MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

EDDAR System Survey

J. Steuerwald
Ch. Tichmann

R/34

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 S Y S T E M S U R V E Y

J. Steuerwald
Ch. Tichmann

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

Inhaltsverzeichnis

0. Einfuehrung

0.0 Organisation des Systems ..... 0 - 1
0.1 User Identification ..... 0 - 2
0.2 Flexibilitaet ..... 0 - 3
0.3 Kommunikation mit den Experimenten ..... 0 - 3

1. Beschreibung der Phase I

1.0 Uebersicht ..... 1 - 1
1.1 AMOS-Files im User-Id EXPP ..... 1 - 3
1.1.0 Uebersicht ..... 1 - 3
1.1.1 AMOS-File EXPP:INFO ..... 1 - 3
1.1.2 AMOS-File EXPP:DESC ..... 1 - 4
1.1.3 AMOS-Files EXPP:SOP und CBJ ..... 1 - 5
1.1.4 AMOS-File EXPP:START ..... 1 - 5
1.2 AMOS User Identitification <expname> ..... 1 - 6
1.2.0 Uebersicht ..... 1 - 6
1.2.1 AMOS-File <expname>:GINF ..... 1 - 6
1.2.1.0 Uebersicht ..... 1 - 6
1.2.1.1 Segment DIAGN ..... 1 - 6
1.2.1.2 Segment DION ..... 1 - 7
1.2.1.3 Segment VNRD ..... 1 - 8
1.2.2 AMOS-File <expname>:INFO ..... 1 - 9
1.2.2.0 Uebersicht ..... 1 - 9
1.2.2.1 Segment MSGS ..... 1 - 9
1.2.2.2 Segment ONLSH ..... 1 - 9
1.2.2.3 Segment ONWSH ..... 1 - 10
1.2.3 AMOS-File <expname>:SANF(I) ..... 1 - 10
1.2.3.0 Allgemeine Angaben ..... 1 - 10
1.2.3.1 Segment SCBD ..... 1 - 11
1.2.3.2 Segment NCBD(J) ..... 1 - 12
1.2.4 AMOS-File <expname>:SDIR(J) ..... 1 - 13
1.2.4.0 Organisation des Files ..... 1 - 13
1.2.4.1 Organisation des Segments TAB(K) 1 - 13
1.3 Organisation der Systemtabellen ..... 1 - 16
1.3.0 EDDAR Namelist Dataset ..... 1 - 16
1.3.1 EDDAR Systemdaten ..... 1 - 17
1.3.2 Generieren der Systemdaten ..... 1 - 18

1.4 Random Access Datasets ..... 1 - 19
1.4.0 Uebersicht ..... 1 - 19
1.4.1 Dataset DATA.TIME ..... 1 - 19
1.4.2 Dataset DATA.EXP ..... 1 - 20
1.4.3 Dataset DATA.RECV.<expname> ..... 1 - 22
1.4.4 Dataset DATA.<expname> ..... 1 - 23
1.4.5 Workbereich DATA.WORK.<expname> ..... 1 - 23
1.5 System Implementation ..... 1 - 24

2. Beschreibung der Phase II

2.0 Uebersicht ..... 2 - 1
2.1 Organisation des Zwischenspeichers ..... 2 - 1
2.2 Organisation der on-line Schussdaten ..... 2 - 3
2.3 Directory SHBTN ..... 2 - 5
2.4 Maintenance und Migrate ..... 2 - 6

0. Einfuehrung

0.0 Organisation des Systems

EDDAR ist ein Programmsystem, das die mit Unterstuetzung von Computern erfassten Daten archiviert, verarbeitet und die Ergebnisse dokumentiert, sodass ein gezieltes Retrieval nach Versuchen mit determinierten Eigenschaften moeglich wird. Dabei wurde versucht, eine logische Fortsetzung des Datenerfassungssystems GALE zu konzipieren. Wesentliche Systemteile sind in Listenstruktur entworfen, um ein Hoechstmass an Flexibilitaet zu erreichen. Wegen der Portabilitaet wurden fast alle Programmteile in "IBM System/360 and System/370 FORTRAN IV Language" geschrieben. Fuer wenige Subroutinen konnte aus Effizienzgruenden auf maschinennahe Sprachen nicht verzichtet werden.

Wegen der Speicherung der Roh- und Ergebnisdaten in verschiedenen Ebenen der Informationsdichte wurde eine Unterteilung in vier Phasen gewaehlt.

Phase 1 Erzeugung und Update der Systemtabellen fuer Phase 1, Konversion und Aufbereitung der GALE Namelist Files, Konversion und Archivierung der Rohdaten.

Phase 2 Erzeugung und Update der Systemtabellen fuer Phase 2, Verarbeiten der Rohdaten zu bereinigten Daten, Sammeln und Archivieren der bereinigten Daten.

Phase 3 Erzeugung und Update der Systemtabellen fuer Phase 3, Verarbeiten der bereinigten Daten zu Ergebnisdaten, Sammeln und Dokumentation der Ergebnisdaten.

Phase 4 Programmsystem zum Retrieval mit der Querysprache ERL (EDDAR Retrieval Language, siehe dort) auf den Ergebnisdaten.

Die Auswerteprogramme sind durch das System ADPCS (Automatic Data Processing Control System, siehe dort) in die Phasen 2 und 3 eingebunden. Sie koennen ohne Umstaende geaendert und ausgewechselt und damit der Entwicklung angepasst werden. Die Organisation der Ergebnisdaten erlaubt eine autcmatische Anpassung an Benutzerwuensche.

Alle in diesem Manual beschriebenen Programme, Informationen usw. sind auf Band gerettet.

0.1 User Identification

Fuer jedes Experiment wird ein User-Id, bestehend aus drei zusammenhaengenden Buchstaben, der <expname>, vergeben und ein AMOS-Bereich mit zehn Zylindern angelegt. Allgemeine Informationen zum System EDDAR sind unter der User-Id EXPP zu finden. Darunter zaehlen Manual, Programmbeschreibungen, Programme, Messages usw. Unter LOAD.EXPP und LOAD.WSA existieren Load Libraries der wesentlichen Programme.

