

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK
GARCHING BEI MÜNCHEN

EDDAR Benutzeranleitung

Ch. Tichmann
J. Steuerwald

R/33

November 1979

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

E D D A R B E N U T Z E R A N L E I T U N G

Ch. Tichmann
J. Steuerwald

Voraussetzung zum Verstaendnis:
Kenntnis des GALE-Systems

Eine Kopie dieses Manuals erhaelt man durch das AMOS -
Kommando "XS EXPP:INFO.EDUSGD".

Inhaltsverzeichnis

0. Einfuehrung	1
1. Kommunikation zwischen Experiment und Rechenzentrum ..	3
2. Speicherung der Daten und die zugehoerigen Directories	12
3. Benutzerinformationen in AMOS-Files	14
4. System Implementation	16
5. Checkliste	19
Anhang (JCL-Karten)	20

Zur Implementation des Systems EDDAR im Experimentrechner genuegt das Verstaendnis der Abschnitte 4 und 5. Es wird trotzdem empfohlen, die Abschnitte 1 bis 3 zu lesen, um sich einen besseren Ueberblick zu verschaffen.

0. Einfuehrung

Das EDDAR-System uebertraegt die auf einer PDP11 vom GALE-System (vgl. GALE-Manuals) gesammelten Experimentdaten in das Rechenzentrum und archiviert und verarbeitet sie dort. Das System besteht aus 4 voneinander unabhaengigen Komplexen, von denen zur Zeit nur der erste implementiert ist. Dieser erste Teil dient zur Konversion und Archivierung der Rohdaten. Die Beschreibung der uebrigen Teile kann man dem EDDAR System Survey entnehmen.

Die Rohdaten eines Schusses sollten moeglichst vor dem naechsten Schuss oder, wenn die Zeit zwischen zwei Schuessen nicht ausreicht, am Ende eines Schusstages zusammenhaengend uebertragen werden. Deshalb ist das Hauptprogramm des EDDAR-Systems (PETER) ein Dauerjob im Rechenzentrum, der auf Daten vom Experimentrechner wartet, die er verarbeiten kann. Die Arbeitsweise von PETER wird im naechsten Kapitel erkluert.

Die Daten sind im EDDAR-System bis auf einen Unterschied so organisiert wie im GALE-System: Es koennte nicht, wie im GALE-System, fuer die Daten jedes Schusses ein neues File eroeffnet werden, sondern die Daten der verschiedenen Schuesse stehen hintereinander in einem OS-File. Um die Daten eines bestimmten Schusses wiederfinden zu koennen, braucht man also ein Inhaltsverzeichnis (AMOS-Segment DICN).

Die notwendigen Informationen darueber werden im 2. Kapitel gegeben. Ausserdem gibt es noch AMCS-Files, die Informationen fuer den Benutzer enthalten, z.B. ein Verzeichnis der uebertragen Daten und ein Protokoll der verarbeiteten Information. Davon ist im 3. Kapitel die Rede. Schliesslich wird noch eine Anleitung gegeben, wie man das System startet und verwendet.

1. Kommunikation zwischen Experiment und Rechenzentrum

Im Rechenzentrum laeuft der Dauerjob PETER, der darauf wartet, dass ein Experiment ihm Daten zur Verarbeitung schickt. Jedes Experiment hat in seinem Peripherierechner (PDP11) auch einen Dauerjob (EDRRCV), der PETER Daten schickt und wartet, bis diese Daten verarbeitet sind. PETER kann bis zu 20 Experimente gleichzeitig bedienen. Zunaechst wird ein Ueberblick ueber den Kommunikationsprozess zwischen diesen Jobs gegeben, anschliessend werden die einzelnen Punkte naeher erlaeutert.

Folgende Speicher, die im Zentralrechner angelegt wurden, sind am Kommunikationsprozess direkt beteiligt:

A) Fuer jedes Experiment ein Pufferspeicher DATA.RECV.<expn> (<expn> bedeutet: Experimentname), in den die Rohdaten ueber RJE mit Hilfe des Programms TRANS von dem zugehoerigen Peripherierechner geschrieben werden.

