich; kann auch Modem gewesen sein. Da er nun mein völlig verdutztes Gesicht sah, gab er zunächst seine Aufklärungsarbeit auf, um diesmal wieder ohne Erfolg nach Hause zu gehen und wie er mir erklärte – zu surfen (sicher meinte er skaten; denn zum Surfen braucht man doch eine Wasseroberfläche, jedenfalls glaubte ich das bisher). Diesmal war ich zufrieden, daß er sich doch entschied, an die frische Luft zu gehen. Da er jedoch keine Jacke mit hatte, riet ich ihm, diese erst zu holen. Seine Antwort war nur: Ja, ja Opa. Und er ver-

Wieder einmal erfreute uns unser Enkel mit seinem Besuch. Er hat es nicht weit und so nutzt er gerne die Gelegenheit für einen Sprung zu Omi und dem Opa. Natürlich sind die Gründe zum Besuch recht unterschiedlich, meist aber Überraschungen mit denen man nicht gerechnet hatte. Diesmal war es wieder ein „beiläufiges“ Gespräch über den Computer. Es begann aber nicht direkt, sondern allgemein. Wozu auch die Frage zählte, ob ich auch die Gelegenheit nutze, mehr mit dem Computer zu machen, als nur Briefe zu schreiben, etwas zu rechnen oder einige Spiele auszuprobieren. Was sollte ich dazu sagen. Meine einzige Antwort, was sollte ich sonst noch mit dem Computer tun, er verbraucht schon so genug Zeit. An seinem Gesicht merkte ich, daß er nach einer bestimmten Antwort grübelt. Als wäre es ein anderes Thema, erwähnte er so beiläufig das Wort „Internet“

Was das allerdings mit dem Computer zu tun hat, war mir nicht klar. Zwar gab es so manches, was ich nett oder auch weniger nett fand, Dazu zählt natürlich auch die Entwicklung der Sprache. Zu unserer Zeit war etwas schlicht und einfach nett, Aber schon bald wurde aus dem Nett ein „Urstnett“, was nicht aus dem lateinischen oder Englischen kommt; sondern aus dem Sächsischen und zwar von „äußerst“ abgeleitet wurde. Etwas Besonderes erhielt also nicht mehr die Bezeichnung „Schön“, sondern Urst. Das Wort war „in“. Aber schon bald reichte auch dieser Ausdruck nicht mehr, und alles wurde nun plötzlich super; so auch supernett, was wohl aus dem Lateini-

schen stammt. Aber weit später als das Wort „urst“ in unserer Sprache Eingang fand. Ist ja bei der weit aus größeren Entfernung von Rom verständlich. Auch von Internet hatte ich schon was gehört, konnte aber nichts damit anfangen. Mit Zeitmangel begründete ich, warum ich mich nicht um diese Form des Nettseins bemüht hatte.

Da ich schon genug Ärger mit meinem Computer hatte, bog ich das Gespräch ab. Er spürte meine Ablehnung und handelte wahrscheinlich nach dem Motto: „Es gibt viele Wege nach Rom, dieser war nicht der richtige. Es wird sicher einen besseren geben.“ Man merkte, daß er kein Politiker ist, denn der ist stets von sich überzeugt; als einziger den richtigen Weg zu kennen. Nicht ganz zu frieden mit sich ging er nach dem Abendbrot nach Hause. Jedenfalls vertrat die Oma diese Ansicht. Trotzdem war das Gespräch so intensiv, daß das Wort „Internet“ nicht nur in meinem Gedächtnis blieb, sondern auch darin herumgeisterte.

Zuerst griff ich einmal zu meinen Wörterbüchern um beim nächsten Gespräch, das ich mit Sicherheit erwartete, nicht ganz unwissend vor meinem Enkel zu stehen, wenn er wieder mal mit diesem Begriff anfang. Also suchte ich erst einmal in meinen alten Wörterbüchern. Das half mir aber nicht weiter. Beim Durchlesen von Presseartikeln konzentrierte ich mich nun auf das Wort „Internet“ und konnten zu



schwand so schnell, wie er gekommen war.