0.2 Flexibilitaet

Das Programmsystem EDDAR wurde als logische Fortsetzung des Datenerfassungssystems GALE entworfen. Die Organisation der Rohdaten, die auf EDDAR uebertragen wurde, ist im "GALE System Programmer's Handbook" (Bericht IPP/R-27 v. Feb. 1978) dargestellt. Die zugehoerige GALE Namelist Struktur wurde fuer die Darstellung der Rohdaten uebernommen. Fuer die EDDAR System-, Ergebnisdaten usw. wurde eine analoge Datenorganisation entworfen, die es erlaubt, statt fester Offsets Variable zu verwenden. Damit koennen gewisse Eingriffe in das System allein durch Aenderung uebersichtlicher Listen durchgefuehrt werden, ohne dass deswegen Programme geaendert werden muessen. Dazu findet man weitere Information in Abschnitt 1.3 (Organisation der Systemtabellen).

0.3 Kommunikation mit den Experimenten

Die Kommunikation mit den Experimenten (max. 20) erfolgt ueber den EDDAR Message Controler EMC (siehe dort). Im Rechenzentrum wird er durch das Programm PETER (PErmanent Task EDDAR Receive) repraesentiert, das taeglich bei Beginn des Rechenbetriebes oder nach Aufforderung (durch Nachricht ueber RJE) vom Console Operator gestartet und automatisch kurz vor 24.00 Uhr abgebrochen wird. Beim Experiment kann beliebig verfahren werden. Das Programm EDRRCV (Gegenstueck

zu PETER) wird fuer die PDP-11 zur Verfuegung gestellt, um die ausgetauschten Messages ordnungsgemaess zu verarbeiten. Im anderen Falle muss durch ein geeignetes Programm die Organisation des Messagerecord (siehe Abschnitt 1.4.3) und die durch seinen Inhalt angeforderten Reaktionen buchstabengetreu erfuellt werden. Das Programm PETER ruft die Programme

RECV - Konversion und Archivierung der Rohdaten
STRGEN - Konversion und Archivierung des GALE Namelist-File und Erzeugen Strukturvektoren
PHAS2 - Sammelprogramm fuer EDDAR Phase II.

(das System wird noch erweitert) und erledigt Auftraege, wie Annahme von Dauerjobs innerhalb des Auswertesystems ADPCS (siehe dort), Update von Systemdaten, Erzeugung und Initialisierung von AMOS-Segmenten als Systemdaten usw.

1. Beschreibung der Phase 1

1.0 Uebersicht

Im Abschnitt 0.0 wurden der Phase 1 des Programmsystems EDDAR folgende Taetigkeiten zugeordnet:

- Erzeugung und Update der Systemtabellen,
- Konversion und Aufbereitung der GALE Namelist Files,
- Konversion und Archivierung der Rohdaten.

Das EDDAR Namelist File, das die Organisation der Systemdaten beschreibt, wird durch Programm GENSD mit den Daten des AMCS-Segments EXPP:START.CLIST im AMCS-Segment EXPP:CBJ.CLIST erzeugt. Es dient als Grundlage fuer die EDDAR System Tabelle LIST im AMOS-SEGMENT EXPP:OBJ.LIST mit Programm GENTB und den Daten im AMOS-Segment EXPP:START.VLIST. Das Update der Daten in LIST und VLIST wird manuell in beiden Segmenten, in der Reihenfolge uebereinstimmend, durchgefuehrt.

Zur Uebertragung der Daten mit RJE steht das Programm TRANS zur Verfuegung. TRANS uebertraegt eine Message oder eine Message mit Daten an den Dauerjob PETER.

Schliesslich wird der on-line Datenbestand jedes Experiments bei Bedarf mit Programm MIGRATE automatisch auf Band gerettet. Der Nachschub an Migration Tapes erfolgt mit Programm RESBAND, das mit EXPP:START.RESBAND gestartet werden kann. Man gibt als DATA-Karte das neue Band an, das

dann in den Plattenbereich DATA.RECV.<expname> uebertragen wird. Dieses Programm ist zu starten, sobald MIGRATE die Nachricht bringt, dass das Reserveband verwendet wurde. Die Subroutine GETDAT uebertraegt die Rohdaten in Bereiche der Benutzerprogramme.

Weitere Dienstprogramme zum Listen von Uebersichten (Diagnostik-Verzeichnis, angeschlossene Diagnostiken, Schuesse eines Tages, Listen von Schussdaten usw.) sind in Entwicklung und werden im AMOS-Segment EXPP:INFO.MANU nachgetragen.

1.1 AMOS-Files im User-Id EXPP

1.1.0 Uebersicht

In der User-Id EXPP sind alle die Auswertung von Experimentdaten allgemein betreffenden Informationen zusammengefasst. Dazu gehoeren die Files

- INFO - allgemeine Information fuer die Auswertung aller Experimente
- DESC - Programmbeschreibungen
- SOR - Source Decks
- OBJ - Object Decks
- START - Job Setup's fuer die Auswertung der Experimentdaten.

1.1.1 AMOS-File EXPP:INFO

Die Segmente dieses Files

- MSGS - Messages, alle Experimente betreffend
- MANS(I) - EDDAR System Survey (i=1,...,4)
- MANI - EDDAR User Information (HELP)
- MANU - EDDAR Utilities (Uebersicht und naehere Angaben)
- RAHMEN - Rahmen fuer Programmbeschreibung (nach EDDAR Muster)
- ERRM - Error Messages des Systems EDDAR
- LOAD - Inhalt der LOAD.EXPP Library
- PROT - Protokoll des Ablaufs von PETER

sind editierbar.

Die bei der Verarbeitung der Experiment Messages entstehen-

den Nachrichten werden in dem AMOS-Segment EXPP:INFO.PROT in der Form

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

gesammelt. Sofern die Zahl der Fehlermeldungen 14 ueberschreitet, wird die letzte Fehlernachricht auf 999 - sonst ohne Bedeutung - gesetzt.

Mit PRINT des AMOS-Segments kann man die Nachrichten am Ende eines Schusstages ausdrucken und danach sollten sie, um den Umfang zu beschraenken, geloescht werden. Mit dem Dauerjob PETER wird die jeweils letzte Nachricht angehaengt. Aus dieser Tabelle kann mit Subroutine GETSHT die letzte uebertragene Schussnummer erhalten werden.