B) Ein Speicher DATA.TIME zur "Koordinierung" der Arbeiten der Peripherierechner und von PETER:

Ein Eintrag erfolgt, wenn ein Job Daten in seinen Pufferspeicher geschrieben hat.

PETER verarbeitet die Daten, die in den Pufferspeicher geschrieben wurden und loescht den Eintrag in DATA.TIME.

C) Jeder Peripherierechner besitzt einen Speicher, in dem

PETER die Quittung eines Auftrags vom Experiment oder einen Auftrag fuer den Experimentrechner deponiert.

Der Kommunikationsprozess selbst laeuft folgendermassen ab:

- 1) Ein Peripherierechner schreibt ueber RJE mit Hilfe des Programms TRANS Daten in seinen Pufferspeicher DATA.RECV.<expn>.
- 2) Der Peripherierechner traegt dann (auch mit TRANS) einen Hinweis auf die in seinem Pufferspeicher deponierten Daten in den Koordinierungsspeicher DATA.TIME ein und wartet nun auf Nachricht von PETER.
- 3) PETER liest DATA.TIME in regelmaessigen Abstaenden ein und stellt dabei fest, dass neue Daten da sind.
- 4) PETER loescht den Eintrag in DATA.TIME, liest die Daten aus DATA.RECV.<expn> und verarbeitet sie.
- 5) PETER schreibt eine Meldung in den Speicher des Peripherierechners, dass er entweder
 - a) die Daten verarbeitet hat und der Auftrag somit erledigt ist oder
 - b) infolge eines Fehlers die Daten nicht verarbeitet werden konnten (jeder Fehler hat eine Nummer; die Nummer des Fehlers wird zurueckgegeben).
- 6) Bei 1) wieder anfangen.

Es muss verhindert werden, dass mehrere Jobs gleichzeitig

auf den Koordinierungsspeicher DATA.TIME zugreifen. Deshalb laeuft ein Zugriff auf DATA.TIME immer so ab:

Ein Job, der DATA.TIME lesen oder beschreiben will, "sperrt" dieses File fuer alle anderen Benutzer, so dass nur der Job, der das File gesperrt hat, darauf zugreifen kann. Alle anderen Jobs muessen warten, bis der Zugriff beendet und das File DATA.TIME freigegeben ist. Dann erst kann ein anderer Job das File sperren.

Der Pufferspeicher DATA.RECV.<expn> braucht im Gegensatz dazu nicht gesperrt zu werden, obwohl sowohl PETER als auch ein Experiment darauf zugreifen. PETER liest DATA.RECV.<expn> naemlich nur dann ein, wenn er einen entsprechenden Eintrag in DATA.TIME gefunden hat. Der Dauerjob (EDRRCV) des Experiments macht seinen Eintrag in DATA.TIME aber erst dann, wenn er seine Daten auf DATA.RECV.<expn> geschrieben hat. Er schreibt erst dann die naechsten Daten auf DATA.RECV.<expn>, wenn er die "Fertig"-Meldung von PETER, der nun nicht mehr auf DATA.RECV.<expn> zugreift, bekommen hat.

In dem Koordinierungsspeicher DATA.TIME ist jedem Experiment ein Bereich von 6 Bytes Laenge fest zugeordnet. In diesen Bereich wird bei der Uebersendung eines Auftrags Datum und Uhrzeit eingetragen. Daran erkennt PETER, in welcher Reihenfolge die Experimente Anforderungen an ihn gestellt

haben. In dieser Reihenfolge werden sie abgearbeitet. Da es verschiedene Arten von Daten gibt (Configuration- und Namelistfiles und Rchdatenfiles, vgl. GALE-Manuals), muss PETER mitgeteilt werden, wie die Daten zu verarbeiten sind, die er auf DATA.RECV.<expn> findet. Deshalb wurde der sogenannte "Messagerecord" eingefuehrt. Dieser Messagerecord wird den eigentlichen Daten vorangestellt, so dass PETER, wenn er DATA.RECV.<expn> liest, als erstes den Messagerecord findet und daraus entnehmen kann, wie er die nachfolgenden Daten verarbeiten muss. Dieser Messagerecord wird als Meldung, mit Informationen aus dem Rechenzentrum versehen, an den Peripherierechner des Experiments zurueckgeschickt, wenn der Auftrag erledigt ist.