Ich war mir jedoch fast sicher, daß dieses Thema noch nicht beendet war. Und wie ich es gehaut habe, begann mein liebes Eheweib abends vor dem Einschlafen ein Gespräch, in dem mehrmals das Wort „Internet“ vorkam. Obwohl sie sicher auch nicht mehr als ich davon verstand, erklärte sie mit fester Überzeugung, daß es doch schön wäre, wenn wir auch ein Internet hätten. Es war nicht schwer zu erraten, woher der Vorschlag kam.

Also begann schließlich für mich wieder der Weg zu verschiedenen Stellen und ein neues Suchen von Symbolen und ihrem Zusammenhang. Da das recht langsam ging, erwähnte mein Enkel den Onkel Fritz, der eine besondere Karte haben müsse. Ich dachte immer, er verstehe überhaupt nichts von Computern. Nun hatte er meinen Nerv getroffen und schon bald surfte auch ich, allerdings nicht mit Surfbrettern durch das Internet. Zum Leidwesen meiner lieben Ehefrau, hatte ich

nun überhaupt keine Zeit mehr für den Staubsauger und andere Haushaltsgeräte! Aber daran war sie selbst schuld. Sie wollte ja das Internet – oder gab es andere Gründe für ihren Wunsch?!



APL, ein Gedicht

**Die Welt, das sind die Bilder,
Klänge, Geräusche, Gerüche.
Meer & Wind, Rhythm & Blues,
Farben, Formen, Felder.¹⁾**

**Wir, die wir in der Welt sind,
tragen den Kopf hoch oben,
verbunden mit Umwelt & Mensch
durch Sprachen, Wörter, Symbole.**

**Das Verstehn, ist es so schwierig
mit Formeln, Noten, Diagrammen.
Wieviel Code verstehn der Chip
und der Partner, wohnt nebenan.**

**Das Überflüssige, laßt es weg,
ein bis drei Gebote statt zehn.
Babylon, Strafe, wirkt sie noch in
den Zeichen allein, ohne den Sinn.**

**Die Ungläubigen, was ist ihre Schuld,
Signal unerkannt, Datenfehler.
Die Botschaft nur zum Prozessor hin
Schleife & Irrweg, Ausgang verpasst.**

**Die Bücher, gestorben bevor sie gelebt
welche Hand kann sie tragen, wozu?
Zerfallen das Wissen, liegt im Staub
Phönix in Computerasche, ohne Ideen.**

**Die Romane, die schreibt das Leben
nur unklar. Unser Schicksal, das leichte?
Vergoldete Enden, durch Kabelreste
verknüpfte Begriffe, verronnen dahin.**

**Die Buchstaben, wirre Objekte.
Bibliothek, nicht mehr gefunden.
Ungepflegte Metapher, Bedeutung passè
das ist der Müllplatz, Abkürzung tot.**

**Habt ihr die Richtung, weiter den Weg.
Die Macht der Zeichen, einfach Magie.
Symbole waren Bilder, müssen leben
wie in einem Gedicht, das ist APL.**

¹⁾ Arrays

Autor: Martin@Banghoorn.com

Bummel durch einen Versandhaus-Katalog

Natürlich wollen sie verkaufen, ich meine die Versandhäuser. So passen sie den Ton der Zeit an. Sicher ist es wichtig, dass man sie versteht. Aber wenn man einen Katalog durchblättert, kommt man manchmal aus dem Kopfschütteln nicht mehr heraus.