1.1.2 AMOS-File EXPP:DESC

Das AMOS-File EXPP:DESC enthaelt die Segmente

EDA - Uebersicht ueber die Programme der EDDAR Programm-Bibliothek

<prog> - mit der Beschreibung des im AMOS-Segment EXPP:SOR.<prog> gespeicherten Programms mit Namen <prog>.

EDA enthaelt Namen und Zweck aller in EXPP:SOR bzw. EXPP.OBJ abgelegten Programme. Bei Uebernahme eines Programms <prog> mit Beschreibung nach EDDAR Muster, im AMOS-Segment

EXPP:INFO.RAHMEN dargestellt, mit Programm BBLS, werden zwei Segmente, naemlich EXPP:SOR.<prog> und EXPP:DESC.<prog>, erzeugt und AMOS-Segment EXPP:DESC.EDA auf den neuesten Stand gebracht. Alle Segmente sind editierbar.

1.1.3 AMOS-Files EXPP:SOR und EXPP:OBJ

Diese Files enthalten die vollstaendige, in EXPP:DESC.EDA und den Segmenten EXPP:DESC.<prog> beschriebene Source- bzw. Object-Library.

Die AMOS-Segmente EXPP:OBJ.OLIST und VLIST haben spezielle Bedeutung, sie sind im Abschnitt 1.3 naeher erklaert.

1.1.4 AMOS-File EXPP:START

In diesem File sind alle Job Setup's und die Daten fuer die Auswertung von Experimentdaten abgelegt. Sofern in JCL-Records statt EXPP der String '<expname>' eingesetzt ist, kann dieser Text mit Hilfe des Programms FEIST durch den aktuellen Experimentnamen, eingegeben in den ersten drei Bytes eines einzulesenden Records, ersetzt werden. Die Uebertragung in ein vom Benutzer angegebenes AMOS-Segment (vom vierten Zeichen des einzulesenden Records ab ':<filename>.<seg-name>') wird, falls nicht leer, ebenfalls durchgefuehrt.

Die Segmente EXPP:START.OLIST und VLIST haben spezielle Bedeutung, sie sind im Abschnitt 1.3 naeher erklaert.

1.2 AMOS User Identification <expname>

1.2.0 Uebersicht

Fuer jedes Experiment wird ein AMOS-Bereich mit der User Identification (s. auch Abschnitt 0.1) <expname> eingerichtet. Er enthaelt die Files

- INFO - allgemein interessierende Information innerhalb des Experiments
- GINF - experimenteigene Systemtabellen
- SANF(I) - GALE Namelist Files (indiziert 1...99)
- SDIR(J) - off-line Shot Directory (indiziert 1...99).

1.2.1 AMOS-File <expname>:GINF

1.2.1.0 Uebersicht

Zur allgemeinen Information existieren die Segmente

- DIAGN - Verzeichnis der Diagnostiken (ed.),
- DION - Directory of on-line Shots,
- VNRD - Version Number Directory.

Die Organisation der genannten Segmente wird in den folgenden Abschnitten beschrieben. Die editierbaren, mit (ed.) gekennzeichneten, Segmente koennen am Bildschirm gelesen und geaendert werden. Die anderen Segmente werden stets als ein Record behandelt (mit IPPGET gelesen, mit IPPPUT ausgegeben), Update ist nur mit Programm moeglich.

1.2.1.1 Segment DIAGN

**** muss noch entworfen werden ****

1.2.1.2 Segment DION

Es nimmt die Information ueber die on-line Experimentdaten auf. Der Header

0	2	Gesamtlaenge in Bytes
2	2	frei
4	2	Pointer auf den Anfang der Pointerliste in Halbworten
6	2	Laenge des logischen Records in Bytes
8	2	Anzahl der logischen Records im Header
10	2	Anzahl der on-line Schuesse
12	2	Anzahl der archivierten Schuesse
14	2	niedrigste Schussnummer insgesamt
16	2	hoechste Schussnummer insgesamt
18	2	niedrigste Schussnummer on-line
20	2	hoechste Schussnummer on-line
22	2	Adresse des zuletzt beschriebenen Items in Ganzworten
24	4	Adresse Anfangsblock im Maintenance Bereich
28	4	Adresse Endblock im Maintenance Bereich
32	4	Adresse Anfangsblock im freien Bereich
36	4	Adresse Endblock im freien Bereich
40	1	Maintenance Byte
41	1	true, falls Platte schon einmal bei der zyklischen Speicherung beschrieben wurde
42	3	Datum der Eroeffnung
45	3	Uhrzeit der Eroeffnung
48	3	Datum der letzten Aenderung
51	3	Uhrzeit der letzten Aenderung
54	2	Nummer des zuletzt uebertragenen Schusses
56	16	frei

enthaelt die allgemeinen Daten. Ihnen folgen die Reccrds der on-line Shot Daten. Die Laenge des logischen Records ist im Offset 6 des Headers gegeben (z. Zt. 24 Bytes). Die logischen Records bestehen aus

0	2	Schussnummer
2	1	Maintenance Byte
3	1	frei
4	4	Adresse des ersten Blocks
8	8	angeschlossene Diagnostiken

16	3	Datum des Schusses
19	3	Uhrzeit des Schusses
22	2	Laenge der Schussdaten in Bloecken.

Zur Aufnahme von n Schuessen ergibt sich die Laenge l dieses Segmentes mit den Bezeichnungen

hdl - Anzahl der logischen Records im Header
sdr - Laenge des logischen Records in Bytes

aus der Vorschrift

l := (n + hdl) * sdr.

Bei der derzeitigen Wahl n = 450 folgt somit l := 10872 Bytes. Daran schliesst sich eine Pointerliste mit 1116 INTEGER*2 an, sodass insgesamt 13104 Bytes belegt werden. Das Segment DION wird mit Programm VNRDION generiert und von den Programmen RECV und MIGRATE benutzt und jeweils auf den neuesten Stand gebracht.