In den Messagerecord muss der Dauerjob des Experiments folgende Eintraege machen (vgl. EDDAR System Survey, Abschnitt 1.4.2):

- Messagenummer des Experiments: Das Experiment nummeriert die Messages, die es schickt, durch. Dadurch, dass die Eintraege des Experiments im Messagerecord unveraendert zurueckgegeben werden, kann kontrolliert werden, ob der Auftrag ordnungsgemaess erledigt wurde.
- Experimentname: Anhand des Experimentnamen erkennt PETER, von welchem Experiment der Messagerecord stammt.
- Datentyp: Durch den Datentyp wird PETER mitgeteilt, wie

die Daten zu verarbeiten sind. Im Augenblick existieren folgende Datentypen:

- 3 - Leitungstest: Mit dem Leitungstest kann man feststellen, ob PETER bereits laeuft. Wenn man nach einiger Zeit (ca. 10 Min) keine Antwort erhaelt, muss PETER erst gestartet werden.
- 4 - SHUT Experiment: Das Experiment signalisiert PETER, dass es die Kommunikation einstellen moechte.

Bei den bisher genannten Datentypen werden keine Daten, sondern nur eine Message geschickt.

Weitere Datentypen:

- 41 - Senden eines Datenfiles: Hier wird mit dem Messagerecord ein Rohdaten-File geschickt.
- 44 - Senden eines Configuration- und Namelist-Files (das Configurationfile beschreibt die Struktur der Rohdaten-Files; vgl. GALE-Manuals): Hier wird der Messagerecord zusammen mit einem Configuration- und einem Namelist-File geschickt.

Wenn PETER seinen Auftrag erledigt hat, schickt er diesen Messagerecord mit folgenden zusaetzlichen Eintraegen zurueck:

- Messagenummer PETER: Auch PETER nummeriert die Messages, die er schickt, durch.
- Error Code: Falls bei der Verarbeitung der Daten ein

Fehler auftritt, wird die Nummer dieses Fehlers im Error Code zurueckgegeben. Die Bedeutung der Fehlernummer kann man im AMOS-Segment EXPP:INFO.ERRM nachlesen.

- Datum und Zeit

Es kann vorkommen, dass PETER einen Auftrag an den Dauerjob des Experiments schicken muss, der unabhaengig von dem Auftrag ist, den er auszufuehren hatte. Dann setzt PETER einen neuen Messagerecord auf, den er dann, unabhaengig von dem, der ihm geschickt wurde, an das Experiment schickt. In diesen Record traegt er folgendes ein:

- Seine Messagenummer, den Experimentnamen, Datum und Zeit und den

- Datentyp:

21 - MIGRATE muss gestartet werden. Grund: Das OS-File DATA.<expn>, auf dem die konvertierten Daten gespeichert werden, ist voll und kann die zuletzt geschickten Daten nicht mehr aufnehmen. Deshalb muessen die aelteren Daten auf Band ausgelagert werden. Das geschieht mit dem Programm MIGRATE. Damit diese Meldung moeglichst selten erscheint, sollte man MIGRATE nach jedem SHUT (vgl. Datentyp 4) starten. MIGRATE loescht keine Daten von DATA.<expn>, sondern markiert nur die Daten, die auf Band uebertragen wurden, so dass PETER feststellen

kann, welche Daten er bei Ueberlauf von DATA.<expn> ueberschreiben kann. Auch fuer den Fall eines Head Crash sollten moeglichst viele Daten auf Band gerettet sein.

- 66 - Das AMOS-File SDIR(i) muss generiert werden: Das AMOS-File SDIR enthaelt ein Verzeichnis der Daten, die auf Band ausgelagert wurden. Damit SDIR bei einer groesseren Menge von Daten nicht unnuetig umfangreich wird, wurde es in mehrere indizierte Files unterteilt. Sobald ein File ueberlauft, muss an einem AMOS-Terminal mit dem AMOS-Befehl "GENERATE" ein File mit dem naechsten Index generiert werden. Der Index i wird in der Message angegeben.
- 67 - Das AMOS-File SANF(i) muss generiert werden: SANF enthaelt die verschiedenen Versichten der Configuration- und Namelist-Files (vgl. GALE-Manuals). Hier gilt dasselbe wie bei Datentyp 66.
- 68 - SHUT PETER: PETER wird beendet. Die zuletzt geschickten Auftraege werden noch abgearbeitet. Wenn weitere Daten uebertragen werden sollen, muss PETER danach neu gestartet werden. Insbesondere ist zu beachten, dass PETER wegen der Kostenabrechnung um Mitternacht beendet wird.