Junge Leute (so sagte man früher) sollen natürlich besonders angesprochen werden: *Topaktuelle Fashion-Girls* sind das, und sie bevorzugen in diesem Sommer bei fast allen Modellen den *Transparenz-Look* (bei Bedarf mit *Push up* BHs mit einschiebbaren Zauberkerzen), es sei denn, sie stehen mehr auf dem *Colour blocking workerlook Style*. Da gibt es einen *Long-Bluse-Trend*. Als besonderer *Trendsetter* gehört dazu eine *Casual Jacke*, was immer das ist. *Bootcut Hosen* mit *Frontzipper* sind angesagt. *Trend-Tipp* ist ein *Ajour-Twin-Set* und alles möglichst aus *High-Tech-Material*. Wenn man dazu noch *Tattoo* Schmuck trägt, kann man sagen: *trendiger* geht's nicht mehr!

Die ältere Generation wird nicht vernachlässigt. Die Wörter sind nicht mehr ganz so aufregend, dominant ist die Dehnbundhose. Sie gehört zu den *Lady's Basis Hits* und ist aus *softem* Obermaterial. Ein echtes *Highlight* ist die *Wellness-Hose*. Bei den Herren findet man den *Streetwear-Trend*, wozu sicher die *Streethose*, die *Longjacke* und vielleicht auch der *Business-Schnürschuh* gehört. Bei der *Sport-fashion* ist wohl eher eine *British Knights Shorts* mit *Light & Soft* Schuhen vorzuziehen.

Haushaltsgegenstände werden empfohlen. Dabei wird auf den schlichten Hausfrauenverstand Rücksicht genommen. Wenn man Gardinen mit *Crinkle-Effekt* kennt und Möbel mit *gesofteten* Profilkanten akzeptiert: Länge, Breite, Tiefe werden weiterhin in den vertrauten Metermaßen angegeben. Anders bei technischen Wunderwerken, da kann man wieder ordentlich zulegen. Sehr erbaulich ist der Armbanduhr Computer, weil er super für alle *Outdoor Fans* ist. Empfehlenswert auch ein Radio-Cassetten-Recorder mit *Incredible-Surround-Raumklang* oder eine *Home Entertainment-Anlage* mit integriertem *Dolby Pro Logic Decoder*. Beruhigend ist sicherlich, dass man bei den Videospielen Geräte finden kann, deren *Erweiterungs-Slots* für *visual Memory Dreamcast Vibration* zum *D. Controller kompatibel* sind. Wenn es um die *Fitness* geht, findet man - schon vom Aussehen her - erstaunliche Konstruktionen, die irgendwie an modernisierte Foltergeräte erinnern: *Airwalker*, *Stepper* und *Beincurler*. Besondere Aufmerksamkeit findet bei mir ein Paar *Inlihe Skates*, die mit *Memory-Schnalle*, *Veico Strap*, *Speed Lacing System*, *Cushion-fit-Innenschuh*, *Anatomie-Einlegesohle*, *Dark Clear ROVES-Rollen*, *Messing Spacer*, *Alu Rocker Washer* für nur 249,-DM preislich wirklich günstig angeboten werden. Fast unnötig zu sagen, dass *Easy Shopping per Internet* möglich ist!

(Irmgard Peters, 2000)

Urlaub auf der Insel

Ein Tag wie diesen kann man nicht bezahlen, so sehr das mancher sicherlich begehrt. Das Wetter lässt uns keine freien Wahlen, es ist, wie's ist. Es wird uns nur beschert.

Um mich herum ein Farbspiel ohne Grenzen, das geht durch die Palette kreuz und quer. Der helle Strand und das kristallne Glänzen

von grün bis blau im schaumgekrönten Meer.

Ein Flugzeug schreibt bedächtig seinen Streifen ins tiefe Blau am hohen Himmelszelt. Mit meinen bloßen Füßen Sand zu greifen, scheint wichtiger als alles auf der Welt.