1.2.1.3 Segment VNRD

Dieses Segment enthaelt die Version Number Directory der GALE Namelist Files. Sie besteht aus dem Header:

0	2	Gesamtlaenge in Bytes
2	4	frei
6	2	Laenge des logischen Records in Bytes
8	2	Anzahl logischer Records im Header
10	2	Anzahl Items gesamt
12	2	hoechste Versionsnummer
14	2	Laenge des logischen Records in SCBD (SRECL)
16	2	Anzahl logischer Records im Header des SCBD (HEADL)
18	3	Datum der letzten Aenderung
21	3	Uhrzeit der letzten Aenderung
24	4	frei

und den Items:

0	2	Versionsnummer
2	2	Filenummer i in SANF [SANF(i)]
4	2	Adresse des Beginns dieses Teils des SCBD in physikalischen Records
6	2	Laenge dieses Teils des SCBD in Bytes
8	3	Datum der Generation
11	3	Uhrzeit der Generation.

Das Segment VNRD wird mit Programm VNRDION generiert und mit Programm STRGEN durch GALE Configuration und Namelist File auf den neuesten Stand gebracht. Das Segment nimmt maximal 231 Versionen bei 3240 Bytes Laenge auf.

1.2.2 AMOS-File <expname>:INFO

1.2.2.0 Uebersicht

Dieses File enthaelt allgemein informierende Angaben:

MSGS - Nachrichten im Austausch mit dem Rechenzentrum, nur das Experiment betreffend,

ONLSH - Verzeichnis der on-line Shot Numbers

ONWSH - Verzeichnis der Shot Numbers, die auf DATA.WORK.<expnam> stehen

1.2.2.1 Segment MSGS

In diesem Segment werden innerhalb des Experiments oder falls nur das relevante Experiment betroffen ist, von EXPP her Nachrichten ausgetauscht. Das Segment ist editierbar.

1.2.2.2 AMOS-Segment ONLSH

In dieses Segment werden nach Uebernahme eines GALE Datenfiles mit Programm RECV die Shot Numbers der on-line

Data Files in editierbarer Form (12I6 je Zeile) ausgegeben.

1.2.2.3 AMOS-Segment ONWSH

In diesem Segment werden die Shot Numbers derjenigen Datenfiles in editierbarer Form (12I6 je Zeile) ausgegeben, die sich auf dem OS-File DATA.WORK.<expname> befinden (vgl. Abschnitt 1.4.5).

1.2.3 AMOS-File <expname>:SANF(I)

1.2.3.0 Allgemeine Angaben

Der Index I ergibt sich mit den Bezeichnungen

vnr - Versionsnummer,

segnz - Anzahl Segmente in einem AMOS-File

aus der Vorschrift

$$i := vnr / (segnz - 5) .$$

Jedes File besteht aus dem Header SCBD (Structure Control Block Directory) mit maximal vier AMOS-Bloecken und den Segmenten NCBD(J) (Namelist Control Block Data), jeweils aus einem AMOS-Block bestehend. Dabei wird der Index J aus

$$j := vnr - i * (segnz - 5)$$

ermittelt. In die Segmente NCBD(J) werden die konvertierten GALE Namelist Files uebertragen. Im I/O-Handling werden alle Segmente NCBD(J) durch IPPGET und IPPPUT als ein Block behandelt.

1.2.3.1 AMOS-Segment SCBD

Das Header-Segment jedes Files enthaelt die allgemeinen Angaben:

0	2	Gesamtlaenge in Bytes
2	2	Anzahl der Kennziffern
4	2	Niedrigste Kennziffer
6	2	Hoechste Kennziffer
8	2	Versionsnummer
10	2	frei

und die Items

0	2	Kennziffer
2	2	Offset des Datenbeginns in Bytes
4	2	Startknoten im Namelist File in logischen Records

in der Anzahl, die durch Offset 2 im Header gegeben ist.

Die Structure Lists beginnen jeweils am Anfang eines physikalischen Records (72 Bytes) und sind in folgender Art organisiert:

0	2	Kennziffer
2	2	Anzahl Datentypen (n),

gefolgt von den n Items der Datentypen

0	2	Nummer des Datentyps
2	2	Anzahl der Gruppen [m(i)].

Die m(i) Items der Gruppen der Datentypen bestehen aus

0	2	Offset im Control Block
2	2	Datenlaenge in Bytes

deren Anzahl durch Offset 2 gegeben ist. Die Laenge des SCBD in Bytes ist im Offset 6 im Item der relevanten Versionsnummer in VNRD abgelegt.

Das Segment SCBD wird stets in Einheiten von physikalischen

Records (72 Bytes) gelesen und geschrieben.

1.2.3.2 AMOS-Segment NCBD(J)

Am Anfang dieses Segments werden die Pointer auf die zu einer Diagnostik gehoerenden Namelists gespeichert:

i-1	2	Pointer auf das Item der Diagnostik i (i=1,256)

Der Header enthaelt allgemeine Angaben:

0	2	Laenge der Tabelle in Bytes
2	4	frei
6	2	Laenge des logischen Records in Bytes
8	2	Anzahl logischer Records im Header
10	2	frei
12	2	Anzahl angeschlossener Diagnostiken
14	2	Pointer auf Anfang der Namelists

und die Items:

0	1	Anzahl Gruppen (i)
1	1	Diagnostiknummer
2	2	frei
4	2	Pointer auf zugehoeriges Namelist
8	4*i	Gruppen, bestehend aus:

0	2	Module ID
2	2	Pointer auf zugehoeriges Namelist

Danach folgen die Namelists. Der Variablenname eines NCB wird wegen der Umkodierung von RADIX-50 auf EBCDIC Code um zwei Zeichen verlaengert, er beginnt somit im Offset N.HLP statt in N.NAM. Die uebrige Organisation entspricht Kapitel 2.5 (Namelist Structure, page 2-12) aus dem "GALE System Programmer's Handbook".

1.2.4 AMOS-File <expname>:SDIR(J)

1.2.4.0 Organisation des Files

Die Aufzeichnungen der Shot Directory sind in Segmenten TAB(K) von der Laenge eines AMOS-Blocks abgelegt. Fuer jede natuerliche Zahl, beginnend mit der niedrigsten Schussnummer, wird ein logischer Record angelegt, gleichgueltig, ob Daten fuer den relevanten Schuss vorhanden sind oder nicht. Damit koennen die Pointer ohne Tabellensuche ermittelt werden.