- Filenummer: enthaelt den Index i bei Datentyp 66 oder 67.

Der Messagerecord hat folgendes Format:

Spalte	Laenge	
0	15	\$\$\$\$E\$R TASKNM (
15	3	Experiment Messagenummer
19	3	PETER Messagenummer
23	3	Errcr Code
27	3	Datentyp
31	3	Experimentname
35	8	Datum (DD-MM-YY)
44	8	Zeit (HH:MM:SS)
53	8	Nummer des zu generierenden AMOS-Files
.56	1)

Die einzelnen Eintraege sind durch Blank zu trennen.

In der WSA-Load-Library existiert das Programm TRANS. Es dient dazu, den Messagerecord und die Daten eines Experiments auf DATA.RECV.<expn> zu schreiben und den Eintrag in DATA.TIME zu machen. Der Dauerjob eines Experiments hat folgende Aufgaben:

1. Messagerecord aufstellen
2. Das Programm TRANS ueber RJE mit folgenden Datenkarten abschicken:
 - entweder nur ein Messagerecord,
 - oder ein Messagerecord, gefolgt von einem Configuration- und Namelist-File,
 - oder ein Messagerecord, gefolgt von einem Datenfile.
3. Den Messagerecord als Antwort von PETER erwarten und anhand dieses Messagerecords entscheiden, was weiter geschehen soll (z.B. naechstes Datenfile schicken oder

SHUT)

Bei dem Experiment W7 wurde ein solcher Dauerjob im Zusammenhang mit RJE implementiert. Eine Beschreibung dazu findet man im Remote Job Entry System User's Guide.

2. Speicherung der Daten und die zugehoerigen Directories

Wie man den GALE-Manuals entnehmen kann, braucht man als Rahmen fuer die Schussdaten und die Beschreibung ihrer Struktur ein Configuration- und Namelist-File. Das Configurationfile liefert die Informationen ueber Diagnostiken und angeschlossene Geraete. Das Namelistfile gibt die Datenart, Offsets und Feldlaengen der Geraetedaten an. Da die Configuration- und Namelist-Files die Struktur der Experimentdaten in der Maschine widerspiegeln, muss bei jeder Aenderung dieses Aufbaus ein neues Configuration- File erzeugt werden. Deshalb gibt es mehrere Versionen der Configuration- und Namelist-Files. Das EDDAR-System speichert diese Information im AMOS-File SANF(i) in folgender Form:

Aus dem Configuration-File wird ein sogenanntes Strukturfile gebildet (SCBD), das neben den Informationen des GALE-Configuration-Files auch die Informationen zur Datenkonversion enthaelt. Sie sind wegen der unterschiedlichen Darstellung von Zeichen und Zahlen in PDP11- und IEM-Rechnern notwendig. Da das Namelistfile nicht nur, wie im GALE-System, zum Aufbau des Configurationfiles dient, sondern auch umgekehrt die Daten wieder ihren Namen zueordnen soll, wurde es mit einer zusaetzlichen Pointerliste

versehen. Im Segment SCBD werden die einzelnen Versionen der Configurationfiles einfach aneinander gehaengt. Eine Pointerliste (VNRD) enthaelt die Anfangsadressen der einzelnen Versionen in SCBD. Die Versionen der Namelist-Files sind in den AMOS-Segmenten NCBD(j) gespeichert. Genauere Informationen ueber das AMOS-File SANF(i) findet man im EDDAR System Survey unter Abschnitt 1.2.3.