Ein leichtes Prickeln ist das Wundersame, wie man es kennt in einer Schale Sekt. Mir ist, als lief ich fröhlich hier Reklame in einem hochglanzfarbigen Prospekt

(Irmgard Peters, Am Rosenanger 72 a, Berlin-Fronau)

Allgemeine Informationen

(Stand 1. Dezember 2001)

Vorstand

Vorsitzender:

Dieter Lattermann
Rheinstraße 23
69190 Walldorf
Tel. (06227) 63469
E-Mail: dieter_lattermann@compuserve.com

2. Vorsitzender:

Martin Barghoorn
Technische Universität Berlin
Sekt. FR 6-9
Franklinstr. 28
10587 Berlin
Tel. (030) 314 24392
Fax: (030) 314 25901
E-Mail: barg@cs.tu-berlin.de
Web: <http://stat.cs.tu-berlin.de/~barg/>

Schriftführer

Conrad Hoesle-Kienzlen
Allianz AG
Stuttgart
Tel. 0711 663 48 28
Email: chk@hoesle-kienzlen.de

Schatzmeister

Jürgen Beckmann
Datev Nürnberg
E-Mail: JBecki@compuserve.com

Beitragssätze

Ordentliche Mitglieder:

Natürliche Personen	31,00	Euro p.a.
Studenten / Schüler	10,00	Euro p.a.

Außerordentliche Mitglieder:

Juristische / natürliche Personen ≥ 511,00 Euro p.a.

Bankverbindung: BVB Volksbank eG Bad Vilbel
BLZ 518 613 25, Konto-Nr. 523 2694

Hinweis:

Wir bitten alle Mitglieder, uns immer gleich *Adressenänderungen* und *neue Bankverbindungen* mitzuteilen.

Geben Sie bei *Überweisungen* den *Namen* und/oder die *Mitgliedsnummer* an.

Einzugsermächtigung

Ich erkläre mich hiermit widerruflich damit einverstanden, daß APL Germany e.V.

den jeweils gültigen Jahres-Mitgliedsbeitrag von meinem unten angegebenen Konto abbucht.

Einen eventuell bestehenden Dauerauftrag habe ich bei meiner Bank gelöscht.

Bankbezeichnung: _____

BLZ: _____

Konto-Nr.: _____

Datum: _____

Unterschrift: _____

APL Germany e.V. Aufnahmeantrag/Änderungsanzeige

APL Germany e.V.
 c/o Martin Barghoorn
 Technische Universität Berlin
 Sekr. FR 6-9
 Franklinstr. 28
 10587 Berlin

Ich ersuche / wir ersuchen um Aufnahme in den *APL Germany e.V.*
 als außerordentliches Mitglied (jur. Person natürl. Person)

Die Vereinssatzung ist mir / uns bekannt: Ja Nein

Änderungsanzeige

Der Verein darf Aufkleber mit meiner / unserer Postanschrift weitergeben:

Ja Nein

	Dienstanschrift/ Hauptansprechpartner	Gegebenenfalls weitere Mitglieder der Organisation
Firma		
Abteilung o.ä.		
zu Hd. von		
Straße, Hausnummer		
Postfach		
PLZ, Ort		
Bei Ausland: Staat		
Telefon (Vorwahl / Ruf)		
FAX (Vorwahl / Ruf)		
E-Mail-Adresse		

Die Angaben dieses Formulars werden für vereinsinterne Zwecke auf Datenträger gespeichert.