1.2.4.1 Organisation der Segmente TAB(K)

Sie bestehen aus dem Header

0	2	Gesamtlaenge in Bytes
2	4	frei
6	2	Laenge des logischen Records in Bytes
8	2	Anzahl logischer Records im Header
10	2	Anzahl Items der Tabelle
12	2	frei
14	2	niedrigste Schussnummer der Tabelle
16	2	hoechste Schussnummer der Tabelle
18	3	Datum der letzten Aenderung
21	3	Uhrzeit der letzten Aenderung

und den Items der Off-line Shot Directory, deren Anzahl durch den Offset 10 vorgegeben ist. Jedes Item enthaelt drei Gruppen mit Aufzeichnungen ueber die Adressen von

0	24	Rohdaten
24	24	bereinigten Daten
48	24	Ergebnisdaten

auf Baendern. Die einzelnen Gruppen geben Aufschluss ueber:

0	8	angeschlossene Diagnostiken
8	3	Datum des Schusses
11	3	Uhrzeit des Schusses
14	2	Laenge der Schussdaten in Bloecken
16	6	Bandnummer
22	2	Filenummer.

Die Schussnummern werden fortlaufend gefuehrt. Item 1 von SDIR(0).TAB(0) beginnt mit der niedrigsten Schussnummer s(0). Fuer die Ermittlung der Indizes j und k gilt daher fuer die Schussnummer s mit den Bezeichnungen

- blocl - Laenge des AMOS-Blocks in Bytes
- recl - Laenge des physikalischen Records in AMOS in Bytes
- srecl - Laenge des logischen Records (Offset 6)
- headl - Anzahl logischer Records im Header
- segnz - Anzahl Segmente eines AMOS-Files
- j - Index des Files
- k - Index des Segments
- l - Pointer auf den Anfang des logischen Records fuer die Schussnummer s im AMOS-Segment <expname>:SDIR(J).TAB(K)

die Rechenvorschrift

```
div := blocl / (recl + 4) * recl / srecl - headl
diw := (segnz - 1) * div
anz := s - s(0) + 1
j   := (anz - 1) / diw
anz := anz - j * diw
k   := (anz - 1) / div
anz := anz - k * div
l   := (anz - 1 + headl) * srecl + 1 .
```

Das Programm MIGRATE erzeugt mit Beginn des 20. Segments im AMOS-File SDIR(n) eine Message, die als

"AMOS-File SDIR(n+1) erzeugen"

interpretiert wird. Nach Pruefung der Uebereinstimmung der angeforderten File-Nummer durch Command FLIST, muss das

entsprechende AMOS-File so bald wie moeglich erzeugt werden. Die Initialisierung der Segmente erfolgt danach automatisch.

1.3 Organisation der Systemtabellen

1.3.0 EDDAR Namelist Dataset

Die Struktur der Systemdaten wird durch den EDDAR Namelist Dataset (ENDS) im AMOS-Segment EXPP:OBJ.OLIST beschrieben. Er besteht aus dem Header

0	1	Datentyp = 0
1	1	Laenge des Namens
2	6	Name der Tabelle
8	1	Offset des Namens
9	1	Laenge des logischen Records in Bytes
10	2	Laenge des Datensatzes in logischen Records

und den EDDAR Namelist Blcecken (ENB) mit

0	2	Pointer zur Pointerliste
2	6	Name des Offsets
8	1	Datentyp
9	1	Laenge des Datenfeldes
10	1	Offset innerhalb der Gruppe
11	1	Gruppe in der Tabelle.

Die Datentypen beschreiben die Laenge der Datenitems

1 - EBCDIC-Zeichen in ununterbrochener Folge
2 - INTEGER*2
4 - INTEGER*4.

Das Segment besteht aus vier Gruppen von Daten

- logische Einheiten, Datenumfang, EDDAR System Daten,
- GALE System Daten,
- gemischte Daten,
- alphanumerische Daten zum Aufruf von AMOS-Segmenten.

Jede Gruppe enthaelt einen oder mehrere Records von 72 Bytes Laenge. Dieses Segment wird durch das Programm GENTB mit Commands und Dateneingabe in EXPP:START.GENTB generiert.

Update wird manuell am Bildschirm vorgenommen. Man beachte die Uebereinstimmung in der Reihenfolge der Namen in den AMOS-Segmenten EXPP:START.OLIST und VLIST. Sofern die Bootstrapdaten in VLIST auf mehr als einen Record ausgedehnt werden, ist dies im Programm OPNJB entsprechend zu beruecksichtigen.
Daten aus diesem Segment werden den Variablennamen durch die Subroutinen OTON und OTNE zugeordnet.

1.3.1 EDDAR Systemdaten (EXPP:START.OLIST)

Um Aenderungen transparent zu machen, sind die Daten zu ENDS im AMOS-Segment EXPP:START.OLIST editierbar (s. Anlage I). Jeder logische Record nimmt eine Zeile am Bildschirm ein. Der Header

0	1	Datentyp = 0
1	1	frei
2	6	Name der Tabelle
8	2	Offset des Namens
10	2	Laenge des Namens
12	3	Recordlaenge des ENB in Bytes
15	3	Offset des Datentyps
18	3	Offset der Datenlaenge
21	3	Offset des Offsets

wird von den Items

0	6	Name des Offsets
6	4	Datentyp
10	4	Laenge des Datenfeldes
14	4	Offset innerhalb der Gruppe
18	2	Gruppe in der Tabelle

gefolgt. Dabei enthalten alle Entities ab Offset 8 mindestens ein fuehrendes Blank. Mit Programm GENSD wird das

EDDAR Namelist File erzeugt.

1.3.2 Generieren der Systemdaten

Die erzeugenden Daten sind in EXPP:START.VLIST und wegen der Transparenz in editierbarer Form (s. Anlage II) gespeichert.

Die Headerzeile enthaelt

0	1	Datentyp = V
1	1	frei
2	6	Name der Tabelle.

Auf sie folgen die Offset Items

0	6	Name des Offsets
6	1	frei
7	x	Daten der zugelassenen Datentypen in freier Notation mit mindestens 1 Blank als Trennung.

Zugelassene Datentypen sind

1 - EBCDIC-Zeichen in ununterbrochener Folge
2 - INTEGER*2
4 - INTEGER*4.

Die Offset Items koennen bis zu zwei Bildschirmzeilen beanspruchen. In Spalte 72 wird die Fortsetzung der Daten in einer weiteren Zeile durch ein beliebiges signifikantes Zeichen vermerkt. Die Eingabedaten muessen daher spaetestens in Spalte 71 enden. Mit Hilfe des Programms GENTB wird das AMOS-Segment EXPP:OBJ.LIST erzeugt. Wird der Headerrecord ueber 72 Zeichen hinaus ausgedehnt, so ist das Programm OPNJB entsprechend zu aendern.