Fuer Datenfiles kann nicht, wie im GALE-System, jeweils ein neues File angelegt werden, sondern sie werden nacheinander auf das Random-Access-File DATA.<expn> geschrieben. Deshalb muss ein Verzeichnis (Directory) angelegt werden, aus dem Anfangsadresse und Laenge eines Datenfiles hervorgehen. Dieses Verzeichnis ist in dem AMOS-Segment GINF.DION gespeichert. Ausserdem enthaelt es unter anderem die Information ueber Migration, sowie allgemeine Informationen, wie die 1. freie Adresse auf DATA.<expn>, die Anzahl der auf DATA.<expn> gespeicherten Schuesse usw. Die genaue Organisation kann man dem EDDAR System Survey, Abschnitt 1.2.1 entnehmen.

In dem AMOS-File SDIR(i) befindet sich die Directory der auf Band ausgelagerten Daten. Wenn SDIR(i) ueberlauft, muss diese Directory um ein AMOS-File SDIR(i+1) erweitert werden (vgl. Datentyp 66).

3. Benutzerinformationen in AMOS-Files

PETER fuehrt ueber alle Auftraege, die er von den verschiedenen Experimenten erhaelt, Protokoll. Eine Zeile dieses Protokolls sieht folgendermassen aus:

Spalte	Laenge	
0	5	uebertragene Schussnummer (nur bei Uebertragung von Rohdatenfiles, sonst frei)
7	18	Datum und Zeit
20	6	Jobnummer von PETER
29	3	Experimentname
31	...	Fehlermeldungen (koennen evtl. in der naechsten Zeile fortgesetzt werden)

Welche Bedeutung die einzelnen Fehlernummern haben, die PETER an das Experiment zurueckschickt, kann man in dem AMOS-Segment EXPP:INFO.FRRM nachlesen.

Wenn PETER beendet wird (vgl. Datentyp 68), wird dieses Protokoll auf der REMOTE Station 8 ausgedruckt. Zusaetzlich wird es in das AMOS-Segment EXPP:INFO.PROT geschrieben. Diese beiden Protokolle muss man sich unbedingt ansehen (moeglicherweise ist das Protokoll in EXPP:INFO.PRCT nicht vollstaendig), um sicherzustellen, dass alle Daten ordnungsgemaess uebertragen wurden. Aeltere Protokollzeilen in EXPP:INFO.PROT muessen nach dem Durchsehen geloescht werden, damit dieses AMOS-File nicht ueberlauft.

Um die in das EDDAR-System uebertragenen Daten fuer die Auswerteprogramme der Benutzer zur Verfuegung zu stellen,

wurde die Subroutine GETDAT erstellt. Eine Beschreibung dazu findet man in "Anwendung der Subroutine GETDAT". Fuer den Benutzer der Subrcutine GETDAT ist es von Vorteil, wenn er weiss, ob sich die Daten eines bestimmten Schusses auf Platte oder auf Band befinden. Deshalb wurden die AMOS-Segmente <expn>:INFO.ONLSH und <expn>:INFO.ONWSH erzeugt. Diese Segmente enthalten in aufsteigender Reihenfolge alle Schussnummern der Daten, die sich zu dieser Zeit auf dem Dataset DATA.<expn> bzw. DATA.WORK.<expn> befinden.

Die Manuals des EDDAR-Systems findet man ebenfalls im AMCS-Segment EXPP:INFO:

EDDAR System Survey:	EXPP:INFO.MANS
EDDAR Benutzeranleitung:	EXPP:INFC.EDUSGD
Anwendung der Subroutine GETDAT:	EXPP:INFO.GETUSE

Im Kapitel 4 (System Implementation) der EDDAR-Benutzeranleitung fehlt ein wesentlicher Hinweis:

Auf alle Files, die unter dem User-ID <expn> generiert wurden, muss EXPP Lese- und Schreibzugriff haben. Ausserdem muss jeder, der die Subroutine GETDAT benutzt, auf allen Files Lesezugriff und auf den Files INFO und SDIR(i) Schreibzugriff haben.

AMOS-Befehle dazu:

Lesezugriff:

? FILEORG
? USER EXPP

Schreibzugriff:

? FILEORG
? USER *EXPP

Man kann auch allgemein jedem Benutzer Zugriff geben, indem man in obigen AMOS-Befehlen EXPP durch \$\$\$§ ersetzt.