Ort, Datum: _____ Unterschrift: _____

<http://www.acm.org/sigapl/qqv31n2.htm>

APL QUOTE QUAD

APL2001: An Arrays Odyssey

Volume 21 Number 2

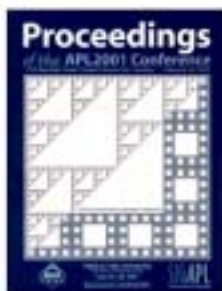


Table of Contents

- Correctly Detecting Intrinsic Type Errors in Typeless Languages Such as MATLAB [*Francis G. Joubert and Prithviraj Suresher*]
- J and speed betting [*Pablo Loureiro*]
- Replacing an Explicit Loop by the "Each" Operator [*Zsuzsanna Liptovacs and Joseph Burchfield*]
- An Array-Oriented (APL) Wish List [*Joe Lucas*]
- D2 Cell Data Mining Utility [*John Manges*]
- A Dynamic APL Gas Equation Solver (ADAGE) [*Stephen M. Minnow*]
- Good Taste in Color and Soft Palettes [*Devon McCormick*]
- Embedding a Tcl Web Server into APL [*Thomas P. Otto*]
- Graph Coloring in J: An Introduction [*Howard A. Finkle*]
- Is There a Way of Combining Array-Processing and Object-Oriented Programming? [*Georg Reichard*]
- Array Form Representation of Idioms Recognition Systems for Numerical Programs [*Mitsunori Sato*]
- Classification Trees in APL: Implementation and Application [*Alexander Stenarobkov and Vladimir Kuznetsov*]
- Suggestions for a Method of Analyzing Binary Images Using Langlet's Parity Logic [*Shaun Smith*]
- A Case for the Representation of Multivariate Arrays in the APL Computing Language [*Richard Drexel Stockbridge*]



APL - LPA 2002

*Array Processing Languages
Lore, Problems and Applications*

22-25 July 2002

MADRID



Call for papers



<http://www.apl2002.com/>

<http://www.vector.org.uk/>

Out Now! 18:2...

Invitation to FinnAPL Forest Seminar 2002

VECTOR

Vol 18 No. 2
October 2001

The Journal
of the British APL
Association

APL News, Views and Code ...

- APL2002 Call for Papers 11
- First Release of A for Windows 14
- McIntyre on Conjunctions 64
- Gustafsson on GUI and Databases 78
- Using the Windows Date-formatter 97

- [Contents](#)
 - [Editorial](#)
 - [A for Windows](#)
 - [GUI and Databases](#)
 - [Hacker's Corner: Date formatting](#)
 - [Persistence and Shadows](#)
-
- [Recent Vectors](#)
 - [Resources](#)
 - [Quick Reference Diary](#)
 - [Product Guide](#)
 - [Bookmarks](#)
 - [Search the Index](#)

APL NEWS

News and Dates for your Diary	7
Correspondence:	9
British APL Association News	
An Implementation of A	Adrian Smith 14
News from Sustaining Members	18
APL Product Guide	Gill Smith 23
The Education Vector	
Zark Newsletter Extracts	edited by Jon Sandles 38
Censored	
Lothings 3D Joesthetics	Norman Thomson 50
Getting Started with A	Adrian Smith 56
An Elementary Tutorial on Conjunctions	Donald McIntyre 64

GENERAL ARTICLES

Using Structures between GUI and Relational Databases	Tomas Gustafsson 78
Can Anyone Tell Me Where I'm Wrong?	Sylvia Camacho 86

TECHNICAL SECTION

Hacker's Corner: Formatting Dates in Dyalog APL and APL+Win	Adrian Smith 97
Technical Correspondence	104
Persistence and Shadows	Stephen Taylor 110
MUCIPHER: a new Short Cipher that Escapes Cryptanalysis	Michel Dumontier 117
Function Models	Phil Chartney 132
Index to Advertisers	143

Out Now! 17:4...

includes reports and pictures of Finnish Forests V



Spot The Axe Competition
Answer and Winner

Report of [BAA AGM 18th May](#)
News of [APL2001 in Yale](#) 25-28 June 2001

VECTOR

Vol.17 No.4
April 2001

The Journal
of the British APL
Association

More eXtreme APL ...