1.4 Die Random-Access Datasets

1.4.0 Uebersicht

Alle Datasets, ausser dem EDDA'R System Dataset DATA.EXP, bestehen aus Bloecken mit 512 Bytes. Die vcm Experiment ueberspielten Messages und Daten werden im Dataset DATA.RECV.<expname> empfangen. GALE Configuration und Namelist File werden verarbeitet und in AMOS-Files <expname>:GINF.VNRD, <expname>:SANF(I).SCBD und <expname>:SANF(I).NCBD(J) gespeichert. Die Rohdaten hingegen werden im Dataset DATA.<expname> zyklisch gespeichert. Nach Abschluss jedes Schusstages oder auf Anforderung wird eine Kopie auf Band abgelegt.

Fuer die Verarbeitung ausgelagerter Schuesse wird ein Plattenbereich DATA.WORK.<expname> benutzt.

1.4.1 Dataset DATA.TIME

(Direct Access File mit 1 Track)

Der Dataset enthaelt

0	120	Datum und Uhzeit der Messages

Dieser Dataset enthaelt pro Experiment (20 sind vorgesehen) ein Item mit

0	3	Datum (Jahr,Monat,Tag)
3	3	Uhrzeit (Stunde,Minute,Sekunde)

Wenn ein Experiment Daten sendet, traegt es in sein Item die Uhrzeit ein. PETER kann also anhand dieses Datasets

erkennen, zu welcher Zeit ein Experiment eine Anforderung gestellt hat und diese entsprechend verarbeiten.

1.4.2 Dataset DATA.EXP

(Direct Access File mit 2 Tracks, insgesamt 50 Records zu 120 Bytes).

Der Dataset enthaelt

0	240	Bootstrapdaten, Experiment- und Aliasnamen
240	1200	statistische Daten
1440	240	DS-Name Blocks
1680	120	frei
1800	40	REMOTE Nummern der Experimente
1840	80	frei
1920	20	Openbytes der Experimente
1940	2100	frei
4120	240	Zaehler fuer Wartezeiten.
4320	240	Bootstrapdaten fuer STRGEN

Die Bootstrapdaten bestehen aus (Bedeutungen, falls nicht naeher erklaert, in EXPP:START.OLIST)

0	2	Laenge des Records in Bytes
2	2	LUNAM
4	2	RECL
6	2	LLSTLG
8	2	LNAMLG
10	2	NNMLG (Laenge Datum DATL)
12	2	Anzahl der angeschlossenen Experimente (NEXP)
14	2	Messagezaehler (MSGZ)
16	2	RNMR (Nummer des Message-Records)
18	1	ITIM (Intervall fuer Wait in sec)
19	1	POPNB (PETER Open Byte)
20	2	EXPZ (Anzahl angeschl. Experimente)
22	2	EXPMAX (max. Zahl angeschl. Experimente)

Offsets im Messagerecord

24	2	Message Nummer Experiment

26	2	Message Nummer EDDAR
28	2	Error Code
30	2	Message Type Experiment
32	2	Message Type EDDAR
34	2	Experiment Name
36	2	Date
38	2	Time

Adressen im Experimentblock

40	2	Laenge Experimentblock
42	2	Experimentname
44	2	Mehrfach-Dauerauswertung
46	2	Dauerauswertung
48	2	SDIRGEN Feld
50	2	VNRDION Feld
52	2	MIGR Feld I
54	2	MIGR Feld II
56	2	MIGR Feld III
58	2	Openbyte
60	2	SANF-Block
62	6	frei

DS-Name Block

68	2	Laenge DSN-Block
70	2	Anzahl DSN-BLOECKE je Exp.
72	8	frei
80	80	20 Items fuer Experimentnamen
160	80	20 Items fuer Aliasnamen der Experimente

Im DSN-Bereich gilt:

0	13	DSN fuer Empfangsplatte
13	13	DSN fuer Rohdatenplatte
26	13	DSN fuer Workplatte
39	13	DSN fuer Empfangsplatte Phase II
52	13	DSN fuer Platte der bereinigten Daten
65	13	DSN fuer Empfangsplatte Phase III
78	13	DSN fuer Platte der Ergebnisdaten
91	39	frei.

Die Bootstrapdaten fuer STRGEN bestehen aus

54 Integer*2 fuer die vorgegebenen Strukturvektoren
9 Integer*2 fuer die Kennziffern 1 ... 3

1.4.3 Dataset DATA.RECV.<expname>

Dieser Datensatz enthaelt einen Header Block mit

0	4	Anzahl uebertragener Bloecke
4	3	Experiment Name
7	1	Nummer des Messagerecords
8	4	SIZE
12	4	RSIZ
16	4	WSIZ
20	6	Bandnummer aktuell
26	2	Filenummer aktuell
28	4	Zahl freier Bloecke (Band aktuell)
32	6	Bandnummer Reserve
38	2	MAXSH
40	4	ZAEHL (Bloecke fuer Migration)
44	2	EXPZHL (min. Anzahl Bloecke fuer Migration)
46	2	Mehrfach-Dauerauswertung Message Nummer
48	3	Anfangsdatum
51	3	Enddatum
54	2	Dauerauswertungs Message Number
56	1	SDIRGEN Control Byte
57	1	SDIRGEN Filenummer
58	3	SDIRGEN Control Date
61	1	VNRDION Control Byte
62	1	VNRDION Filenummer
63	3	VNRDION Control Byte
66	2	frei
68	1	MIGR Control Byte I
69	3	MIGR Control Date I
72	1	MIGR Control Byte II
73	3	MIGR Control Date II
76	1	MIGR Control Byte III
77	3	MIGR Control Date III
80	2	frei
82	1	Filenummer SANF
83	1	Segmentnummer in aktivem SANF
84	2	belegte Bloecke gesamt
86	2	belegte Bloecke in SCBD
88	2	belegte Bloecke in allen NCBD
90	294	frei
384	64	Messagerecord
448	64	frei zum Umspeichern

wobei der Messagereccrd folgendermassen aufgebaut ist:

0	15	$$$EDR TASKNM (

15	3	Experiment Messagenummer
19	3	PETER Messagenummer
23	3	Error Code
27	3	Datentyp
31	3	Experimentname
35	8	Datum (dd-mm-yy)
44	8	Zeit (hh:mm:ss)
53	3	Nummer des zu generierenden AMOS-Files
56	1	)

und die anschliessenden GALE Daten-Bloecke in PDP-11 Notation. Die Groesse des Bereichs wird der im Experiment bei vollem Betrieb aller Diagnostiken erwarteten Datenmenge fuer einen Schuss angepasst.