4. System Implementation

Zunaechst muss man beim Dispatcher ein Benutzerkennzeichen (User-ID) beantragen, das dem Experimentnamen entspricht (wie z.B. WSA, ASD), dazu einen AMOS-Bereich von mindestens 4 Zylindern, und zusaetzlich zwei Baender. In dem AMOS-Bereich muss man mit "GENERATE" die AMOS-Files GINF, SANF(0), SDIR(0) und INFO und mit "CREATE" das Segment SANF(0).SCBD erzeugen.

Weiter muessen die OS-Files DATA.RECV.<expn>, DATA.<expn> und DATA.WORK.<expn> beantragt und als Random-Access-Files in Bloecken von 512 Bytes angelegt werden. DATA.RECV.<expn> muss so gross sein, dass die Daten eines Schusses sicher darauf Platz haben. DATA.<expn> sollte die Daten von etwa 3 Schusstagen fassen, DATA.WORK.<expn> sollte etwa genauso gross sein. Es ist darauf zu achten, dass die CS-Files DATA.<expn> und DATA.RECV.<expn> aller angeschlossenen Experimente auf derselben physikalischen Platteneinheit liegen muessen.

Zum Start des EDDAR-Systems benoetigt man das Programm EDRGEN. Dazu muss man sich ueber RJE die AMOS-Files EXPP:START.EDRGENF, EXPP:START.SCHIEB und EXPP:START.EDRGENJ auf die RSX-11M-Files EDRGEN.FTN, SCHIEB.FTN und EDRGEN.JCL holen. Nun muss man EDRGEN.FTN und SCHIEB.FTN mit FORTRAN (FOR oder F4P) compilieren und anschliessend den Task EDRGEN

bilden (SCHIEB ist ein Unterprogramm). Nun kann man das Programm EDRGEN starten. Dieses Programm fragt unter anderem nach der Groesse der OS-Files und nach den Namen der Baender. Dann werden ueber RJE alle benoetigten AMCS-Segmente erzeugt und vorbesetzt.

Nun kann PETER gestartet werden, indem man das AMOS-Segment EXPP:START.PETER mit SU abschickt (Zeit und CLASS=L angeben!). PETER kann von jedem User-ID aus gestartet werden, aber es ist darauf zu achten, dass PETER nur einmal in der Maschine ist. Der sicherste Weg ist, zuerst einen Leitungstest (vgl. Datentyp 3) zu schicken. Wenn man darauf keine Antwort bekommt, muss PETER gestartet werden. Es ist vorgesehen, dass PETER spaeter nur noch von einem Operator gestartet werden kann.

Wenn eine groessere Menge Daten uebertragen wurde, sollte MIGRATE gestartet werden, um die Daten auf Band zu retten. Die JCL-Karten fuer diesen Job stehen im AMOS-Segment EXPP:START.MIGRATE. Dieses Segment holt man sich ueber FJE in die PDP11 und aendert darin <expn> (vgl. Anhang).

Achtung! MIGRATE und der Dauerjob des Experiments duerfen nicht gleichzeitig laufen!

Wenn das erste Band voll ist, bringt MIGRATE eine Nachricht, dass auf das Reserveband geschrieben wurde. Dann muss man ein neues Band beantragen, das man PETER mit dem Programm

RESBAND zur Verfuegung stellt. Die JCL-Karten holt man sich ebenfalls ueber RJE aus dem AMOS-Segment EXPP:START.RESBAND in die PDP11. Hier muss das Reserveband als DATA-Karte eingegeben werden.

Die Fehlermeldung 55 - Platte nicht freigegeben - kann unter anderem bei abnormalem Ende von PETER auftreten. Mit Hilfe des Programms SCRATCH kann man die Platte danach wieder freigeben. Das Programm SCRATCH sollte von Hand gestartet werden und nicht mit dem Dauerjob des Experiments, denn SCRATCH schickt keine Nachricht zurueck. Der Dauerjob darf erst dann wieder gestartet werden, wenn SCRATCH fertig ist! Die im AMOS-Segment EXPP:START.SCRATCH stehenden JCL-Karten sollte man sich ebenfalls in die PDP11 holen.