• the VirNeun Puzzle	38
• Barman on Dyalog 9 and XML	48
• Forests in Finland V	66
• John Scholes on "D"	93
• Anne Wilson on Celtic Knotwork	101
• McDonnell's Erdős Number	113

- [Contents](#)
 - [Editorial](#)
 - [Puzzle Corner](#)
 - [Finnish Forests V](#)
 - [Scholes on "D"](#)
-
- [Recent Vectors](#)
 - [Resources](#)
 - [Quick Reference Diary](#)
 - [Product Guide](#)
 - [Bookmarks](#)
 - [Search the Index](#)

<http://www.vector.org.uk/>

<http://www.vector.org.uk/> [1 van 2] [13.07.01 09:51:27]



Book on Demand

Lesen Sie unsere Anleitung
im Internet:
www.rhombos.de

Ideal für die kostengünstige Publikation von wissenschaftlichen Berichten, Tagungsbänden, Gutachten, Monographien, Buchreihen, Diplom- und Doktorarbeiten ... inklusive Verlagsbetreuung!

BOOK ON DEMAND

Planen Sie, ein Buch zu veröffentlichen? Haben Sie vielleicht schon ein fertiges Manuskript in der Schublade, aber bisher vergeblich versucht, einen Verlag zu finden?

Der Weg, den ein Manuskript gehen muß, bis es in der Bücher- auslage einer Buchhandlung landet, ist beschwerlich. Es sind nicht immer inhaltliche Gründe, die dazu führen, daß Verlage Manuskripte ablehnen. In erster Linie spielen hier finanzielle Erwägungen eine Rolle, denn der Aufwand, der für eine Buch- produktion betrieben werden muß, ist enorm. Entsprechend hoch ist das finanzielle Risiko für den Verlag, das mit einer Buch- veröffentlichung einhergeht.

Vielfach bleiben auf diese Weise exzellente Bücher auf der Strecke, für die nur Kleinauflagen in Frage kommen, weil sie nur einen begrenzten Kreis von Abnehmern erwarten lassen. Besonders hart betroffen davon sind beispielsweise Fachbü- cher. Vielfach leisten Autoren deshalb erhebliche Druckkosten- zuschüsse, um dennoch veröffentlicht zu werden.

Mit dem Book-on-Demand-Verfahren bieten wir hier eine Al- ternative. Wir können auf diese Weise den Druck der Bücher direkt an die Nachfrage koppeln und können Ihr vollwertiges Buch selbst in Kleinstauflagen schnell und preiswert publie- zieren. Gedruckt wird nur noch, was bestellt wird. Damit reduzie- ren wir die Druck- und Lagerkosten auf ein Minimum, entspre- chend sinkt das Investitionsrisiko.

Bei dieser Herangehensweise setzen wir auf eine enge Ko- operation mit unseren Autoren. Sie liefern uns Ihre Arbeit als Datei, am besten im Postscript-Format. Wie das geht, können Sie unserer Anleitung im Internet entnehmen. Anschließend geht Ihr Werk in den Digitaldruck und schon nach wenigen Tagen können Sie Ihr Buch in den Händen halten. Zu einem sagenhaft günstigen Preis!

Und selbstverständlich sind Sie an den Einnahmen beteiligt. Überzeugen Sie sich selbst und informieren Sie sich auf unse- rer Internetseite!

Wir führen die gesamte verlagstechnische Betreuung Ihres Werkes durch, hierzu zählen beispielsweise Vermarktung, Aus- lieferung und Abrechnung. Die gedruckten Exemplare erhalten eine ISBN-Nummer und sind damit weltweit über das Verlags- verzeichnis zu identifizieren. Zusätzlich wird Ihre Publikation mit Inhaltsverzeichnis und Kurzbeschreibung/Vorwort in den RHOMBOS-Online-Shop aufgenommen, über den sie bequem bestellt werden kann, auf Wunsch auch in elektronischer Form (PDF), beispielsweise aus dem Ausland, wenn der Versandweg zu lange dauern würde.

Interessiert? Wir beraten Sie gerne!

RHOMBOS-VERLAG, Kurfürstenstr. 17, D-10785 Berlin
Tel. 0930-261 68 54 eMail: verlag@rhombos.de