1.4.4 Dataset DATA.<expname>

Die Groesse dieses Datenfiles wird so bestimmt, dass mindestens die Rohdaten von drei Schusstagen aufgenommen werden koennen. Die konvertierten Daten werden zyklisch im FIFO Modus gespeichert. Die Directory dieses Bereichs ist im AMOS-Segment <expname>:GINF.DION abgelegt.

1.4.5 Workbereich DATA.WORK.<expname>

Die Verarbeitung ausgelagerter Daten wird ueber den Plattenbereich DATA.WORK.<expname> abgewickelt. Seine Groesse wird gemaess

- vorhandenem Speicherplatz
- der Intensitaet der Benutzung und
- der Organisation des Experiments

festgelegt.

Die vom Band eingelesenen off-line Shots werden auf Platte

gebracht, um Random Access zu ermoeglichen. Diese Work Area enthaelt eine Directory , die mit Programm EDRGEN initialisiert wird, bestehend aus dem Header

0	4	Groesse des Plattenbereiches
4	3	Experimentname
7	1	frei
8	4	letzte belegte Adresse
12	2	Anzahl der Datasets
14	2	Anzahl Bloecke der Directory

und den Items

0	2	Schussnummer
2	2	Laenge des Schusses in Bloecken
4	4	Anfangsadresse.
8	6	Datum, Uhrzeit
14	2	frei

Die Organisation der eingelesenen konvertierten Rohdaten entspricht den GALE Konventionen, man kann mit den ueblichen Utilities zugreifen.

1.5 System Implementation

Es sind folgende Punkte durchzufuehren:

- beim Dispatcher Platz fuer die Random Access Data Sets DATA, DATA.RECV und DATA.WORK besorgen
- beim Dispatcher User Id fuer AMOS-Bereich mit 10 Zylindern beantragen,
- Generation der Files (GINF, SANF(0), SDIR(0), INFO)
- Kreieren des Segments SANF(0).SCBD
- von einer PDP11 aus Programm EDRGEN mit den notwendigen Parametern (werden abgefragt) starten.

2. Beschreibung der Phase 2

2.0. Uebersicht

In der Phase 2 des Programmsystems EDDAR werden die Rohdaten in parallel arbeitenden Prozessen (ADCPS) zu bereinigten Daten verarbeitet. Die Ergebnisse werden schussweise in dem Random-Access File DATA.INTZ.<expname> zwischengespeichert. Von dort werden sie mit Programm PHAS2 in das Random-Access File DATA.BERG.<expname> uebertragen. Die Bereinigung der Rohdaten wird nicht zwingend in einem Zuge durchgefuehrt. Andererseits soll der zur Verfuegung stehende Speicherplatz optimal genutzt werden. Daraus ergibt sich folgende Organisation:

- Directory SHTBN fuer on-line Daten der Phase 2,
- Directory SDIR(I).TAB(J) fuer off-line Daten der Phase 2 (s. Abschnitt 1.2.5),
- on-line Schussdaten DATA.BERG.<expname>,
- off-line Schussdaten auf Magnetbaendern,
- Zwischenspeicher DATA.INTZ.<expname>.

Ein Maintenance Programm wird taeglich gestartet und durchgefuehrt, sofern in der on - line Shot Directory SHTBN das Maintenance Byte gesetzt ist.

2.1. Organisation des Zwischenspeichers

Die Prozesse liefern die bereinigten Daten eines einzelnen Schusses in feststehender Organisation an den

Zwischenspeicher DATA.INTZ.<expname> ab. Die Directory nimmt ausser dem Header

0	2	Laenge Header in Bytes
2	1	Processing Code
3	3	Experimentname
6	2	Laenge des logischen Records in Bytes
8	2	Anzahl logischer Records im Header
10	2	Schussnummer
12	2	Anzahl bereinigter Diagnostiken
14	3	Datum der Auswertung
17	3	Uhrzeit der Auswertung
20	8	Anzahl angeschlossener Diagnostiken
28	2	Versions-Nummer des Namelist File
30	6	Jobnummer
36	28	frei

die Items

0	1	Diagnostik-Nummer
1	1	Processing Code
2	2	Pointer auf die Daten der Diagnostik in logischen Records
4	2	Pointer auf Anfangsknoten im Namelist File in logischen Records
6	2	frei
8	4	Laenge der Daten in Bytes
12	1	Laenge Daten-Item
13	1	Art des Datentyps
14	2	Nummer des Eichschusses
16	2	Programmnummer
18	14	frei

aller moeglichen in den Rohdaten angeschlossenen Diagnostiken auf. Der Processing Code im Header hat nur die Bedeutungen TRUE und FALSE. Im Diagnostik-Item jedoch muss zwischen 1 und 2 unterschieden werden koennen. Dabei wird 2 dann benutzt, wenn Daten einer Diagnostik nach einer geaenderten Auswertung ueberschrieben werden sollen. Fuer derzeit 40 Diagnostiken sind 2 Bloecke vorgesehen. Der Platz

fuer die bereinigten Daten wird entsprechend der Menge der maximal in der jeweiligen Diagnostik erwarteten Daten bestimmt.

2.2. Organisation der on-line Schussdaten

Die bereinigten Daten werden mit Programm PHAS2 vom Random-Access File DATA.INTZ.<expname> uebernommen und sequentiell, beginnend mit dem naechsten freien Block im Random-Access File DATA.BERG.<expname>, gespeichert. Die Adresse des Anfangsblocks wird in SHTBN abgelegt. Ein Schuss-Segment enthaelt die Directory und die bereinigten Daten. Die Groesse l der Directory in logischen Records richtet sich nach der Zahl anz der in den Rohdaten angeschlossenen Diagnostiken (aus Offset 20 des Headers DATA.INTZ.<expname> mit Subrcutine ANZD)

l := {[anz + halba(5)]*halba(4) + bksize - 1} / bksize.