5. Checkliste:

- 1.) AMOS-Files GINF, SANF(0), SDIR(0) und INFO generieren.
- 2.) AMOS-Segment SANF(0).SCDB kreieren.
- 3.) Random-Access OS-FILES DATA.RECV.<expn>, DATA.<expn> und DATA.WORK.<expn> anlegen.
- 4.) 2 Baender bereitstellen
- 5.) Program EDRGEN von einer PDP11 aus starten.
- 6.) PETER starten
- 7.) Configuration-File bzw. Schussdaten schicken
- 8.) MIGRATE starten
- 9.) Protokoll im AMOS-Segment EXPP:INFO.PRCT durchsehen und anschliessend den Inhalt des Segments lceschen
- 6.) bis 9.) beliebig oft wiederholen

Anhang

JCL-Karten zu Programm PETER (EXPP:START.PETER)

```

/*ROUTE PRINT REMOTE8
/*CONTROL REST=Y,REGN=300K
// EXEC FOG,LIB=EXPP,NAME=PETER,REGION.G=300K
//G.FT10F001 DD DSN='READ:EXPP:NULLFILE.A;',UNIT=DISK,
// VOL=SER=AMOS,DISP=SHR
//G.FT11F001 DD DSN=DATA.DUM1,UNIT=DISK,VOL=SER=ASDEX1,
// DISP=SHR
//G.FT12F001 DD DSN=DATA.DUM2,UNIT=DISK,VOL=SER=ASDEX1,
// DISP=SHR
//G.FT13F001 DD DSN=DATA.NUL,DISP=SHR
//G.FT20F001 DD DSN='WRITE:WSA:NULLFILE(1).B;',UNIT=DISK,
// VOL=SER=AMOS,DISP=SHR
//G.FT22F001 DD DSN='WRITE:EXPP:NULLFILE(2).C<10.10>;',
// UNIT=DISK,VOL=SER=AMOS,DISP=SHR

```

JCL-Karten zu Programm TRANS (EXPP:START.TRANSN)

```

/*CONTROL REST=Y
// EXEC FOG,LIB=WSA,NAME=TRANS
//G.FT10F001 DD DSN='READ:EXPP:NULLFILE.A;',UNIT=DISK,
// VOL=SER=AMOS,DISP=SHR
//G.FT11F001 DD DSN=DATA.RECV.<EXPN>,DISP=SHR
//G.FT12F001 DD DSN=DATA.NUL,DISP=SHR
//G.FT22F001 DD DSN='WRITE:EXPP:NULLFILE(2).C<10.10>;',
// UNIT=DISK,VOL=SER=AMOS,DISP=SHR
//G.SYSIN DD *

```

JCL-Karten zu Programm MIGRATE (EXPP:START.MIGRATE)

```

/*CONTROL REST=Y,REGN=240K
// EXEC FOG,LIB=EXPP,NAME=MIGRATE,REGION.G=240K
//G.FT10F001 DD DSN='READ:EXPP:NULLFILE.A;',UNIT=DISK,
// VOL=SER=AMOS,DISP=SHR
//G.FT11F001 DD DSN=DATA.RECV.<EXPN>,DISP=SHR
//G.FT12F001 DD DSN=DATA.<EXPN>,DISP=SHR
//G.FT13F001 DD DSN=DATA.NUL,DISP=SHR
//G.FT20F001 DD DSN='WRITE:<EXPN>:NULLFILE(1).B;',
// UNIT=DISK,VOL=SER=AMOS,DISP=SHR
//G.FT22F001 DD UNIT=(T6250,,DEFER),VOL=SER=F219DE,
// LABEL=(,SL),DISP=OLD,
// DCB=(DEN=4,RECFM=FB,LRECL=512,BLKSIZE=12800)

```

JCL-Karten zu Programm RESBAND (EXPP:START.RESBAND)

```
/*CONTROL REST=Y
// EXEC FOG,LIB=WSA,NAME=RESBAND
//G.FT11F001 DD DSN=DATA.RECV.<EXPN>,DISP=SHR
//G.SYSIN DD *
BANDXY
```

JCL-Karten zu Programm SCRATCH (EXPP:START.SCRATCH)

```
/*CONTROL REST=Y
// EXEC FOG,LIB=EXPP,NAME=SCRATCH
//G.FT11F001 DD DSN=DATA.RECV.<EXPN>,DISP=SHR
```

<EXPN> ist jeweils durch den Experimentnamen zu ersetzen.