Die Directory besteht aus dem Header

0	2	Laenge der Directory in Bloecken
2	1	Version Number des Namelist Files
3	3	Experiment Name
6	2	Laenge des logischen Records in Bytes
8	2	Anzahl logischer Records im Header
10	2	Schussnummer
12	8	angeschlossene Diagnostiken
20	2	Anzahl bereinigter Diagnostiken
22	2	Laenge der Daten in Bloecken
24	1	Tages-Byte
25	1	frei
26	3	Datum des letzten Eintrags
29	3	Uhrzeit des letzten Eintrags
32	30	5 Jobnummern mit je 6 Bytes

62	2	frei

und den Diagnostik-Items

0	1	Diagnostiknummer
1	1	Index der Jobnummer
2	2	Pointer auf Anfangsknoten im Namelist File in logischen Records
4	4	Pointer auf Datenbeginn in Bytes
8	4	Laenge der Daten in Bytes
12	1	Laenge des Daten-Items
13	1	Art des Daten-Items
14	2	Nummer des Eichschusses
16	2	Nummer des Auswerte-Programms
18	3	Datum der Auswertung
21	3	Uhrzeit der Auswertung
24	8	frei.

Nach den Diagnostikitems folgen zwei weitere Items, die die Tabelle der zu diesem Schuss gehoerigen Bloecke enthaelt (Laenge : 64 Bytes). Der Anfangsblock ist die Directory und liegt daher fest.

Die Tabelle ist folgendermassen aufgebaut:

0	4	Anzahl der Bloecke, die in aufsteigender Ordnung folgen
4	4	Nr. Des naechsten zugehoerigen Blocks, der die Reihenfolge durchbricht
8	4	Anzahl der folgenden Bloecke
12	4	Nr. Des naechsten Blocks
u.s.w.		
56	4	Anzahl der folgenden Bloecke
60	4	0

Wenn die letzten beiden Ganzworte (52 und 56) besetzt werden, wird Maintenance automatisch angefordert.

Das Tages-Byte wird jeweils bei Uebergabe oder Aenderung des Schuss-Segmentes gemaess Tageszaehler in der Directory SHTBN gesetzt.

Die bereinigten Daten beginnen in dem auf die Directory folgenden Block. Die Daten einer Diagnostik beginnen im naechsten auf die Daten der vorhergehenden Diagnostik folgenden Fullword. Alle Datenbloecke zeigen mit

0	4	Pointer

auf den naechsten von dieser Diagnostik oder dem laufenden Schuss belegten Datenblock, der Pointer des letzten Blocks wird geloescht.

Die Daten beginnen daher an beliebiger Stelle innerhalb eines Blocks an dem Offset eines Fullword mit den Prolog

0	2	Diagnostik Nummer
2	2	Adresse der Diagnostik in der Directory in Ganzworten.

Bei spaeterer Auswertung einzelner Diagnostiken koennen Daten nachgetragen werden. Sie werden an die bestehende Schussnummer in der Art von Listenstrukturen angehaengt, jedoch im letzten nicht abgeschlossenen Block begonnen.

Auf dem ersten Block von DATA.INTZ.<expname> und DATA.BERG.<expname> steht im ersten Ganzwort jeweils die Groesse der Platte.

2.3. Directory SHTBN

Sie enthaelt Aufzeichnungen ueber die on-line Schuesse der bereinigten Daten. An den Header

0	2	Gesamtlaenge in Bytes
2	4	frei

6	2	Laenge des logischen Records in Bytes
8	2	Anzahl logischer Records im Header
10	2	Anzahl on-line Schuesse
12	1	aeltester Schusstag
13	5	frei
18	2	niedrigste on-line Schussnummer
20	2	hoechste on-line Schussnummer
22	1	Tageszaehler
23	1	Maintenance Byte
24	4	Anfangsadresse des freien Bereichs
28	3	Datum der Eroeffnung
31	3	Uhrzeit der Eroeffnung
34	3	Datum der letzten Aenderung
37	3	Uhrzeit der letzten Aenderung
40	2	erste freie Folgeadresse
42	22	frei

schliessen sich die Schuss-Items

0	2	Tausender der Schussnummer
2	2	Folgeadresse fuer weitere Schussnummern mit dem gleichen Divisionsrest
4	4	Adresse Anfangsblock der Schussdaten

an. Die Adresse des Schuss-Items wird durch

[mod(shnr,1000) + halba(5)] * halba(4) + 1

bestimmt. Es sind 28 Bloecke fuer die Directory vorgesehen, sodass etwa 1783 Schuesse aufgenommen werden koennen.

2.4. Maintenance und Migrate

Zuerst wird das Migration Programm gestartet, um moegliche Doppelarbeit zu vermeiden. Sobald der freie Speicherplatz weniger als 150 maximale Schuesse aufnehmen kann, werden die bereinigten Daten des aeltesten Schusstages sequentiell auf Band uebertragen. Nach Aufsuchen des Schuss-Segment-Items wird an der Anfangsadresse der Schussdaten begonnen. Der Pointer am Anfang jedes Blocks fuehrt zu den weiteren

Bloecken. Wegen der relativen Adressierung innerhalb eines Segmentes brauchen keine Adressen geaendert zu werden. In SDIR(I).TAB(J) werden geeignete Eintragungen hinterlassen, im Schuss-Segment-Item der Directory die Anfangsadresse geloescht.

Anschliessend wird das Maintenance Programm gestartet. Es endet ohne Aktion, falls das Maintenance-Byte FALSE gesetzt ist. Im anderen Falle wird zuerst die Directory SHBTN auf den neuesten Stand - nach durchgefuehrtem Maintenance und Condense - gebracht, danach werden die verbliebenen Schuss-Segmente sequentiell auf Band ausgegeben. Nach Datenende und Rewind des Bandes werden sie in der ausgegebenen Folge wieder auf die Platte uebertragen. Durch die Art des Vorgehens, die Blockpointer, die relative Adressierung innerhalb eines Schuss-Segmentes und das Loeschen der ausgelagerten Schuesse werden ineffiziente Transaktionen vermieden.

Organisation, Handling und Benutzung des Namelist Files der Phase 2 werden noch entworfen.